Università degli Studi di Cagliari

Dipartimento di Matematica e Informatica

Doctor of Philosophy in

Computer Science

Similarity and Diversity:

Two Sides of the Same Coin in the Evaluation of Data Streams

Doctoral Dissertation of

Roberto Saia

PhD Coordinator:

Prof. Gian Michele Pinna

Supervisor:

Prof. Salvatore Carta

Academic Year 2014/15 - XXVIII Cycle - INF/01

Men are only as good as their technical development allows them to be.

George Orwell

Acknowledgements

We gratefully acknowledge the *Regione Autonoma della Sardegna* for the financial support to my PhD program under the project *Social Glue*, through *PIA, Pacchetti Integrati di Agevolazione Industria, Artigianato e Servizi* (Annuity 2010), as well as the *Ministero dell'Istruzione, dell'Università e della Ricerca* (MIUR) for the financial support through *PRIN 2010-11*, under the *Security Horizons* project.

Moreover, a special thanks to Dr. *Ludovico Boratto*, extremely qualitative and efficient co-supervisor of my thesis work, together with Prof. *Salvatore Carta*.

Abstract

The *Information Systems* represent the primary instrument of growth for the companies that operate in the so-called e-commerce environment. The *data streams* generated by the users that interact with their websites are the primary source to define the user behavioral models.

Some main examples of *services* integrated in these websites are the *Recommender Systems*, where these models are exploited in order to generate recommendations of items of potential interest to users, the *User Segmentation Systems*, where the models are used in order to group the users on the basis of their preferences, and the *Fraud Detection Systems*, where these models are exploited to determine the legitimacy of a financial transaction.

Even though in literature *diversity* and *similarity* are considered as two sides of the same coin, almost all the approaches take into account them in a mutually exclusive manner, rather than jointly. The aim of this thesis is to demonstrate how the consideration of both sides of this coin is instead essential to overcome some well-known problems that afflict the state-of-the-art approaches used to implement these services, improving their performance.

Its contributions are the following: with regard to the recommender systems, the detection of the diversity in a user profile is used to discard incoherent items, improving the accuracy, while the exploitation of the similarity of the predicted items is used to re-rank the recommendations, improving their effectiveness; with regard to the user segmentation systems, the detection of the diversity overcomes the problem of the non-reliability of data source, while the exploitation of the similarity reduces the problems of understandability and triviality of the obtained segments; lastly, concerning the fraud detection systems, the joint use of both diversity and similarity in the evaluation of a new transaction overcomes the problems of the data scarcity, and those of the non-stationary and unbalanced class distribution.

Contents

Chapter 1

Introduction

In the last times, the Information Systems (IS) represent the primary instrument of growth for the companies that operate in the so-called e-commerce environment. Indeed, these systems aggregate the data they collected to generate information provided to both the users (e.g., items recommendations) and to those who run the business (e.g., segments of users to target, or possible fraudulent transactions). In order to generate these types of services, an IS has to build predictions on the usefulness of a particular piece of information (i.e., it has predict wether or not an item should be recommended to a user, in which user segment a user should be placed, or if a transaction can be successfully completed). In this matter, detecting the *similarity* and *diversity* between the behavior of a user and that of the other users, or with respect to her/his previous behavior, is essential in order to build accurate predictions. Therefore, the main objective of this thesis is to *inspect on* *the role of the similarity of diversity in these classes of IS, and to show why they represent two faces of the same coin (i.e., why it is necessary to exploit both of them in order for a system to perform well).*

1.1 Information Systems and Joint Evaluation of Similarity and Diversity: Motivation

1.1.1 Recommender Systems

One of the most important classes of IS are the Recommender Systems (RSs), since their ability to perform accurate prediction on the future user preferences about items is strongly (and directly) related to the earnings of the commercial operators. Considering that, typically, a RS produces its results on the basis of the historic interactions of the users with it, by evaluating the similarity/diversity between their previous choices and the items not evaluated yet, the ability to define a user profile able to reflect the real tastes of them represents a crucial task.

In order to face this problem, the joined evaluation of similarity and diversity can lead toward significant improvements. In fact, the predictive models of a RS are usually determined through the analysis of the data stream related with the past activities of the users, and the similarity aspect (i.e., between users or items, in accordance with the adopted recommendation strategy) represents the primary criterion to determine their output. In such context, the diversity aspect is considered as a mere implicit element of the problem, a specular factor of the similarity, which in many cases it is not even taken into account by the involved strategies.

This happens because almost all these systems operate in accord with the assumption that the past choices of the users represent a reliable source of information that can be exploited in order to infer their future preferences. For this reason, they usually generate the recommendations on the basis of the interpretation of all historic interactions of the users with them, using algorithms primarily based on the evaluation of the similarity between items (similarity with items not yet evaluated and items already evaluated in the past) or between users (similarity with the other users who share part of the past choices of her/his).

Although it may sound correct, such approach could lead to wrong results due to several factors, such as a changes in user taste over time, the use of her/his account by third parties, or when the system does not allow their users to express a feedback (or when it is possible, but they do not use this option). A RS that adopts the previously mentioned criteria of similarity produces non optimal results. It happens because its recommendations are based only on the explicit characteristics of the users, which can be trivial, since present a low level of novelty and serendipity (i.e., the ability to suggest something interesting to users, without they have expressly searched it).

1.1.2 User Segmentation Systems

Another important class of IS are those that perform a segmentation of users with related interests, in order to target them (*behavioral targeting*). The set of target users is detected from a segmentation of the user set, based on their interactions with the website (pages visited, items purchased, etc.). Recently, in order to improve the segmentation process, the semantics behind the user behavior has been exploited, by analyzing the queries issued by the users. However, nearly half of the times users need to reformulate their queries in order to satisfy their information need. In this thesis, we tackle the problem of semantic behavioral targeting considering *reliable* user preferences, by performing a semantic analysis on the descriptions of the items positively rated by the users. We also consider widely-known problems, such as the *interpretability* of a segment, and the fact that *user preferences are usually stable over time*, which could lead to a trivial segmentation. In order to overcome these issues, our approach allows an advertiser to automatically extract a user segment by specifying the interests that she/he wants to target, by means of a novel boolean algebra; the segments are composed of users whose evaluated items are semantically related to these interests. This leads to interpretable and non-trivial segments, built by using reliable information.

1.1.3 Fraud Detection Systems

Any business that operates on the Internet and accepts payments through debit or credit cards, also implicitly accepts that some transaction may be fraudulent. The design of effective strategies to face this problem is challenging, due to factors such as the heterogeneity and the non stationary distribution of the data, as well as the presence of an imbalanced class distribution, and the scarcity of public datasets. The state-of-the-art strategies are usually based on a unique model, built by analyzing the past transactions of a user, whose similarity with the current transaction is analyzed. In order to overcome the aforementioned problems, it would be advisable to generate a set of models (behavioral patterns) to evaluate a new transaction, by considering the behavior of the user in different temporal frames of her/his history. These models can be built by evaluating different forms of similarity and diversity between the financial transactions of a user.

1.2 Contributions

It should be observed, which in spite the fact that similarity and diversity can be considered as two sides of the same coin, in many contexts these two factors are taken into account in a mutually exclusive manner, rather than jointly.

For instance, almost all the approaches at the state of the art, used to generate recommendations, are basically based on metrics of similarity between users/items, without taking into account any factor of diversity. Otherwise, by taking in consideration the diversity between items, they could evaluate the coherence in the past choices of the users, removing from their profiles the incoherent elements, making these as close as possible to their real tastes. This thesis shows how by performing a pre-processing phase (based on the concept of diversity and addressed to remove the incoherent items from the user profiles), followed by a post-processing phase (based instead on the concept of similarity and aimed to re-rank the suggestions generated by a state-of-the-art algorithm of recommendation), we can lead the recommender system toward better performance.

This thesis also proposes a novel approach to improve the user segmentation process, by reducing the triviality that characterizes the results of many of the state-of-the-art approaches. It is based on the evaluation of the semantic similarity/diversity between users, in terms of preference for classes of items, allowing us to group them in a non-trivial way, increasing the serendipity factor when the results are exploited to perform a behavioral targeting.

In the fraud detection context, where the previously mentioned problems occur, this thesis proposes a new strategy to detect frauds. The main idea is to perform a joined evaluation of the similarity/diversity between the financial transaction to evaluate and a series of models defined by using different temporal frames of the user activity (exploiting all transaction fields). It allows us to overcome the problem of data scarcity (by using multiple models) and data unbalance (it does not use fraudulent transactions to train the models), operating in a proactive mode.

1.3 Thesis Structure

This thesis is organized as follows: it first presents the background and related work (Chapter I) of the main concepts involved in the performed research, continuing by presenting all details (adopted notation, problem definition, used datasets, involved metrics, adopted strategy, experiments, and conclusions) about the three areas taken in account in this thesis, respectively related with the role of the similarity and diversity in the context of the recommender systems (Chapter II), in that of the user segmentation systems (Chapter III), and in that of the fraud detection systems (Chapter IV). It ends with some concluding remarks (Chapter V).

Background and Related Work

Chapter 2

Recommender Systems

2.1 Introduction

The context taken in consideration is that of the *Recommender Systems* (RS) [1], where the rapid growth of the number of companies that sell goods through the Word Wide Web has generated an enormous amount of valuable information, which can be exploited to improve the quality and efficiency of the sales criteria [2]. Because of the widely-known information overload problem, it became necessary to deal with the large amounts of data available on the Web [3]. The recommender systems represent an effective response to this problem, by filtering the huge amount of information about their customers in order to get useful elements to produce suggestions to them [4, 5, 6]. The denomination RS denotes a set of software tools and techniques providing to a user suggestions for items, where the term *item* is used to indicate what the system recommends to users. This research addresses one of the most important aspects related to the recommender systems, i.e., *how to represent a user profile, so that it only contains accurate information about a user, and it allows a system to generate effective recommendations*.

2.2 User Profiling

When it comes to producing personalized recommendations to users, the first requirement is to understand the needs of the users and, according to them, to build a user profile that models these needs. User profiles and context information are the key elements that allow to perform personalized recommendations by a wide range of techniques developed for using profile information to influence different aspects of search experience. There are several approaches to build profiles: some of them focus on *short-term* user profiles that capture features of the user's current search context [7, 8, 9], while others accommodate *long-term* profiles that capture the user preferences over a long period of time [10, 11, 12]. As shown in [13], compared with the *short-term* user profiles, the use of a *long-term* user profiles generally produces more reliable results, at least when the user preferences are fairly stable over a long time period. Otherwise, we need a specific strategy able to manage the changes in the user profile that not reflect the real taste of the user and that represent a form of "noise".

Given this analysis of the literature, the definition of approaches able to detect the presence of *diversity* in a user profile represents a novel problem.

Some important concepts related to the so-called Natural Language Processing (NLP) are also presented in Appendix A.

2.2.1 Explicit, Implicit and Hybrid Strategies

The most common strategies to get useful information to build the user profiles are two, i.e., explicit or implicit, or even a combination of these (hybrid strategies). Explicit profiling strategies interrogate users directly by requesting different forms of preference information, from categorical preferences [10, 12] to simple result ratings [11]. Implicit profiling strategies attempt to infer preference information by analyzing the user behavior, and without a direct interaction with users while they perform actions in a website [10, 14, 15]. The hybrid strategies combine the advantages of the implicit and explicit approaches of user profiling, by considering both the static and dynamic characteristics of the users, these last ones obtained by retrieving the behavioral information of them. This approach represents a good compromise between advantages and disadvantages related with the two main approaches of user profiling (i.e., explicit and implicit).

An explicit way to build the user profiles is reported in [16], where the user profiling activity is considered as a process of analyzing of the static and inferable characteristics of the users. Following this approach, their behavior is inferred by the analysis of the available information about them, usually collected through the use of on-line forms or other similar methods (e.g., specific surveys). This approach is classified as *static profiling* or *factual profiling*. It should be noted that such strategy of data collecting presents some problem, such as those related with the *privacy* aspect (many users do not like to reveal their information), and those related with the form filling process (many users do not like to spend their time for this activity). Regards this last kind of problem, is observed as the accuracy of a filled form depends on the time needed to fill it.

The same work [16] reports a dynamic user profiling strategy, which instead to adopt a static approach of data collecting, based exclusively on the explicit information of the users, tries to learn more data about them. Such strategy is also classified as *behavioral profiling*, *adaptive profiling*, or *ontological profiling* of the users. It is performed by exploiting several filtering approaches, such as the *rule based filtering*, the *collaborative filtering*, and the *content based filtering* [17].

As previous said, the hybrid strategies represent a good compromise between the advantages and the disadvantages related with the implicit and explicit approaches of user profiling. A more sophisticated hybrid approach is reported in [18], a strategy for learning the user profiles from both static and dynamic information. In addition to the canonical static information about the users, it exploits the tags associated with the items rated by the users. The tags taken in consideration are the *user tags*, but also the so-called *social tags*, i.e., the tags used by other users who rated the same items. It should be observed how this way to proceed, based allows us to exploit the different knowledge of the users in the domain taken in consideration, because the *social tags* represent a way to extend the *content-based* paradigm toward a *hybrid content-collaborative* paradigm [19].

In order to face the problem related with the non univocity of the tags (due to the fact that they are arbitrarily chosen by users), in the same work [19] is suggested a semantic approach of disambiguation (i.e., word sense disambiguation) performed by exploiting a lexical ontology such as Wordnet [20, 21]. Another hybrid approach of user profiling, where the content-based profiles and the interests revealed through tagging activities are combined, is reported in [22].

Concluding, it is possible to state that the strategies of user profiling that proved to be most effective are the implicit ones, where the preferences of the users are inferred without any direct interaction with her/him. These implicit approaches usually requires *long-term* user profiles, where the information about the tastes is considered over an extended period of time. However, there are some implicit approaches that involve a *short-term* profiling, related to the particular context in which the system operates [7].

2.2.2 Information Reliability

Regardless of the type of profiling that is adopted (e.g., *long-term* or *short-term*), there is a common problem that may affect the goodness of the obtained results, i.e., the capability of the information stored in the user profile to lead toward reliable recommendations. Unreliable information in a user's profile can be found in many cases, e.g. when a private user account is used by other people, when the user has expressed a wrong preference about an item, and so on. In order to face the problem of dealing with unreliable information in a user profile, the state of art proposes different strategies.

Several approaches, such as [23], take advantage from the Bayesian analysis of the user provided relevance feedback, in order to detect non-stationary user interests. The work [24] describes an approach to learn the users preferences in a dynamic way, a strategy able to work simultaneously in short-term and long-term domains. Also exploiting the feedback information provided by the users, other approaches such as [13] make use of a *tree-descriptor* model to detect shifts in user interests. Another technique exploits the knowledge captured in an ontology [25] to obtain the same result, but in this case it is necessary that the users express their preferences about items through an explicit rating. There are also other different strategies that try to improve the accuracy of information in the user profiles by collecting the implicit feedbacks of the users during their natural interactions with the system (reading-time, saving, etc.) [26].

However, it should be pointed out that most of the strategies used in this area are usually effective only in specific contexts, such as for instance [27], where a novel approach to model automatically the user profile according to the change of her/his tastes is designed to operate in the context of the articles recommendation. Despite the fact that implicit feedbacks from users are usually less accurate than those explicitly expressed, in certain contexts this approach leads toward pretty good results.

2.2.3 Magic Barrier Boundary

It should be noted that there is a common issue, related to the concept of items incoherence, that afflicts the recommender approaches. This is a problem that in the literature is identified as *magic barrier* [28], a term used to define the theoretical boundary for the level of optimization that can be achieved by an algorithm of recommendation on known transactional data [29].

The inconsistency in the behavior of the users represents a well known aspect in the context of recommender systems, a problem it has been investigated since this study [30], where the reliability of the user ratings is questioned, as well as in the work [28], which the level of noise in the user ratings has been discussed. The evaluation models assume as a ground truth that the transactions made in the past by the users, and stored in their profiles, are free of noise.

This is a concept that has been studied in [31, 32], where a study aimed to capture the noise in a service that operates in a synthetic environment was performed. It should be noted that this is an aspect that, in the context of the recommender systems, was mentioned for the first time in 1995, in a work [30] aimed to discuss the concept of reliability of users in terms of rating coherence, as well as in the work [28], where the level of noise in the user ratings has been discussed.

The proposed approach differs from the others in the literature, in the sense that it does not need to focus on a specific type of profile (i.e., *short-term* or *long-term*), it can operate with any type of data that contains a textual description, and it overcomes the limitation introduced by the magic barrier from a novel perspective, represented by the semantic analysis of the items.

2.3 Decision Making Process

Content-based recommender systems suggest to users items that are similar to those they previously evaluated [33, 34]. The early systems used relatively simple retrieval models, such as the Vector Space Model, with the basic TF-IDF weighting. The Vector Space Model is a spatial representation of text documents, where each document is represented by a vector in a n-dimensional space (known as *bag of words*, and each dimension is related to a term from the overall vocabulary of a specific document collection. Examples of systems that employ this type of content filtering are [35, 36, 37, 38]. Due to the fact that the approach based on a simple bag of words is not able to perform a semantic disambiguation of the words in an item description, content-based recommender systems evolved and started employing external sources of knowledge (e.g., ontologies) and semantic analysis tools, to improve their accuracy [39, 40, 41].

Regarding the user profile considered by a recommender system, there is a common problem that may affect the effectiveness of the obtained results, i.e., the capability of the information stored in the user profile to lead toward reliable recommendations. In order to face the problem of dealing with unreliable information in a user profile, the state of art proposes different strategies. Several approaches, such as [23], take advantage from the Bayesian analysis of the user provided relevance feedback, in order to detect non-stationary user interests. Also exploiting the feedback information provided by the users, other approaches such as [13] make use of a tree-descriptor model to detect shifts in user interests. Another technique exploits the knowledge captured in an ontology [25] to obtain the same result, but in this case it is necessary that the users express their preferences about items through an explicit rating. In [42, 43, 44], the problem of modeling semantically correlated items was tackled, but the authors consider a temporal correlation and not the one between the items and a user profile.

Considering the item incoherence problem, it should be noted that there is another common issue that afflicts the recommendation approaches. This is a problem that in the literature is identified as *magic barrier*. The evaluation models assume as a ground truth that the transactions made in the past by the users, and stored in their profiles, are free of noise. This is a concept that has been studied in [31, 32], where a study aimed to capture the noise in a service that operates in a synthetic environment was performed.

No approach in the content-based recommendation literature ever studied how the architecture and the flow of computation might be affected by the item incoherence and magic barrier issues. It is then an open problem to be tackled.

2.3.1 Non-personalized Models

The recommender systems based on the so-called non-personalized model [45], propose to all users the same list of recommendations, without taking into account their preferences. This static approach is usually based on two algorithms, the first of them (TopPop), operates by suggesting the most rated items (i.e., those most popular), while the second (MovieAvg), works by suggesting the highest rated items (i.e., those most liked).

The exclusive use of the non-personalized models, leads toward the absence of two important characteristics that a recommender system should have, i.e., novelty and serendipity [46]. Novelty occurs when a system is able to recommend unknown items that a user might have autonomously found, while the serendipity happens when it helps the user to find a surprisingly interesting item that a user might not have otherwise found, or if it is very hard to find.

2.3.2 Latent Factor Models

The type of data with which a recommendation system operates is typically a sparse matrix where the rows represent the users, and the columns represent the items. The entries of this matrix are the interaction between users and items, in the form of ratings or purchases. The aim of a recommender system is to infer, for each user u, a ranked list of items, and in literature many of them are focused on the rating prediction problem [1]. The most effective strategies in this field exploit the so-called latent factor models, but especially, the matrix factorization techniques [47].

Other CF ranking-oriented approaches that extend the matrix factorization techniques, have been recently proposed, and most of them use a ranking oriented objective function, in order to learn the latent factors of users and items [48].

SVD++, the Koren's version of the Singular Value Decomposition (SVD) [49], is today considered one of the best strategies in terms of accuracy and scalability.

Chapter 3

User Segmentation Systems

3.1 Introduction

Behavioral targeting addresses ads to a set of users who share common properties. In order to choose the set of target users that will be advertised with a specific ad, a *segmentation* that partitions the users and identifies groups that are meaningful and different enough is first performed.

3.2 Latent Space Discovering

The user segmentation is a process aimed at partitioning the potential audience of an advertiser into several classes, according to specific criteria. Almost all the existing approaches take into account only the explicit preferences of the users, without considering the hidden semantics embedded in their choices, so the target definition is affected by widely-known problems. The most important is that easily understandable segments are not effective for marketing purposes due to their triviality, whereas more complex segmentations are hard to understand. For this reason, the definition of a new strategy able to perform an untrivial grouping of the users is an open problem.

3.2.1 Behavioral Targeting

A high variety of behavioral targeting approaches has been designed by the industry and developed as working products. Google's *AdWords*[1] performs different types of targeting to present ads to users; the closest to our proposal is the "Topic targeting", in which the system groups and reaches the users interested in a specific topic. *DoubleClick*[2] is another system employed by Google that exploits features such as browser information and the monitoring of the browsing sessions. In order to reach segments that contain similar users, Facebook offers *Core Audiences*[3], a tool that allows advertisers to target users with similar location, demographic, interests, or behaviors; in particular, the interest-based segmentation, allows advertisers to choose a topic and target a segment of users interested by it. Among its user targeting strategies, Amazon offers the so-called *Interest-based ads policy*[4], a service that detects and targets segments of users with similar interests, based on what the users purchased, visited, and by monitoring different forms

[1] https://support.google.com/adwords/answer/1704368?hl=en

[2] https://www.google.com/doubleclick/

[3] https://www.facebook.com/business/news/Core-Audiences

[4] http://www.amazon.com/b?node=5160028011

of interaction with the website (e.g., the Amazon Browser Bar). *SpecificMedia*[5] uses anonymous web surfing data in order to predict a user's purchase prediction score. *Yahoo! Behavioral Targeting*[6] creates a model with the online interactions of the users, such as searches, page-views, and ad interactions to predict the set of users to target. Other commercial systems, such as *Almond Net*[7], *Burst*[8], *Phorm*[9], and *Revenue Science*[10] include behavioral targeting features.

Research studies, such as the one presented by Yan et al. [50], show that an accurate monitoring of the click-through log of advertisements collected from a commercial search engine can help online advertising. Beales [51] collected data from online advertising networks and showed that a behavioral targeting performed by exploiting prices and conversion rates (i.e., the likelihood of a click to lead to a sale) is twice more effective than traditional advertising. Chen et al. [52] presented a scalable approach to behavioral targeting, based on a linear Poisson regression model that uses granular events (such as individual ad clicks and search queries) as features. Approaches to exploit the semantics [53, 54] or the capabilities of a recommender system [4, 5, 6] to improve the effectiveness of the advertising have been proposed, but none of them generates segments of target users.

[5] http://specificmedia.com/

[6] http://advertising.stltoday.com/content/behavioral_FAQ.pdf

[7] http://www.almondnet.com/

[8] http://www.burstmedia.com/

[9] http://www.phorm.com/

[10] http://www.revenuescience.com/

3.2.2 Reliability of a semantic query analysis

In the literature it has been highlighted that half of the time users need to reformulate their queries, in order to satisfy their information need [55, 56, 57]. Therefore, the semantic analysis of a query is not a reliable source of information, since it does not contain any information about whether or not a query led to what the user was really looking for. Moreover, performing a semantic analysis on the items evaluated by the users in order to perform a filtering on them can increase the accuracy of a system [53, 54, 58]. Therefore, a possible way to overcome this issue would be to perform a semantic analysis on the description of the items a user positively evaluated through an explicitly given rating. However, another issue arises in cascade.

3.2.3 Segment Interpretability and Semantic User Segmentation

Choosing the right criteria to segment users is a widely studied problem in the market segmentation literature, and two main classes of approaches exist. On the one hand, the *a priori* [59] or *commonsense* [60] approach is based on a simple property, like the age, which is used to segment the users. Even though the generated segments are very easy to understand and they can be generated at a very low cost, the segmentation process is trivial and even a partitioning with the k-means clustering algorithm has proven to be more effective than this method [61]. On the other hand, *post hoc* [62] approaches (also known as *a posteriori* [59] or *data-driven* [60]) combine a set of features (which are known as *segmentation* *base* [63]) in order to create the segmentation. Even though these approaches are more accurate when partitioning the users, the problem of properly understanding and interpreting results arises [64, 65]. This is mostly due to the lack of guidance on how to interpret the results of a segmentation [66].

Regarding the literature on behavioral user segmentation, Bian et al. [67] presented an approach to leverage historical user activity on real-world Web portal services to build behavior-driven user segmentation. Yao et al. [68] adopted SOM-Ward clustering (i.e., Self Organizing Maps, combined with Ward clustering), to segment a set of customers based on their demographic and behavioral characteristic. Zhou et al. [69] performed a user segmentation based on a mixture of factor analyzers (MFA) that consider the navigational behavior of the user in a browsing session. Regarding the semantic approaches to user segmentation, Tu and Lu [70] and Gong et al. [71] both proposed approaches based on a semantic analysis of the queries issued by the user through Latent Dirichlet Allocation-based models, in which users with similar query and click behaviors are grouped together. Similarly, Wu et al. [72] performed a semantic user segmentation by adopting a Probabilistic Latent Semantic Approach on the user queries. As this analysis showed, none of the behavioral targeting approaches exploits the interactions of the users with a website in the form of a positive rating given to an item.

3.2.4 Preference Stability

Burke and Ramezani highlighted that some domains are characterized by a stability of the preferences over time [73]. Preference stability leads also to the fact that when users get in touch with diverse items, diversity is not valued [74]. On the one side, users tend to access to agreeable information (a phenomenon known as *filter bubble* [75]) and this leads to the overspecialization problem [33], while on the other side they do not want to face diversity. Another well-known problem is the so called *selective exposure*, i.e., the tendency of users to make their choices (goods or services) based only on their usual preferences, which excludes the possibility for the users to find new items that may be of interest to them [76]. The literature presents several approaches that try to reduce this problem, e.g., *NewsCube* [77] operates offering to the users several points of views, in order to stimulate them to make different and unusual choices.

3.2.5 Item Descriptions Analysis

For many years the item descriptions were analyzed with a word vector space model, where all the words of each item description are processed by TF-IDF [78] and stored in a weighted vector of words. Due to the fact that this approach based on a simple *bag of words* is not able to perform a semantic disambiguation of the words in an item description, and motivated by the fact that exploiting a taxonomy for categorization purposes is an approach recognized in the literature [79], we decided to exploit the functionalities offered by the *WordNet* environment. More details about the *bag of words* and *Wordnet* are reported in Appendix A.

Chapter 4

Fraud Detection Systems

4.1 Introduction

Nowadays, any business that carries out activities on the Internet and accepts payments through debit or credit cards, also implicitly accepts all the risks related to them, like for some transaction to be fraudulent. Although these risks can lead to significant economic losses, nearly all the companies continue to use these powerful instruments of payment, as the benefits derived from them will outweigh the potential risks involved.

Fraud is one of the major issues related with the use of debit and credit cards, considering that these instruments of payment are becoming the most popular way to conclude every financial transaction, both online and in a traditional way. According to a study of some years ago conduct by the *American Association of*

Fraud Examiners[1], fraud related with the financial operations are the 10-15% of the whole fraud cases. However, this type of fraud is related to the 75-80% of all involved finances with an estimated average loss per fraud case of 2 million of dollars, in the USA alone. The research of efficient ways to face this problem has become an increasingly crucial imperative in order to eliminate, or at least minimize, the related economic losses.

4.2 A Proactive Approach for the Detection of Frauds Attempts

As highlighted in many studies, frauds represent the biggest problem in the E-commerce environment. The credit card fraud detection represents one of the most important contexts, where the challenge is the detection of a potential fraud in a transaction, through the analysis of its features (i.e., description, date, amount, an so on), exploiting a user model built on the basis of the past transactions of the user. In [80], the authors show how in the field of automatic fraud detection there is lack of real datasets (publicly available) indispensable to conduct experiments, as well as a lack of publications about the related methods and techniques.

The most common causes of this problem are the policies (for instance, competitive and legal) that usually stand behind every E-commerce activity, which makes it very difficult to obtain real data from business. Furthermore, such datasets composed by real information about user transactions could also reveal the poten-

[1] http://www.acfe.com

tial vulnerabilities in the related E-commerce infrastructure, with a subsequent *loss of trust*.

Literature underlines how the main two issues in this field are represented by the data unbalance (i.e., the number of fraudulent transactions is typically much smaller than legitimate ones), and by the overlapping of the classes of expense of a user (i.e., due to the scarcity of information that characterizes a typical record of a financial transaction). A novel approach able to face these two aspects represents then a challenging problem.

4.2.1 Supervised and Unsupervised Approaches

In [81] it is underlined how the *unsupervised* fraud detection strategies are still a very big challenge in the field of E-commerce. Bolton and Hand [82] show how it is possible to face the problem with strategies based both on statistics and on *Artificial Intelligence* (*AI*), two effective approaches in this field able to exploit powerful instruments (such as the *Artificial Neural Networks*) in order to get their results.

In spite the fact that every *supervised* strategy in fraud detection needs a reliable training set, the work proposed in [82] takes in consideration the possibility to adopt an *unsupervised* approach during the fraud detection process, when no dataset of reference containing an adequate number of transactions (legitimate and non-legitimate) is available. Another approach based on two *data mining* strategies (*Random Forests* and *Support Vector Machines*) is introduced in [83], where the effectiveness of these methods in this field is discussed.

4.2.2 Data Unbalance

The unbalance of the transaction data represents one of the most relevant issues in this context, since almost all of the learning approaches are not able to operate with this kind of data structure [84], i.e., when an excessive difference between the instances of each class of data exists. The unbalanced training sets represent one of the most big problems in the context of supervised learning, because the presence of a huge disproportion in the number of instances of the classes generates a wrong classification of the new cases (i.e., they are assigned to the majority class). This happens because the canonical learning approaches are not able to perform a correct classification of the new cases in such contexts, in fact they report a good accuracy only for the cases that belong to the majority class, reporting unacceptable values of accuracy for the other cases that belong to the minority class. In other words, it means that is possible that, in presence of a data unbalance, a classifier predicts all the new cases as belonging to the major class, ignoring the minor class.

To face this problem, several techniques of pre-processing have been developed, aimed to balance the set of data [85], and they can be grouped into three main categories: *sampling based*, *algorithms based*, and *feature-selection based.*

Sampling based: this is a pre-processing strategy that faces the problem by re-sampling the set of data. The sampling can be performed through different ways:

by *under-sampling* the majority class, by *over-sampling* the minority class, or by an *hybrid-sampling* that combines these two approaches. The *under-sampling* technique randomly removes the transactions, until the balancing has been reached, while the specular *over-sampling* technique, obtains the balancing by adding new transactions, created through an interpolation of the elements that belong to a same class [86].

Algorithms based: this strategy is aimed to optimize the performance of the learning algorithm on unseen data. A *single-class* learning methods is used to recognize the cases that belongs to that class, rejecting the other ones. This is a strategy that in some contexts (i.e., the multi-dimensional data sets) gives better performance than the other strategies [87].

Feature-selection based: this strategy operates by selecting a subset of features (defined by the user) that allows a classifier to reach the optimal performance. In the case of big data sets, some filters are used in order to score each single feature, on the basis of a rule [87].

4.2.3 Detection Models

The *static approach* [88] represents a canonical way to operate to detect fraudulent events in a stream of transactions. It is based on the initial building of a user model, which is used for a long period of time, before its rebuilding. An approach characterized by a simple learning phase, but not able to follow the changes of user behavior during the time.

In a static approach, the data stream is divided into blocks of the same size, and the user model is trained by using a certain number of initial and contiguous blocks of the sequence, which use to infer the future blocks. In the so-called *updating approach* [89], instead, when a new block appears, the user model is trained by using a certain number of latest and contiguous blocks of the sequence, then the model can be used to infer the future blocks, or aggregated into a big model composed by several models. In another strategy, based on the so-called *forgeting approach* [90], a user model is defined at each new block, by using a small number of non fraudulent transactions, extracted from the last two blocks, but keeping all previous fraudulent ones. Also in this case, the model can be used to infer the future blocks, or aggregated into a big model composed by several models.

The main disadvantages related of these approaches of user modeling are: the incapacity to follow the changes in the users behavior, in the case of the *static approach*; the ineffectiveness to operate in the context of small classes, in the case of the *updating approach*; the computational complexity in the case of the *forgetting approach.*

There are several kind of approaches that are used in this context, such as those based on Data Mining [91], Artificial Intelligence [92], Fuzzy Logic [93], Machine Learning [94], or Genetic Programming [80]. However, regardless of the used approach, the problem of the non stationary distribution of the data, as well as that of the unbalanced classes distribution, remain still unaltered.

4.2.4 Differences with the proposed approach

The proposed approach introduces a novel strategy that, firstly, takes in account all elements of a transaction (i.e., numeric and non numeric), reducing the problem related with the lack of information, which leads toward an overlapping of the classes of expense [95]. The introduction of the *Transaction Determinant Field* (TDF) set, also allows to give more importance to certain elements of the transaction, during the model building. Secondly, differently from the canonical approaches at the state of the art, the proposed approach is not based on an unique model, but instead on multiple user models that involve the entire set of data. This allows us to evaluate a new transaction by comparing it with a series of behavioral models related with many parts of the user transaction history.

The main advantage of this strategy is the reduction, or removal, of the issues related with the stationary distribution of the data, and the unbalancing of the classes. This because the operative domain is represented by the limited event blocks, and not by the entire dataset. The discretization of the models, according to a certain value of d, permit us to adjust their sensitivity to the peculiarities of the operating environment.

In more details, regarding the analysis of the textual information related to the transactions, the literature presents several ways to operate, and most of them work in accord with the *bag-of-words* model, an approach where the words (for instance, type and description of the transaction) are processed without taking into account of the correlation between terms [23, 13].

This trivial way to manage the information does usually not lead toward good results, and just for this reason the basic approaches are usually flanked by complementary techniques aimed to improve their effectiveness [54, 79], or they are replaced by more sophisticated alternative based on the semantic analysis of the text [96], which proved to be effective in many contexts, such as the recommendation one [58].

Considering the nature of the textual data related to a financial transaction, the adoption of semantic techniques could lead toward false alarms, as well as a trivial technique based on simple matching between words. This happen because, a conceptual extension of a the textual field of a transaction could evaluate as similar two transactions instead very different, while a simple matching technique could lead to consider as different some string of text, due to the existence of some slight differences (i.e., plural forms instead of singular, words different but with a common root, and so on). For this reason, this work adopts the *Levenshtein Distance* described in Appendix B, a metric that measure the similarity between two textual fields in terms of minimal number of insertions, deletions, and replacements, needed to transforming the content of the first field into the content of the second one.

On the Role of Similarity and Diversity in Recommender Systems

Preface

The concepts of similarity and diversity are here taken in account in order to improve the reliability of the user profiles (i.e., making them as close as possible to the real tastes of the users), in the context of a recommender systems. Subsequently, the same concepts of similarity/diversity have been exploited to improve the decision making process of a recommender system that operates in the e-commerce environment.

Chapter 5

User Profiling

5.1 Introduction

The main motivation behind this work is that most of the solutions regarding the *user-profiling* involve the interpretation of the whole set of items previously evaluated by a user, in order to measure their similarity with those that she/he did not consider yet, and recommend the most similar items. Indeed, the recommendation process is usually based on the principle that users' preferences remain unchanged over time and this can be true in many cases, but it is not the norm due to the existence of temporal dynamics in their preferences. Therefore, as discussed in Chapter 2.2, a static approach to user profiling can lead toward wrong results due to various factors, such as a simple change of tastes over time or the temporary use of their own account by other people. Several works have showed that the user ratings can be considered as outliers, due to the fact that the same user may rate the same item with different ratings, at different moments of time. This is a well-known problem, which in literature is defined as *magic barrier* (Chapter 2.2.3).

Premising that the user profiling context taken into consideration is that related to recommendation systems, where such activity has a primary role, the proposed approach aims to evaluate the similarity between a single item and the others within the user profile, in order to improve the recommendation process by discarding the items that are highly dissimilar with the rest of the user profile. Considering that in the literature a formalization of a system architecture that implements such mechanism of profile cleaning does not exist, before moving toward the implementation steps, this research defines (first at a high-level, then in detail) such an architecture.

5.2 Architecture

5.2.1 A State-of-the-Art Architecture for Content-based Recommender Systems

This section will present the high-level architecture of a content-based recommender system proposed in [33] and presented in Figure 5.1. In order to highlight the limits of this architecture and present our proposal, we will explore it by presenting the flow of the computation of a system that employs it.

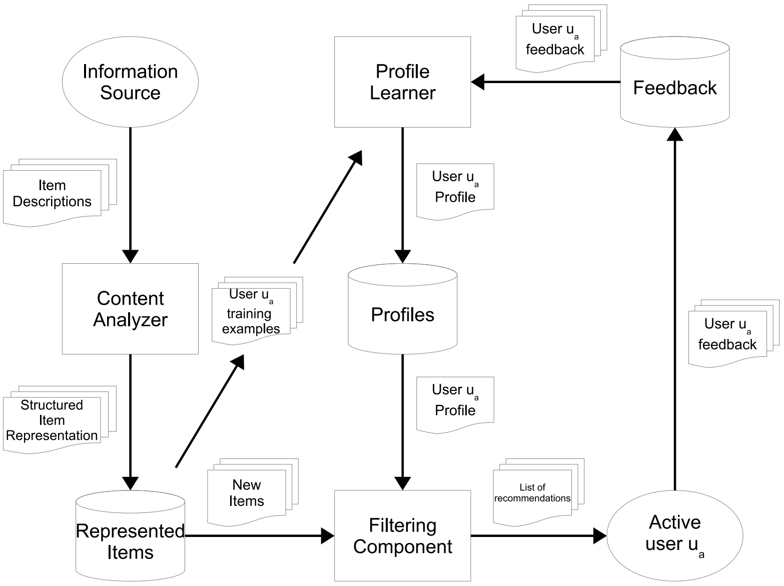

Figure 5.1: Architecture of a content-based recommender system.

The description of the items usually has no structure (e.g., text), so it is necessary to perform some pre-processing steps to extract some information from it. Given an *Information source*, represented by the *Item Descriptions* (e.g., product descriptions, Web pages, news, etc.) that will be processed during the filtering, the first component employed by a system is a CONTENT ANALYZER. The component coverts each item description into a format processable by the following steps (i.e., keywords, n-grams, concepts, etc.) thanks to the employment of feature extraction tools and techniques. The output generated by this component is a *Structured Item Representation*, stored in a *Represented Items* repository.

Out of all the represented items, the system consider the ones evaluated by each active user u_a to whom recommendations have to provided (*User u_a training examples*), in order to build a profile that contains the preferences of the user. This task is accomplished by a PROFILE LEARNER component, which employs Machine Learning algorithms to combine the *structured item representations* in a unique model. The output produced by the component is a *user profile*, stored in a *Profiles* repository.

The recommendation task is performed by a FILTERING COMPONENT, which compares the output of the two previous components (i.e., the profile of the active user and a set of items she/he has not evaluated yet). Given a new item representation, the component predicts wether or not the item is suitable for the active user u_a, usually with a value that indicates its relevance with respect to the user profile. The filtered items are ranked by relevance and the top-*n* items in the ranking represent the output produced by the component, i.e., a *List of recommendations*.

The *List of recommendation* is proposed to the *active user u_a*, which either accepts or rejects the recommended items (e.g., by watching a recommended movie, or by buying a recommended item), by providing a feedback on them (*User u_a feedback*), stored in a *Feedback* repository.

The feedback provided by the active user is then used by the system to update her/his user profile.

Limits at the State of the Art and Design Guidelines

In the previous section, we presented the state-of-the-art architecture of a content-based recommender system. We will now present the possible problems that might occur by employing it and provide design guidelines on how to improve it.

The possible problems that might occur will be presented through possible use cases/scenarios that might occur.

Scenario 1. The account of the active user is used by another person, who evaluates items that the user would have never evaluated (e.g., she/he buys items that the active user would have never bought). This would lead to the presence of noise in a user profile, since the *Structured Item Representation* of these incoherent items with respect to the user profile would be considered by the PROFILE LEARNER component, which would make them part of the *user u_a profile*, stored as it is in the *Profiles* repository, and employed in the recommendation process by the FILTERING COMPONENT. This would generate bad recommendations and the accuracy of the system would strongly be affected.

Scenario 2. The preferences of the active user change over time, but the oldest items that do not reflect the current preferences of the user, but positively evaluated by her/him, are still part of the user profile. A form of *aging* of the items in a user profile would allow the system to ignore such items after some time, but until that moment those items would represent noise. That noise might affect the system for a lot of time, since the aging process is usually gradual and the items age slowly. Again, this would affect the recommendation accuracy.

Scenario 3. If a mix of the two previous scenarios occurs and these type of problems are iterated over time, the system would reach the so-called *magic barrier*. As previously highlighted, the problem has been widely studied in the Collaborative Filtering literature, in order to identify and remove the noisy items based on their ratings. No work in the literature studied the magic barrier from a content-based point of view, so the state-of-the-art architecture previously presented is limited also from that perspective.

The three previously presented scenarios put in evidence that the architecture of a content-based system should be able to deal with the presence of incoherent items in the user profiling process, in order to avoid the previously aforementioned problems. Therefore, we will now present design guidelines on how to improve the state-of-the-art-art architecture of a system.

The first scenario highlighted the need for a system to detect how coherent is an item with the rest of the items that have been evaluated by a user, in order to detect the presence of noise. This could be done by comparing the content of the item (i.e., the *structured item representation*) with that of the other items evaluated by the user *user u_a training examples*.

Scenario 2 confirms the need for a system to evaluate the temporal correlation of an item with the rest of the items in the user profile. Indeed, if an item is too old and, as previously said, too different with respect the other items, it should be removed from a user profile.

Both the second and the third scenarios highlighted that the presence of noisy/incoherent items on a user profile should be reduced to a very limited amount of time. In particular, thanks to scenario 3 we know that these items should not be ignored gradually, but the system should be able to do a one-off removal. This would allow the filtering component to consider only items that are coherent with each other and with the preferences of the users.

In Section 5.2.2 will adopt these design guidelines to present an architecture that overcomes these issues.

5.2.2 Recommender Systems Architecture

Overview

This section proposes my novel architecture. The updated high-level architecture of the system is first proposed (Section 5.2.2), and in Section 5.2.2 are presented the details of the novel component that faces the open problems highlighted in the previous section. This part will close with a brief analysis that shows how this proposal fits with the development of a real-world system in the big data era (Section 5.2.2).

Proposed Solutions

This part of research first analyzes the state-of-the-art architecture of a content-based recommender system, then it will explore in detail the possible problems that might occur by employing it. Some design guidelines on how to enrich that architecture will be proposed, and a novel architecture, which allows the system to tackle the highlighted problems and improve the effectiveness of the recommendation process, will be presented.

Even though we will focus on the emerging application domain we previously mentioned (i.e., the semantics-aware systems), we will also show the usefulness of our proposal on classic content-based approach.

The scientific contributions coming from the thesis are now summarized:

- it will analyze the state-of-the-art architecture of a content-based recommender system to study, for the first time in the literature, what might happen in the recommendation process if incoherent items are filtered by the system;
- this is the first study in which the magic problem is studied in a content-based recommender system and from the architectural point of view;

- it presents design guidelines and a novel architecture, in order to improve the existing one and overcome the aforementioned issues;
- it will analyze the impact of the components we will introduce in the proposed architecture from a computational cost point-of-view.

Approach

High-Level Architecture Figure 5.2 presents an updated version of the state-of-the-art architecture illustrated in Section 5.2.1. The proposed architecture integrates a novel component named PROFILE CLEANER, with the name to analyze a profile and remove the incoherent items, before storing it in the *Profiles* repository. In order to solve the previous problems, the component should be able to remove an item if it meets the following two conditions:

1. the coherence/content-based similarity of the item with the rest of the profile is under a *Minimum Coherence* threshold value;
2. it is located in the first part of the user iteration history. Based on this requirement, an item is considered far from the user's preferences only when it goes up in the first part of the iterations (i.e., when the distance with the last evaluated item is higher than a *Maximum Temporal Distance* threshold).

By removing the incoherent old items, the FILTERING COMPONENT would consider only the real preferences of the users and the previously mentioned problems are solved. Indeed, by checking that both conditions are met, the system avoids removing from a profile the items that are diverse from those she/he previously considered, but that might be associated to a recent change in the preferences of the user.

Regarding scenario 1, if among a *user u_a training examples* there is an incoherent item evaluated by a third party, it would be detected by the component, since it receives it as an input. Regarding scenarios 2 and 3, by checking the temporal correlation of an item with the others in the user profile, the component would be able to remove an item as soon as it becomes old and incoherent, avoiding the problems related to the *aging* strategies (which might still be employed by the PROFILE LEARNER, but are not enough) and to the presence of too many incoherent items that would lead to the *magic barrier* problem.

Low-level Representation of the Profile Cleaner The Figure 5.3 inspects furthermore on the component introduced in this novel architecture, to present a low-level analysis and the subcomponents it should employ to accomplish its task.

As Figure 5.2 showed, the profile cleaner takes as input both an *item i* a user has evaluated (i.e., one of the *training examples* or of the *feedbacks* provided by a user) and a *user profile*.

The ITEMS COHERENCE ANALYZER subcomponent compares the structured representation of an item *i* with the rest of the user profile, in order to the detect the coherence/similarity of the item with the rest of the profile. If the *Structured Item Representation* involves semantic structures (e.g., Wordnet synsets), as the modern content-based systems do, several metrics can be employed to evaluate the

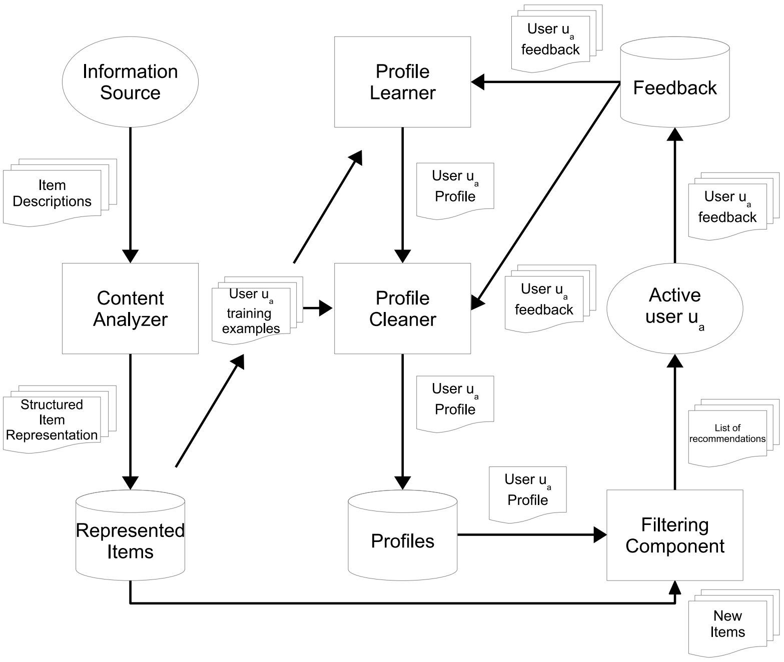

Figure 5.2: Architecture of a semantics-aware content-based recommender system.

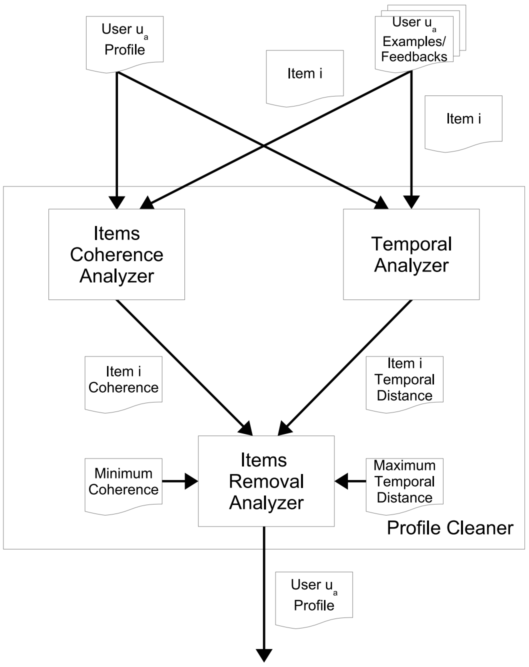

Figure 5.3: Architectural organization of the profile cleaner task.

semantic similarity between two structured representations that involve synsets. The state-of-the-art ones are the following five: i.e., *Leacock and Chodorow* [97], *Jiang and Conrath* [98], *Resnik* [99], *Lin* [100], and *Wu and Palmer* [101]. However, any type of similarity/coherence might be employed, even if no semantic information is available the item representation (e.g., TF-IDF). The output produced by the subcomponent is an *Item i Coherence* value, which will be later employed by the ITEMS REMOVAL ANALYZER subcomponent to decide if the item should be removed or not.

In parallel, the *Temporal Analyzer* subcomponent will consider how far was the evaluation of the considered with respect to that of the other items in the user profile (and especially the last evaluated one). The distance threshold might be defined as a fixed value, or by defining *regions* based on the chronology with which the items have been evaluated (e.g., remove an item if it was evaluated in the first two quartiles that contain the oldest items). The output is an *Item i Temporal Distance*, which will also be employed by ITEMS REMOVAL ANALYZER subcomponent.

The output of the two previously subcomponents is then handled by the ITEMS REMOVAL ANALYZER which also receive as input the *Minimum Coherence* and *Maximum Temporal Distance* thresholds, and decides if the considered item *i* should be removed from a user profile or not. The output produced by the subcomponent (and by the PROFILE CLEANER main component) is a cleaned *user u_a profile*, which does not contain the incoherent and oldest items.

Developing a System that Employs this Architecture It becomes natural to think that the introduction of a Profile Cleaner component, even if useful, might be lead to heavy tasks to be computed by the system. Indeed, the component has to deal with a comparison between each item and the rest of the user profile, and this similarity might involve semantic elements and measures, which are usually very heavy to compute. Given the widely-known *big data* problem that characterizes and affects each systems nowadays, this work will try to inspect on how to develop this component in real-world scenarios.

Indeed, the computation of the coherence of each of the new items with the rest of the user profile might distributed over different computers, by employing large scale distributed computing models like MapReduce. Moreover, this process can be handled in background by the system, since when a user evaluates a new item, it would hardly make any instant difference on the computed recommendations. Therefore, if it gets removed in a reasonable time and with a distributed approach, the employment of Profile Cleaner component would be both effective and efficient at the same time.

Moreover, we studied the structure of the Profile Cleaner component to let it run two subcomponents in parallel, so that even under this perspective the process can be parallelized and efficient.

In conclusion, we believe that even if we are introducing a possibly heavy computational process, the improvements in terms of accuracy and the structure of the component would overcome the complexity limits. Moreover, this complexity would also be efficiently dealt with the current technologies employed to face the big data problems (e.g., Hadoop's MapReduce).

Conclusions and Future Work

This work deals with the problems that might occur with the current way in which content-based recommender systems are engineered and designed.

Given the high impact that emerging aspects are having in research and real-world recommender systems, such as the introduction of the semantics in the filtering process and the so-called *magic barrier* problem, it analyzed the current architecture employed by a content-based recommender system and highlighted current limits. Indeed, it showed that a form of cleaning of the user profiles is necessary in order to overcome these limitations.

It then proposed an updated architecture, which was analyzed both from a high-level point of view and by inspecting on the component that allows a system to clean a profile. Moreover, it studied the application of this proposal in real-world scenarios, which would probably be characterized by the big data problem.

Future work will move from the software engineering perspective of our study, to develop real-world efficient implementations of this architecture (e.g., on a grid), in order to study its efficiency and effectives in scenarios characterized by the big data (e.g., the recommendations performed by an e-commerce website).

5.3 Implementation

5.3.1 Overview

Recommender systems usually produce their results to the users based on the interpretation of the whole historic interactions of these. This canonical approach sometimes could lead to wrong results due to several factors, such as a changes in user taste over time or the use of her/his account by third parties. This research proposes a novel dynamic coherence-based approach that analyzes the information stored in the user profiles based on their coherence.

The main aim is to identify and remove from the previously evaluated items those not adherent to the average preferences, in order to make a user profile as close as possible to the user's real tastes. The conducted experiments show the effectiveness of the proposed approach to remove the incoherent items from a user profile, in order to increase the recommendation accuracy.

5.3.2 Proposed Solution

The coherence of an item, with respect to the user profile, in literature is usually measured as the variance in the feature space that defines the item, typically based on the rating given by the users [102]. This is done by employing several metrics, such as the entropy, the mean value, or the standard deviation. Differently from the approaches at the state of the art, this research considers the semantic distance between the concepts expressed by each item in a user profile, and the concepts expressed by the other ones. This way to proceed presents a twofold advantage: firstly, it allow us to evaluate the coherence of an item in a more extensive way (by employing semantic concepts) w.r.t. a limited mathematical approach; secondly, it reduces the cause of the *magic barrier* problem. This happens because the assumption of the magic *barrier problem* is the presence of incoherent items in the user profiles. Considering that this approach removes them, keeping in the user profiles only those items that are coherent with each other, it allow to consider any observed improvement as real, instead that a mere side-effect (i.e., an overfitting).

To perform the task of removing semantically incoherent items from a user profile, this research introduces the *Dynamic Coherence-Based Modeling* (DCBM), an algorithm based on the concept of *Minimum Global Coherence* (MGC), a metric that allows us to measure the semantic similarity between a single item with the others within the user profile. Moreover, the algorithm takes into account two other factors, i.e., the position of each item in the chronology of the user choices, and the distance from the *mean value* of the *global similarity* (the term *global* identifies all the items in a user profile). These metrics allow us to remove in a selective way any item that could make the user's profiles non-adherent to their real tastes. The main idea is that the more information in the user profile is coherent, the more the recommendations based on this profile will be reliable. Differently from other strategies designed for specific contexts, this approach is able to operate in all scenarios. Through it, the process of evaluation of the items coherence has been moved from a domain based on rigorous mathematical criteria (i.e., variance of the user's ratings in the feature space), to a new semantic domain, which presents a considerable advantage in terms of evaluation flexibility.

In order to evaluate the capability of this approach to produce accurate user profiles, the DCBM algorithm is implemented into a state-of-the-art semantic-based recommender system [39], where the accuracy of the recommendations is evaluated. Since the task of the recommender system that predicts the interest of the users for the items relies on the information included in a user profile, more accurate user profiles lead to an improved accuracy of the whole recommender system. Experimental results show the capability of this approach to remove the incoherent items from a user profile, increasing the accuracy of recommendations.

The main contributions of this part of research can be summarized as following:

- introduction of a novel algorithm able to remove incoherent items from a user profile, with the aim to improve the recommendation accuracy;
- integration of this algorithm into a state-of-the-art recommender system, in order to improve its effectiveness and validate the proposed approach;
- verification on two datasets: a synthetic one that allows us to analyze the behavior of the proposed approach under different settings, and a real-world one used to compare the accuracy of the recommender system with and without the incoherent items removing.

5.3.3 Adopted Notation

Definition 5.1 (User preferences) *We are given a set of users $U = \{u_1, \dots, u_N\}$, a set of items $I = \{i_1, \dots, i_M\}$, and a set V of values used to express the user preferences (e.g., $V = [1, 5]$ or $V = \{like, dislike\}$). The set of all possible preferences expressed by the users is a ternary relation $P \subseteq U \times I \times V$. We denote as $P_+ \subseteq P$ the subset of preferences with a positive value (i.e., $P_+ = \{(u, i, v) \in P | v \geq \overline{v} \vee v = like\}$), where $\overline{v}$ indicates the mean value (in the previous example, $\overline{v} = 3$).*

Definition 5.2 (User items) *Given the set of positive preferences P_+, we denote as $I_+ = \{i \in I | \exists (u, i, v) \in P_+\}$ the set of items for which there is a positive preference, and as as $I_u = \{i \in I | \exists (u, i, v) \in P_+ \wedge u \in U\}$ the set of items a user u likes.*

Definition 5.3 (Item semantic description) *Let $BoW = \{t_1, \dots, t_W\}$ be the bag of words used to describe the items in I; we denote as d_i the binary vector used to describe each item $i \in I$ (each vector is such that $|d_i| = |BoW|$). We define as $S = \{s_1, \dots, s_W\}$ the set of synsets associated to BoW (that is, for each term used to describe an item, we consider its associated synset), and as sd_i the semantic description of i. The set of semantic descriptions is denoted as $D = \{sd_1, \dots, sd_M\}$ (note that we have a semantic description for each item, so $|D| = |I|$). The approach used to extract sd_i from d_i is described in detail in Section 5.3.5.*

Definition 5.4 (Semantic user model) *Given the set of positively evaluated items by a user I_u, we define a* semantic user model *M_u as the set of synsets in the seman-*

tic descriptions of the items in I_u. More formally, $M_u = \{s_w | s_w \in sd_m \wedge i_m \in I_u\}$.

Definition 5.5 (Item coherence) *An item $i \in I_u$ is* coherent *with the rest of the items in the user profile I_u, if the similarity between the semantic description sd_i of the item and the union of the semantic descriptions of the rest of the items (i.e., $M_u \setminus sd_i$) is higher than a threshold value.*

5.3.4 Problem Definition

Given a set of items I_u that a user likes, the objective is to extract a set $\overline{I_u} \subseteq I_u$, such that each item $i \in \overline{I_u}$ is *coherent* with the others.

5.3.5 Approach

As already highlighted during the description of the limits that affect the user profiling activity, individual profiles need to be as adherent as possible to the real tastes of the users, because they are used to predict their future interests. For this reason, this section proposes a novel approach defined *Dynamic Coherence-Based Modeling* (DCBM) able to find and remove the incoherent items within the user profiles, regardless of the profiling method chosen. The implementation on a recommender system of the DCBM is articulated in the following four steps:

1. **Data Preprocessing**: preprocessing of the text present in the items that compose a user profile, as well as of the text present in the items not yet considered, in order to remove the useless elements and the items with a user rating lower than the average;

2. **Semantic Similarity**: WordNet features are used to retrieve, from the preprocessed text, all the possible pairs between the WordNet synsets in the text of the items not evaluated and the synsets in the text of the user profile, keeping as a result only the pairs that have at least an element with the same *part-of-speech*, for which we measure the semantic similarity according to the *Wu and Palmer* metric;

3. **Dynamic Coherence-Based Modeling**: the items dissimilar from the average preferences of a user are identified by measuring the Minimum Global Coherence (MGC). Moreover, in accordance with certain criteria, the items that are more semantically distant from the context of a user's real tastes are removed from the user profile;

4. **Item Recommendation**: to perform the recommendation process, we sort the not evaluated items by their similarity with the user profile, and propose to a user a subset of those with the highest values of similarity.

Note that steps 1, 2, and 4 are followed by a state-of-art recommender system based on the semantic similarity [39], in which the novel Dynamic Coherence-Based Modeling (DCBM) algorithm (step 3) is integrated, in order to improve the user profile and increase the accuracy of the recommender system.

How each step works is described in detail in the following

Data Preprocessing

Before comparing the similarity between the items in a user profile, we need to follow several preprocessing steps.

The first step detects the correct *part-of-speech* (POS) for each word in the text; in order to perform this task, the *Stanford Log-linear Part-Of-Speech Tagger* [103] has been used.

The second step removes punctuation marks and *stop-words*, i.e., the insignificant words (such as adjectives, conjunctions, etc.) that represent noise in the semantic analysis. Several *stop-words* lists can be found on the Internet; this work used a list of 429 *stop-words* made available with the *Onix Text Retrieval Toolkit*[1].

The third step, after the determination of the lemma of each word using the Java API implementation for WordNet Searching (JAWS)[2], performs the so-called word sense disambiguation, a process where the correct sense of each word is determined, which permits us to evaluate the semantic similarity in precise way. The best sense of each word in a sentence was found using the Java implementation of the adapted Lesk algorithm provided by the *Denmark Technical University* (DTU) similarity application [104]. All the collected synsets form the set $S = \{s_1, \dots, s_W\}$ defined in Section 5.3.3.

The output of this step is the semantic disambiguation of the textual description of each item $i \in I$, which is stored in a binary vector sd_i; each element of the vector $sd_i[w]$ is 1 if the corresponding synset appears in the item description, and

[1] http://www.lextek.com/manuals/onix/stopwords.html

[2] http://lyle.smu.edu/ tspell/jaws/index.html

0 otherwise.

Semantic Similarity

Although the most used semantic similarity measures are five, i.e. *Leacock and Chodorow* [97], *Jiang and Conrath* [98], *Resnik* [99], *Lin* [100] and *Wu and Palmer* [101], and each of them evaluates the semantic similarity between two WordNet synsets, we calculate the semantic similarity by using the *Wu and Palmer*'s measure, a method based on the path lengths between a pair of concepts (WordNet synsets), which in the literature is considered to be the most accurate when generating the similarities [105, 39].

Given a set X of i WordNet synsets $x_1, x_2, ..., x_i$ that are related to an item description, and a set Y of j WordNet synsets $y_1, y_2, ..., y_j$ related to another item description, a set Q, which contains all the possible pairs between the synsets in the set X and the synsets in the set Y, is defined as in Equation 5.1.

$$Q = \left(\langle x_1, y_1\rangle, \langle x_1, y_2\rangle, \ldots, \left\langle x_i, y_j\right\rangle\right) \forall x \in X, y \in Y \tag{5.1}$$

In the next step, a subset Z of the pairs in Q (i.e., $Z \subseteq Q$) that have at least an element with the same POS is created (Equation 5.2).

$$Z = \{(x_i, y_j) | POS(x_i) = POS(y_j)\} \tag{5.2}$$

The metric measures the similarity between concepts in an ontology, as shown in Equation 5.3.

$$sim_{WP}\,(x, y) = \frac{2 \cdot A}{B + C + (2 \cdot A)} \tag{5.3}$$

Assuming that the *Least Common Subsumer* (LCS) of two concepts x and y is *the most specific concept that is an ancestor of both x and y*, where the concept tree is defined by the *is-a* relation, in Equation 5.3 we have that *A=depth(LCS(x,y))*, *B=length(x,LCS(x,y))*, *C=length(y,LCS(x,y))*. We can note that $B + C$ represents the path length from x and y, while A indicates the global depth of the path in the taxonomy.

The similarity between two items is defined as the sum of the similarity score for all pairs, divided by its cardinality (the subset Z of WordNet synsets with a common part-of-speech), as shown in Equation 5.4.

$$sim_{WP}(X, Y) = \frac{\sum_{(x,y)\in Z}^{n} sim_{WP}(x, y)}{|Z|} \tag{5.4}$$

This similarity metric is employed both by the proposed algorithm to compute the coherence of an item with the rest of the semantic user model, and by the recommendation algorithm to select and suggest items similar to those that the user prefers.

Dynamic Coherence-Based Modeling

For the purpose of being able to make effective recommendations to users, their profiles need to store only the descriptions of the items that really reflect their tastes.

In order to identify which items positively evaluated by a user ($i \in I_u$) do not reflect the user taste, in that they represent, for instance, the result of past wrong choices or the use by third parties of her/his account, the Dynamic Coherence-Based Modeling (DCBM) algorithm measures the *Minimum Global Coherence* (MGC) of each single item description with the set of other items present in her/his profile. In other words, through MGC, the most dissimilar item with respect to the other items is identified.

The Wu and Palmer similarity metric previously presented can be used to calculate the MGC, as shown in Equation 5.5 (sd_i denotes the semantic description of an item i, and $M_u \setminus sd_i$ indicates the semantic user model from which the synsets in sd_i have been removed).

$$MGC = \min_{i \in I_u} (sim_{WP}(sd_i, M_u \setminus sd_i)) \tag{5.5}$$

The basic idea is to isolate each individual item i in a user profile, semantically described by sd_i, and then measure the similarity with respect to the remaining items (i.e., the merging of the synsets of the rest of the items), in order to obtain a measure of its coherence within the overall context of the entire profile.

In other words, in order to individuate the most distant element from the general context of the evaluated items, we are exploiting a basic principle of the differential calculus, because the MGC value shown upon is nothing other than the *maximum negative slope*, which is calculated by finding the ratio between the changing on y axis and the changing on x axis. This is demonstrated in Theorem 5.1.

Theorem 5.1 *The Minimum Global Coherence coefficient corresponds to the maximum negative slope.*

Proof: P □ lacing on the x axis the user iterations in a chronological order, and on the y axis the corresponding values of GS (Global Similarity) calculated as $sim_{WP}(sd_i, M_u \setminus sd_i), \forall i \in I_u$, we can trivially calculate the slope value (denoted by the letter m), as shown in Equation 5.6.

$$m = \frac{\triangle y}{\triangle x} = \frac{f(x + \triangle x) - f(x)}{\triangle x} \tag{5.6}$$

The mathematics of differential calculus defines the slope of a curve at a point as the slope of the tangent line at that point. Since we are working with a series of points, the slope may be calculated not at a single point but between two points. Considering that for each current user iteration $\triangle x$ is always equal to 1 (in fact, for N user iterations we have that $1 - 0 = 1, 2 - 1 = 1, \ldots, N - (N - 1) = 1$), the slope value m will always be equal to $f(x + \triangle x) - f(x)$. As Equation 5.7 shows,

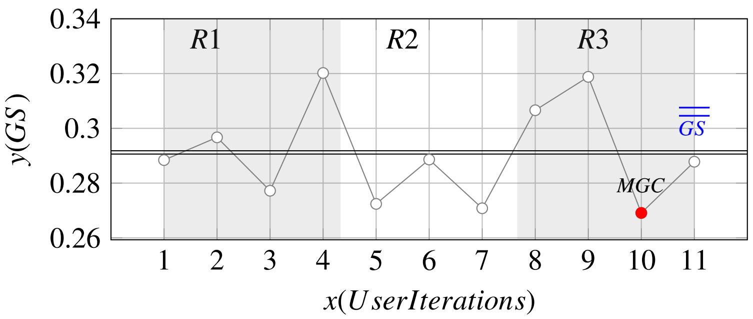

Figure 5.4: The maximum negative slope corresponds to the value of *MGC*

where $sim_{WP}(I_u)$ denotes $sim_{WP}(sd_i, M_u \setminus sd_i), \forall i \in I_u$, the maximum negative slope consequently corresponds to the value of *MGC*.

$$min\left(\frac{\Delta y}{\Delta x}\right) = min\left(\frac{sim_{WP}(I_u)}{1}\right) = MGC \tag{5.7}$$

In Figure 5.4, which displays the data reported in Table 5.1, we can see what we just said in a graphical way.

In order to avoid the removal of an item that might correspond to a recent change in the tastes of the user or an item not semantically distant enough from the context of the remaining items, the DCBM algorithm removes an item only if meets the following conditions:

1. it is located in the first part of the user iteration history. Based on this first requirement, an item is considered far from the user's tastes only when it goes up in the first part of the iterations. This condition is checked thanks

Table 5.1: User profile sample data

x	y	m
1	0.2884	+0.2884
2	0.2967	+0.0083
3	0.2772	-0.0195
4	0.3202	+0.0430
5	0.2724	-0.0478
6	0.2886	+0.0162
7	0.2708	-0.0178
8	0.3066	+0.0358
9	0.3188	+0.0122
10	0.2691	-0.0497
11	0.2878	+0.0187

to a parameter r, taken as input by the algorithm, which defines the *removal area*, i.e., the percentage of a user profile where an item can be removed. Note that $0 \leq r \leq 1$, so in the example in Figure 5.4, $r = \frac{2}{3} = 0.66$ (i.e., the element related to MGC value is located in the region $R3$, so it does not meet this first requirement);

2. the value of MGC must be within a tolerance range, which takes into account the *mean value* of the *global similarity* (as global we mean are the items in the user profile).

Regarding the first requirement, it should be noted that the regions extension is strongly related both to the type of items and to the frequency of fruition of these last, so it depends on the operative scenario.

With respect to the second requirement, we prevent the removal of items when they do not have a *significant* semantic distance with the remaining items. For this reason, we first calculate the value of the mean similarity in the context of the user profile, then we define a threshold value that determines when an item must be considered incoherent with respect to the current context. Equation 5.8 measures the mean similarity, denoted by $\overline{GS}$, by calculating the average of the *Global Similarity* (GS) values, which are obtained as $sim_{WP}(y_j, \sum y \in Y \setminus y_j), \forall y \in Y$.

$$\overline{GS} = \frac{1}{|I_u|} \cdot \sum_{i \in I_u} (sim_{WP}(sd_i, M_u \setminus sd_i)) \tag{5.8}$$

where $|I_u|$ represents the total number of items stored in the profile (in the case of sample data shown in Table 5.1, the $\overline{GS} = 0.2906$). Obtained this average value, we can proceed to define the condition ρ to be used to decide when an item has to be (1) or not to be (0) removed, based on a threshold value α, defined by adding to average value $\overline{GS}$ a certain tolerance (as shown in Equation 5.9, in the case in example we have defined the tolerance value as *one-eighth* of the average, i.e., $\alpha = \frac{\overline{GS}}{8}$).

$$\rho = \begin{cases} 1, & \text{if } MGC < (\overline{GS} - \alpha) \\ 0, & \text{otherwise} \end{cases} \tag{5.9}$$

Based on the above considerations, we can now define the Algorithm 1, used to remove the semantically incoherent items from a user profile. The algorithm requires as input the set I_u (i.e., the user profile), a parameter α used to define the accepted distance of an item from the average, and a removal area r used to define in which part of the profile an item should be removed. In step 3 we extract the set of synsets M_u (Definition 5.4) from the description of the items in the user profile I_u (Definition 5.2). Steps 4-6 compute the similarity between each couple of synsets that belong to the user profile. In step 7, the average of the similarities is computed, so that in steps 8-15 we can evaluate if an item has to be removed from a user profile or not. In particular, once an item m_i is removed from a profile in step 12, its associated similarity s is removed from the list S (step 13), so that MGC in step 9 can be set as the minimum similarity value after the item removal. In step 16, the algorithm returns the user profile I_u after all the items in the first part of the user profile with the have been removed.

Algorithm 1 DCBM Algorithm

```
Require: $I_u$=set of items in the user profile, $\alpha$=threshold value, $r$=removal area
procedure PROCESS($Y$)
    $N = |I_u|$
    $M_u = GetSynsets(I_u)$
    for each Pair p=($sd_i, M_u \setminus sd_i$) in $I_u$ do
        $S \leftarrow sim_{WP}(p)$
    end for
    $a = Average(S)$
    for each $s$ in $S$ do
        $MGC = Min(S)$
        $i = index(MGC)$
        if $i < r * n$ AND $MGC < (a + \alpha)$ then
            Remove($i$)
            Remove($s$)
        end if
    end for
    Return $I_u$
end procedure
```

Item Recommendation

After the user profile has been processed with the Algorithm 1, this step computes the semantic similarity with all the items not evaluated, and recommends to a user a subset of those with the highest similarity. As previously said, the amount of items to recommend is related to the operative context. In this study we chose to recommend a set of items equal to those in the test set, imagining a scenario in which the user requests a fixed set of items to consume (e.g., "Recommend me three movies I can watch on a Sunday afternoon").

5.3.6 Experiments

The experimental environment for this work is based on the Java language, with the support of Java API implementation for WordNet Searching (JAWS) previously mentioned.

In order to perform the evaluation, two different datasets have been used, one with real data and one with synthetic data. With the real data we estimate the $F_1 - measure$ increment (or decrement) of the proposed novel DCBM approach, compared with a recommender system at the state-of-art based on the semantic similarity [39]. We also used a set of synthetic data, in order to evaluate the proposed approach with different distributions of the incoherent items in the profile.

As highlighted throughout the thesis, the system presented in Section 5.3.5 performs the same steps as the reference one, with the introduction of the DCBM algorithm to remove the incoherent items from the user profile. Since all the steps in common between the two recommender systems are performed with the same algorithms, the comparison of the F_1-measure obtained by the two algorithms will highlight the capability of DCBM to improve the quality of the user profile and of the accuracy of a recommender system.

Regarding the first condition to meet (see Section 5.3.5) in order to remove the items from a user profile, in the experiments we divided the user iteration history into 10 equal parts, considering valid for the items removal only the firsts 9 parts (i.e., parameter $r = 0.9$)[3].

We have not compared the proposed approach with the classic *magic barrier* formulations based on rating coherence, because the DCBM algorithm is performed in a content based recommender system. Since content based approaches do not employ the ratings during the filtering, it would not be useful to consider a form of coherence based on them.

5.3.7 Real Data

The employed dataset is Yahoo! Webscope Movie dataset, described in Appendix B. Given the high sparsity that characterizes this dataset, a sample was extracted, by removing all the users who evaluated less that 17 items and all the items that have been evaluated by 13 users[4]. The final sample consists of 5070 users, 1647 items, and 153461 ratings. Since the algorithm considers only the items with a rating above the average, we selected only the movies with a rating $\geq$ 3, and randomly extracted 33% of them as a test set. In order to evaluate the performance of the proposed approach with this dataset, we use the performance measures precision and recall, which we combine to calculate the $F_1 - measure$ (described in Appendix B).

[3] The choice to divide the history into 10 parts was made based on the frequency of the ratings given by the users. This analysis is not presented to facilitate the reading of the thesis.

[4] These values have been chosen in order to have a dataset in which useful information about each user and item was available to make the predictions.

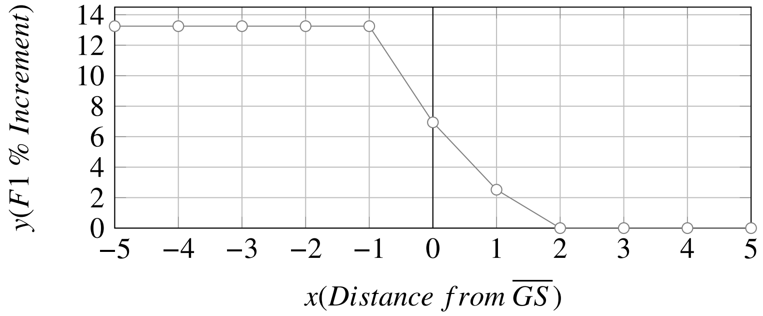

Figure 5.5: $F_1 - measure$ Per Cent Improvement

Strategy

For the experiments, it is necessary to set the value of α in Algorithm 1, which controls when an item is too distant from the average value $\overline{GS}$. We have tested some values positioned around the average value of the *Global Similarity* $\overline{GS}$ (see Equation 5.8). The values interval experimented is the half of the $\overline{GS}$ value (e.g., if $\overline{GS}$ = 0.4, the excursion of the values is from -0.2 to +0.2, so between 0.2 and 0.6). The interval of values is divided into 10 equal parts, labeled from -5 to 5.

Results

Figure 5.5 shows the per cent increment of $F_1 - measure$ of the proposed solution compared with the state-of-the-art recommender system.

From the results shown in the graph of Figure 5.5, we can observe how the average value of coherence (i.e., $\overline{GS}$, represented by the 0 on the x axis) represents the borderline between the improvement and worsening in terms of quality of the carried out recommendations. That happens because we obtain the maximum improvement in correspondence with the -1 value on the x axis, which represents the minimum distance from the mean value of coherence $\overline{GS}$. This improvement is progressively reduced as we approach the value of $\overline{GS}$, becoming zero almost immediately after this, because in this case we are removing from the user profile some items that are coherent with her/his global choices, essential to perform reliable recommendations.

To sum up, the graph in Figure 5.5 shows that the $F_1 - measure$ improvement increases until it becomes stable above certain values and presents no gain below others; this happens because we obtain an improvement only when the exclusion process involves items with a high level of semantic incoherence with respect to the others.

5.3.8 Synthetic Data

The set of *synthetic data* adopted is designed to simulate the real activity of a user at an online site that sells movies, regarding four different types of scenarios:

1. in the first case we simulate a user profile (composed by 10 items) with 2 incoherent items not related with a possible change of tastes (because they are positioned in the oldest part of her/his chronology);
2. also the second scenario presents a profile composed by 10 items with 2 of them incoherent, but one of these two is positioned in the last part of the history, representing a potential change in the user tastes;

3. in the third case we reproduce a scenario where in the next to last user iteration, 2 incoherent items were in the last part of user chronology (so they should not be removed), and in the current iteration the user chooses 2 further incoherent items. The aim of this experiment is to reproduce a scenario when the incoherent items are numerically consistent (4 out of 12 items), and for this reason we have to consider them not as a incoherent but as a clear change in the user tastes;

4. in the last scenario we test the performance of the proposed approach in a big user profile composed by 50 items (40 coherent items and 10 randomly placed incoherent items). The aim is to check how many of these will be properly identified and removed.

In order to avoid introducing a trivial criteria to discriminate incoherent items, we suppose that all the items are evaluated by the users with the maximum rating. Regarding the first and second requirement that we need to meet (see Section 5.3.5) in order to remove the items from a user profile, in the experiment we take in consideration several subdivisions of the user iteration history, considering valid for the items removal only the firsts $N - 1$ parts. We perform the experiments taking into account different distances (the *tolerance range* α) from the mean value of the *global similarity*.

Experimental Setup

The distance between the user iterations is an important aspect that we have to take into consideration to define the regions used to subdivide her/his profile. This happens because we consider as incoherent only the items stored in the first $N - 1$ regions, considering a change of user tastes the items stored in the last region.

In order to evaluate the proposed approach, it is necessary to set the value of α in Algorithm 1, which controls when an item is too distant from the average value $\overline{GS}$. We have tested some values positioned around the average value of the *Global Similarity* $\overline{GS}$ (see Equation 5.8). The values interval experimented is the 5 percent of the $\overline{GS}$ value (e.g., if $\overline{GS} = 0.5$, the excursion of the values is from -0.025 to $+0.025$, centered in $\overline{GS}$, then between 0.475 and 0.525). The interval of values is divided into 10 equal parts, labeled from -5 to 5.

Experimental Results

Here we present the results of the performed experiments, where we tested four different scenarios (case 1, 2, 3, and 4). In the first three cases (1, 2, and 3), the y axis of the graph represents the user profile, and its values are the items (squares inside the graph, in black those removed) progressively numbered (the lowest number denotes the oldest item evaluated by user). In the last case (4) the values in the y axis of the graph are the number of items removed from the user profile. In all cases, the values in the x axis represent the experimented values around the mean value of *global coherence*, in agreement with the criteria previously exposed.

- **Case 1.** In the first experiment we take in consideration a user profile composed by 10 items, and suppose that they have been evaluated by user in a temporal frame of one year. In this case is reasonable to subdivide the items in 5 regions, according to the frequency of the iterations.

  We have introduced 2 incoherent items at the second and fourth position of the user evaluation chronology. As we can observe in Figure 5.6, the items considered as incoherent (2nd and 4th in chronology) are correctly detected and removed by DCBM approach when the value on the x axis reaches the average value of *global coherence* (corresponding to the zero value on the x axis); when we stay away from this value, we either get many false positive (from 1 to 5), i.e., the items are incorrectly removed, or the obtained result does not change (from -5 to -1).

  It should be noted that in this case both items are located outside the *No-remove Region* (i.e., the last region in chronological order).

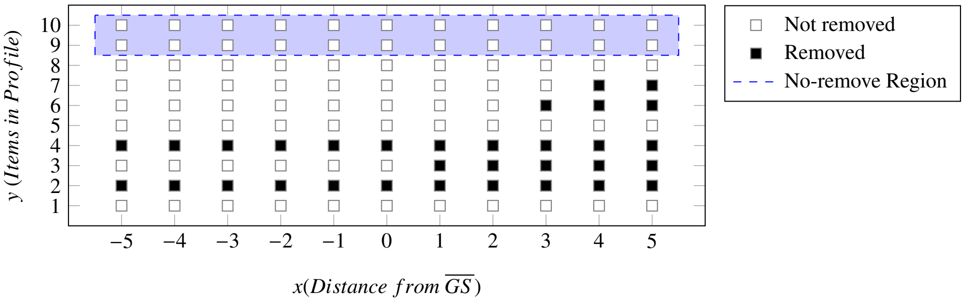

Figure 5.6: Removed Items in the Case 1

- **Case 2.** In the second experiment we process the same data of the previous example, but in this configuration we locate one of the 2 incoherent items inside the *No-remove Region* (in the ninth position of the chronology), and the second item just before it (in the seventh position of the chronology).

  As shown in Figure 5.7, only one of the two items considered as incoherent was removed by the DCBM approach (item 7), because the second (item 9) is evaluated as a change in the user tastes, and for this reason we can remove it only when will be outside the *No-remove Region*, as long as its value of coherence remains far from the mean value of *global coherence*.

  Also in this experiment, the correct items removal takes place only when the value on the x axis reaches the average value of *global coherence*.

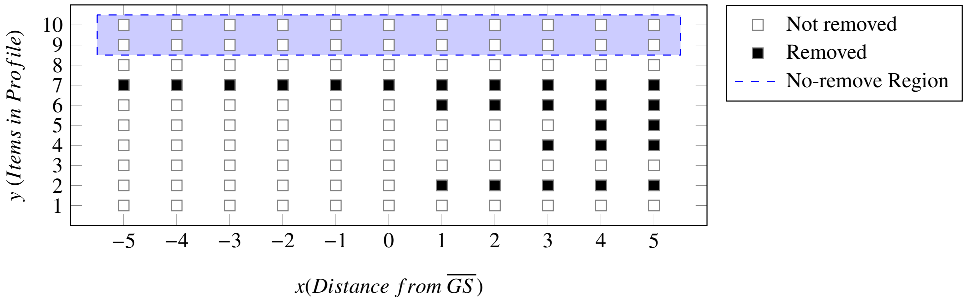

Figure 5.7: Removed Items in the Case 2

- **Case 3.** As introduced before, in this third case we evaluate a scenario where in the next to last user iteration, 2 incoherent items were in the last part of user chronology (*No-remove Region*), and then they were not removed, and in the current user iteration the user chooses 2 further incoherent items. To summarize, we have a total of 12 items stored in the user profile that is divided in 4 regions. At the end of the last user iteration we have a profile with 4 incoherent items stored in the 9th, 10th, 11th and 12th position of the chronology.

  As we can observe in Figure 5.8, in this particular configuration of the profile, none of the items recently evaluated by the user has been removed by the DCBM algorithm, even though they were distant from the value of *global coherence* previously estimated. This is because their numerical relevance has changed this value. The obtained result is that the only item removed (starting from the value zero on the x axis) is one of the items previously close to the value of *global coherence*.

What we observed is that the proposed approach is able to align the user profiles with the change in user tastes, when these are not related to scattered events, but rather represent a real change in the user preferences.

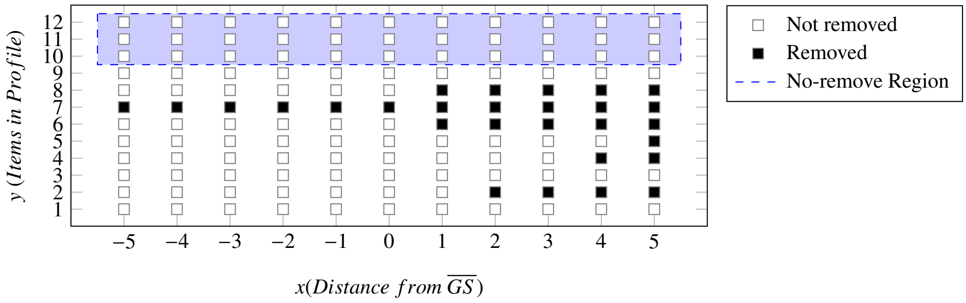

Figure 5.8: Removed Items in the Case 3

- **Case 4.** In this last case we want to test the performance of the DCBM approach related with a big user profile composed by 50 items. Through it, we want simulate the activity of an assiduous customer that evaluates many items. In this configuration it is reasonable to subdivide her/his profile in 10 regions, each containing 5 items. The test consists of introducing 10 incoherent items in a random position and check how many of these are properly identified and removed by the proposed algorithm.

  In a short, we have a profile composed of 40 coherent items and 10 incoherent items placed randomly. The results of the experiment are shown in Figure 5.9 where TP denotes the *True Positives* (i.e., the items correctly removed) and with FP the *False Positives* (i.e., the items incorrectly removed). Considering that 2 of the 10 items randomly placed were positioned within the *No-remove Region* (last 5 positions of the profile), we have to consider as the best possible result a number of 8 items removed (this upper limit is denoted by a dashed line in the graph of Figure 5.9).

Every experiment showed that the best value to use as *threshold* for the removal of an incoherent item is placed around the mean value of *global coherence*, because if we move away from it we get many false positives or no improvement.

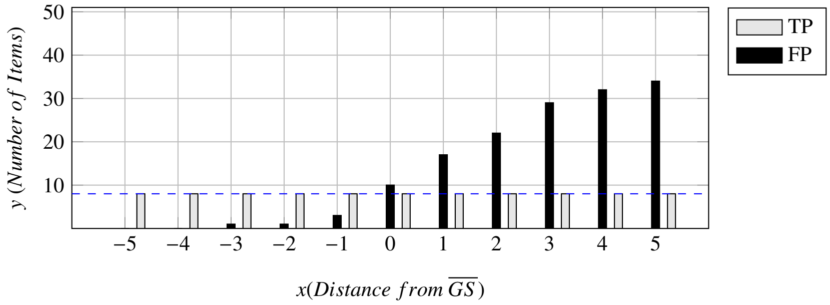

Figure 5.9: Number of Removed Items in the Case 4

5.3.9 Conclusions and Future Work

This part of my work proposes a novel approach to improve the quality of the user profiling, by taking into account the items related to a user, with the aim of removing those that do not reflect her/his real tastes. This is useful in many contexts, such as when the system does not allow the users to express her/his preferences or when the user decides not to make use of this option. If on the one hand the proposed approach conducts toward more accurate recommendations, on the other hand it reduces the number of items in the user profiles, thus the computational complexity. This last aspect represents a very important result, if we relate it with time-consuming approaches of recommendation, such as the semantic ones.

A further possible expansion might involve the use of a large amounts of data also related to contexts from each other as, for example, the scenario present on sales platforms that give access to very heterogeneous goods, in which we could operate in order to discover and process the semantic interconnections between different classes of items and methods to evaluate their semantic coherence during the user profiling activity.

Chapter 6

Decision Making Process

6.1 Introduction

In order to lead the potential buyers toward a number of well-targeted suggestions, related to the large amount of goods or services, a recommender system plays a determinant role, since it is able to investigate on the user preferences, suggesting to users the items that could be interesting. In order to identify these items, it has to *predict* that an item is worth recommending. Most of the strategies used to generate the recommendations are based on the so-called *Collaborative Filtering* (CF) approach, which is based on the assumption that users have similar preferences on a item if they have already rated other items in a similar way. As discussed in Chapter 2.3, the rating prediction has been highlighted in the literature as the core recommendation task, and recent studies showed its effectiveness also in improving classification tasks.

In recent years, the *latent factor models* have been adopted in CF approaches with the aim to uncover latent characteristics that explain the observed ratings. Among these last approaches, the state of the art is represented by SVD++, which exploits the so-called *latent factor model* and presents good performance in terms of accuracy and scalability. Although this approach provides excellent performance, it does not take into account the factor of popularity of the items that are recommended, risking to penalize its performance under certain circumstances. This can happen when the same score is given to multiple items, since not being able to discriminate them on the basis of their popularity, there is the risk to recommend those unpopular, which are less likely to be preferred by the users.

The popularity of the items is an aspect that has been widely studied in the recommender systems literature. While their ability to identify items of potential interest to the users has been recognized, some limitations have been highlighted. The most important of these is that the recommendations made according to popularity criteria are trivial, and do not bring considerable benefits neither to users, nor to those that offer them goods or services. This happens when the so-called *non-personalized model* are used, a naive approach of recommendation that does not take into account the user preferences, because it always recommends a fixed list with the most popular items, regardless of the target user. On the other hand, however, recommending less popular items adds novelty (and also serendipity) to the users, but usually it is a more difficult task to perform.

Another possible limitation that might occur when producing recommendations considering only the ratings is the fact that these approaches ignore the semantic relations between the words in the item descriptions. Therefore, thanks to the advent of the so-called Semantic Web, other strategies, based on semantic criteria, have also spread. The main advantage is their capability to interpret the users preferences in a non-schematic mode, helping to understand the concepts that are connected with a text, which can be used to determine the similarity between items, instead of merely using the single terms in their textual description.

By exploiting both the concepts of similarity and diversity, this part of the research introduces, initially at architectural level, and subsequently at application level, some novel approaches able to improve the performance of a recommender system.

6.2 Recommender Systems Performance

6.2.1 Overview

This part of my research is focused on the role that the *popularity* of the items plays in the recommendation process. If on the one hand, considering only the most popular items generates trivial recommendations, on the other hand, not taking in consideration the item popularity could lead to a non-optimal performance of a system, since it does not differentiate the items, giving them the same weight during the recommendation process. Therefore, there is the risk to exclude from the recommendations some popular items that would have a high probability of being preferred by the users, suggesting instead others that, despite meeting the selection criteria, have less chance to be preferred.

The proposed strategy aims to employ in the recommendation process new criteria based on the items' popularity, by introducing two novel metrics. The first metric evaluates the semantic relevance of an item with respect to the user profile, while the second metric measures how much it is preferred by users. Through a post-processing approach, these metrics are implemented in order to extend one of the most performing state-of-the-art recommendation techniques: SVD++. The effectiveness of this hybrid strategy of recommendation has been verified through a series of experiments, which show strong improvements in terms of accuracy w.r.t. SVD++.

6.2.2 Proposed Solutions

This part of my work aims instead to improve the recommendations produced by the SDV++ approach, by considering also the semantics behind the items and the items' popularity. This is done by employing two different strategies.

The first strategy involves a balanced use of two indices of item popularity: one based on the positive feedbacks of the users, and one based on the conceptual similarity of the textual description of the item with the descriptions of the other ones positively evaluated in the past.

The second strategy consists in the application of these two metrics within the boundaries of a recommendation list, generated through a state-of-the-art ap-

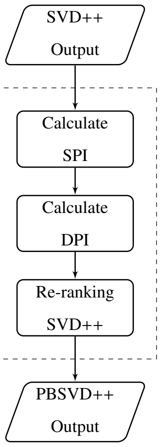

Figure 6.1: Approach Architecture

proach based on the *latent factor model* (the so-called SVD++ approach [49]), instead of using the entire dataset. This way of proceeding allows us to exploit the popularity metrics to perform a *fine-tuning* of the recommendations generated by a strategy at the state of the art, which does not take into account the items popularity, by improving the effectiveness of the generated recommendations.

In conclusion, the proposed metrics enhance the performance of SVD++, since they are able to consider the popularity factor during its ranking process, giving priority to the items that have a high probability of being preferred by the users. The block diagram in Figure 6.1 introduces the high-level architecture of the proposed approach.

The main contributions of this last part of my research are the following:

- definition of the Semantic Popularity Index (SPI), a metric able to evaluate the semantic popularity of an item, relatively to the items in a user profile;

- definition of the Domain Popularity Index (DPI), a metric able to evaluate the preferences of the users about an item;
- creation of the PBSVD++ algorithm, which extends the capabilities of SVD++, adding to it the capability to evaluate the item popularity.

6.2.3 Adopted Notation

We consider a set of users $U = \{u_1, \ldots, u_N\}$, a set of items $I = \{i_1, \ldots, i_M\}$, and a set V of values used to express the user preferences (e.g., $V = [1, 5]$ or $V = \{like, dislike\}$). The set of preferences expressed by the users is a ternary relation $P \subseteq U \times I \times V$. We denote as $P_+ \subseteq P$ the subset of preferences with a positive value (i.e., $P_+ = \{(u, i, v) \in P | v \geq \overline{v} \vee v = like\}$), where $\overline{v}$ indicates the mean value (in the previous example, $\overline{v} = 3$). Moreover, we denote as $I_+ = \{i \in I | \exists(u, i, v) \in P_+\}$ the set of items for which there is a positive preference, and as $np_{i,U} = |(u, i, v) \in P_+|, i \in I, \forall u \in U$ the number of positive preferences of the users in U for an item i. We also denote as $I_u = \{i \in I | \exists(u, i, v) \in P \wedge u \in U\}$ the set of items in the profile of a user u, and as $R_u = \{u \in U \wedge R \subseteq I\}$, the set of items i recommended to a user u. The set of items I without the items already evaluated by the user u (i.e., those in I_u) is denoted as $\hat{I}_u \subseteq I$. Let $BoW = \{t_1, \ldots, t_W\}$ be the bag of words used to describe the items in I; we define as $S = \{s_1, \ldots, s_W\}$ the set of synsets associated to BoW (that is, for each term used to describe an item, we consider the associated synsets), as sd_i the semantic description of i, and as $sd_{I,u}$ the semantic description of all items i in the profile of the user u. The set of semantic descriptions is denoted as $D = \{sd_1, \ldots, sd_M\}$ (we have a semantic description for each item, so $|D| = |I|$). The approach used to extract sd_i and $sd_{I,u}$ from d_i is described in detail in Section 6.2.5.

6.2.4 Problem Definition

We consider the function $f : U \times I \rightarrow V$, adopted to predict the ratings for the not evaluated items with the SVD++ recommender system. The aim is to define, for each item, a Semantic Popularity Index $SPI(i, u)$, able to evaluate the semantic relevance of each item $i \in \hat{I}_u$ with respect to the user profile I_u, and a Domain Popularity Index $DPI(i)$ that represents the popularity of the item with respect to the others in the dataset (in terms of positive evaluations given by the users to it). By defining a combined score α that involves both popularity indexes, the objective is to generate a list of recommended items i^* such that:

$$i^* = \underset{j \in \hat{I}_u}{\operatorname{argmax}} \, f(u, j) + \alpha \tag{6.1}$$

6.2.5 Approach

In this section we present the steps made to generate the recommendations based on the proposed *Popularity-based* SVD++ (PBSVD++) strategy, starting from the extraction of the WordNet synsets related to the textual description of the involved items, and ending with the implementation of the novel algorithm.

These operations can be grouped into the two following steps:

- *Text Preprocessing.* In the first step, we process the textual description of the items in order to remove the useless elements, before the subsequent operation of synset retrieving;

- *PBSVD++ Algorithm Definition.* In the second step, we define the PBSVD++ algorithm, through which we can alter the original ranking of the SVD++ recommendations, by employing the SPI and DPI criteria, formalized in the following Section 6.2.5.

Items Popularity

In the following, we introduce and formalize the two popularity indexes employed in the proposed approach.

Semantic Popularity Index. The Semantic Popularity Index (SPI) for an item $i \in I$, with $SPI \in [0, 1]$, is calculated as shown in Formula 6.2, where sd_i denotes the set of synsets extracted from the description of an item i to evaluate, and $sd_{I,u}$ the set of synsets extracted from the description of the items I in the profile of the target user u. It measures the conceptual similarity between these sets, and represents the *precision* (Appendix B), calculated for the item in the context of the user profile. SPI represents an important indicator, since through it we can estimate the level of (semantic) similarity of an item with the user tastes, represented in terms of items positively evaluated in the past.

$$SPI(i, u) = \frac{|sd_i \cap sd_{I,u}|}{|sd_i|} \tag{6.2}$$

Domain Popularity Index. The value of the Domain Popularity Index (DPI) for an item $i \in I$, with $DPI \in [0, 1]$, represents the number $np_{i,U}$ of positive preferences expressed by all users U for the item i. It is calculated as shown in Formula 6.3. DPI is also an important indicator, because it extends the local information provided by SPI (related to the single users), providing a global measure of the preferences expressed for an item by all users.

$$DPI(i, U) = \frac{np_{i,U}}{\sum\limits_{\forall j \in I} np_{j,U}} \tag{6.3}$$

Text Preprocessing

Motivated by the fact that exploiting a taxonomy for categorization and classification purposes is an approach recognized in the literature [79, 53, 54], in order to calculate the semantic correlation between the items we decided to exploit the functionalities offered by the WordNet environment. Before extracting the WordNet synsets from the text that describes each item, we need to follow several preprocessing steps. The first step is to detect the correct *Part-Of-Speech* (POS) for each word in the text. In order to perform this task, we have used the *Stanford Log-linear Part-Of-Speech Tagger* [103]. In the second step we remove punctuation marks and *stop-words*, which represent noise in the semantic analysis. In the third step, after we have determined the lemma of each word using the Java API implementation for WordNet Searching JAWS[1], we perform the so-called word sense disambiguation, a process where the correct sense of each word is determined, which permits us to individuate the appropriate synset in a precise way. The best sense of each word in a sentence was found using the Java implementation of the adapted Lesk algorithm provided by the *Denmark Technical University* similarity application [104]. All the collected synsets form the set $S = \{s_1, \ldots, s_W\}$ defined in Section 6.2.3.

The output of this step is the semantic disambiguation of the textual description of each item $i \in I$, denoted as sd_i. For each user, we also extract an additional vector $sd_{I,u}$, which contains all the synsets that characterize the items she/he positively evaluated.

PBSVD++ Algorithm

We exploit the SPI and DPI metrics (explained in Section 6.2.5), in order to modify the result of the SVD++ approach, in accord with these two parameters. These two metrics are implemented in the Algorithm 2, where we merge them in a unique value α, generated by their product.

Given a set of recommendations R_u, addressed to a user $u \in U$, the final rating $\rho_{i,u}$ assigned to each item $i \in R_u$ by the proposed algorithm, is composed by

[1] http://lyle.smu.edu/ tspell/jaws/index.html

the $rating_{i,u}$ calculated through the SVD++ approach, normalized in a continuous range from 0 to 1, and denoted as $STD(i,u)$, added to the product of the two indices SPI and DPI (also normalized in a continuous range from 0 to 1), as shown in Formula 6.4. The final rating assigned to an item is then in the range from 0 to 2.

$$\begin{aligned} \rho_{i,u} &= STD(i,u) + \left(\frac{SPI(i,u)}{\sum\limits_{\forall j \in R_u} SPI(j,u)} \cdot \frac{DPI(i,U)}{\sum\limits_{\forall j \in R_u} DPI(j,U)} \right) \\ with\ STD(i,u) &= \frac{rating_{i,u}}{\sum\limits_{\forall j \in R_u} rating_{j,u}} \end{aligned} \tag{6.4}$$

The new rating $\rho_{i,u}$, assigned to an item i for a user u, takes into account, in a balanced way, both its semantic and domain popularities, and this produces a substantial change in the canonical SVD++ ranking during the recommendation process, changing the performance of the recommender system.

Algorithm 2 implements the operations described above. It takes as input the training set s (used by the SVD++ approach, in step 3, to build the latent factor model), the user u to whom address the recommendations, and the number n of these. After the number x of potential items to recommend to the user u has been set (step 2), we calculate through the standard SVD++ approach, for the user u, a set I of x recommendations based on the training set s (step 3). In the steps from 5 to 11, we select from I only the elements i that are candidates for the recommendations based on the proposed approach.

They are those items in which a modification of the score, by adding to the original rating of SVD++ the value of α (parameter calculated in the step 14, whose value is in the range from 0 to 1), could alter the rank proposed by SVD++. For this reason, the candidates are only the items to which, adding at most 1, we get a value higher than that of the item with the maximum SVD++ score (i.e., the first element i_0). We use this process also to calculate (in steps 8 and 9) the sum of the SPI and DPI weights, related to all the items $i \in I$. Starting with this set R of candidate items, in the steps from 12 to 18, we alter the SVD++ score of each item $i \in I$, following Formula 6.4, after which we return a list L of n recommendations, composed by the items with the higher score.

6.2.6 Experiments

In this section, after the definition of the experimental environment and of the adopted datasets' characteristics, we describe the strategy and metrics adopted, concluding with the presentation and discussion of the experimental results.

Experimental Setup

The environment for this work is based on the Java language, with the support of Java API implementation for WordNet Searching (JAWS) to perform the semantic analysis, and the support of Apache Mahout[2] Java framework to implement the state-of-the-art approach that we compare the proposed approach with. In order to evaluate the proposed strategy, we perform a series of experiments on

[2] https://mahout.apache.org

Algorithm 2 PBSVD++

```
Require: $s$=Training set, $u$=User, $n$=Recommendations
Ensure: $L$ = List of $n$ recommendations
procedure GetPbSvdRecs($s$,$u$,$n$)
    x=GetNumOfNotEvaluatedItems($u$)
    I=GetSvdRecs($s$,$u$,$x$)
    t1=0, t2=0
    for each $i$ in $I$ do
        if $(SvdRating(i) + 1) > SvdRating(i_0)$ then
            $R \leftarrow i$
            $t1$+=$GetSPI(i)$
            $t2$+=$GetDPI(i)$
        end if
    end for
    for each $r$ in $R$ do
        rating=(SvdRating($r$)/SumAllSvdRatings($R$))
        $\alpha = (GetSPI(r)/t1) \cdot (GetDPI(r)/t2)$
        SetNewRating($r$,$rating$+$\alpha$)
    end for
    $L = GetRecsDescOrdered(R, n)$
    Return $L$
end procedure
```

three different real-world datasets, extracted by two standard benchmarks for recommender systems: Yahoo! Webscope R4 and Movielens 10M (both described in Appendix B). Using the script provided with the Movielens 10M dataset, we split up the whole dataset in two different datasets with exactly 10 ratings per user in the test set. Both training sets are composed by $69,878$ users ($|U|$), and $9,301,274$ ratings ($|P|$), with $10,667$ movies/items ($|I|$) in the first one, and $10,676$ movies/items ($|I|$) in the second one. Each test dataset contains $69,878$ users ($|U|$), and $698,780$ ratings ($|P|$), with $3,326$ movies/items ($|I|$) in the first one, and $5,724$ movies/items ($|I|$) in the second one. From each of these datasets, we take in account a subset of $20,000$ users.

Strategy

We compare the proposed recommendation strategy with the state-of-the-art approach SVD++. The Mahout framework, used to implement it, in addition to the training set requires two parameters: the number of target features and the number of training steps to run. The first parameter would be equivalent to the number of involved genres, thus we have set this value to 20 for the Yahoo dataset, and to 18 for the Movielens datasets. Regarding the second parameter, we use the value 15, as indicated in the reference paper of the SVD++ algorithm [49]. In order to compare the results of the two approaches of recommendation (i.e., the proposed approach based on the PBSVD++ algorithm, and the canonical one, based on SVD++), we calculate the *F1-Measure* metric, presented in described in Appendix B, for each group of n performed recommendations (denoted as $@n$, with $n = \{2, 4, \ldots, 20\}$), subtracting from the values obtained by the proposed approach those obtained by SVD++. In this way, a positive value denotes that the proposed approach improves the standard one, while a negative value denotes that the proposed approach worsens the standard one. A zero value means that the results are identical (i.e., proposed and standard approaches report the same performance).

Denoting as X_n the set of n recommendations generated by the proposed strategy, as Y_n the set of n recommendations generated by the canonical SVD++ strategy, and as Z_n the set of n real user preferences stored in the testset, we define the measure shown in Equation (6.5).

$$F1\text{-}variation@n = F1\text{-}Measure@n(X_n, Z_n) - F1\text{-}Measure@n(Y_n, Z_n) \tag{6.5}$$

Results

Here, we report the results of the experiments.

Performance Overview and Details: the result presented in Figure 6.2 shows the general performance of the proposed strategy in the context of the three considered real-world datasets. It indicates the percentage of times in which we have done better, or have done worse than SVD++ (respectively, B and W). The overall results show the good performance of the proposed approach with all three datasets.

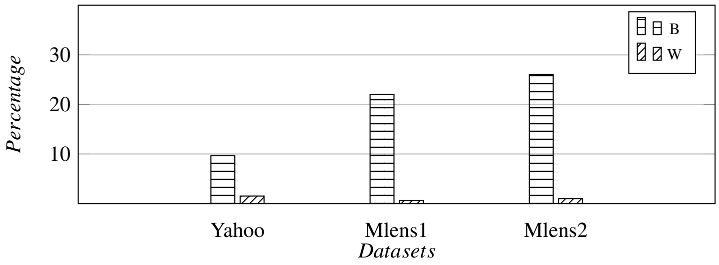

Figure 6.2: General performance

In the second set of experiments we compare the performance of a recommender system where we have implemented the PBSVD++ algorithm, with those of the canonic recommender system based on the SVD++ algorithm. We evaluate the results in terms of *F1-variation@n*, as described in Appendix B.

As we can observe in Figure 6.3, the results are quite similar for all three considered datasets. They show that the proposed strategy outperforms the canonical one, except when we test the maximum number of recommendations (i.e., 20). This is an obvious aspect, since the algorithm PBSVD++ operates in the domain of the SVD++ recommendations, recalculating their ratings: therefore, when we consider the entire domain, the results of SVD++ and PBSVD++ are always identical.

6.2.7 Conclusions and Future Work

The performed experiments, presented in Section 6.2.6, prove that the proposed strategy, based on the novel PBSVD++ algorithm, is able to improve the results

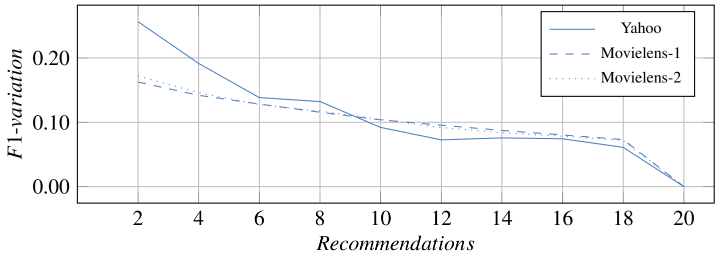

Figure 6.3: F1-Measure@n

of a canonical recommender system based on the SVD++ algorithm. As we can observe, this happens with any number of recommendations, except the case in which the maximum number of these is generated, for the obvious reason explained in the previous section. When evaluating these results, we can observe that the maximum value of positive variation for the metric is 1 (which represents a 100% improvement w.r.t. SVD++).

Considering that we are confronted with a strategy of recommendation to the state of the art as SVD++, that offers a little margin of improvement, the results obtained can be considered highly satisfactory, also considering that we never did worse than SVD++. This proves that is possible to improve a state-of-the-art approach such as SVD++, by using its output as an input domain, in order to perform a *fine-tuning* based on the popularity of the involved items.

Concluding, it should be noted that, although the proposed approach outperforms SVD++ in the entire range of recommendations, it produces the best results with a few number of them. This represents an important aspect, considering the difficulty for a recommender system to make correct predictions, by generating few recommendations.

In future work, we will extend the proposed approach, by adding new metrics able to evaluate the item popularity, in the context of systems that operate within more than one domain of goods/services, trying to parametrize both the popularity aspect of each item, and their interconnections between different operative domains. We will also study the introduction of others metrics of popularity, e.g., based on the geographic or demographic information.

On the Role of Similarity and Diversity in User Segmentation Systems

Preface

Similarity and diversity are both involved in this part of the research, in order to perform a non-trivial user segmentation. Such operation is performed by using a series of novel binary filters, which allow us to evaluate the user preferences in terms of classes of items, instead that in terms of single items.

Chapter 7

Latent Space Discovering

7.1 Introduction

In the literature it has been highlighted that classic approaches to segmentation (like k-means) cannot take into account the semantics of the user behavior, while those approaches that take in account this aspect, have to face several well-known open problems.

One of these problems is the *Reliability of a semantic query analysis* (Chapter 3.2.2), because in the literature it has been highlighted that half of the time the users need to reformulate their queries, in order to satisfy their information need. Another important open problem is the so-called *Preference stability*, discussed in Chapter 3.2.4, related with the fact that there are domains like movies in which the preferences tend to be stable over time. The last considered open problem that has to be faced in this research area is the *interpretability of a segment* (Chapter 3.2.3), considering that easily understandable approaches generate ineffective segments, and that more complex ones are accurate but not easy to use in practice, generates an important gap in this research area.

7.2 Overview

A recommender systems process is aimed to generate suggestions for items that might interest the users. It is a process usually performed at the level of a single item (i.e., for each item not evaluated by a user), based on the rating given by similar users for that item, or for an item with similar content. This leads to the so-called *overspecialization/serendipity* problem, in which the recommended items are trivial and users do not come across surprising items.

The performed research first shows that the preferences of the users are actually distributed over a small set of classes of items, leading the recommended items to be too similar to the ones already evaluated. It also introduces a novel representation model, named *Class Path Information (CPI)*, able to express the current and future preferences of the users in terms of a ranked set of classes of items. This approach to user preferences modeling is based on a semantic analysis of the items evaluated by the users, in order to extend the ground truth and predict where the future preferences of the users will go.

Experimental results show that the proposed approach, by including in the *CPI* model the same classes predicted by a state-of-the-art recommender system, is able to accurately model the preferences of the users in terms of classes and not in terms of single items, allowing recommender systems to suggest non trivial items.

7.3 Proposed Solutions

This part of my research tackles the problem of *defining a semantic behavioral targeting approach, such that the sources of information used to build it are reliable, the generated user segmentation is not trivial and it is easily interpretable*. In order to solve the problem of using reliable sources of information, our proposal is based on a semantic analysis of the description of the items positively evaluated by the users. The choice to start from items with a positive score was made since it is necessary to start from a knowledge-base that accurately describes what the users like, so that our approach can employ the semantics to detect latent information and avoid preference stability.

The approach first defines a binary filter (called *semantic binary sieve*) for each class of items that, by analyzing the description of the items classified with the class, defines which words characterize it. In order to characterize and detect more complex targets, we are going to define an algorithm that takes as input a set of classes that characterize the ads that have to be proposed to the users and a set of boolean operators. The algorithm combines the classes with the operators by means of a boolean algebra, and creates the binary filters that characterize the combined classes. Then we consider the words (that as we will explain later, are actually particular semantic entities named *synsets*) that describe the items evaluated by a user, and use the previously created filters to evaluate a *relevance score* that indicates how relevant is each class of items for the user. The relevance scores of each user are filtered by the segmentation algorithm, in order to return all the users characterized by a specified class or set of classes.

By selecting segments of users who are semantically related to the classes specified by the advertisers, we avoid considering only the users who evaluated items of that class; this allows our approach to overcome the open problems previously mentioned, related to preference stability and to the triviality of a segmentation generated by considering the evaluated items. Moreover, by defining the semantic binary sieves that characterize each class and the relevance scores that characterize each user, we avoid the interpretability issues that usually affect the user segmentation; indeed, each class of items is described by thousands of features (i.e., the words that characterize it), but this complexity is hidden to the advertiser, which is only required to specify the users she/he wants to target (e.g., those whose models are characterized by *comedy AND romantic* movies).

Considering that the evaluation of the users for the items offered in a context of e-commerce, are usually thousands or millions, the proposed approach represents an efficient strategy to represent in a compact way the information related to these big amounts of data.

The main contributions of this part of research can be summarized as following:

- we introduce a novel data structure, called *semantic binary sieve* to semantically characterize each class of items;

- we present the first semantic user segmentation approach based on reliable sources information; with respect to the state-of-the-art approaches that are based on the semantic analysis of the queries issued by the users, we perform a semantic analysis on the description of the items positively evaluated by the users;

- we overcome the overspecialization issues caused by preference stability by building a model for each user that considers a user as interested in a class of items if the items she/he evaluated are semantically related with the words that characterize that class;

- we present a boolean algebra that allows to specify in a simple but punctual way the interests that the segment should cover; the algebra, along with the built models, will avoid the interpretability issues that usually characterize the segmentations built with several features;

- we perform five sets of experiments on a real-world dataset, with the aim to validate our proposal by analyzing the different ways in which the classes can be combined through the algebra. The generated segments will be evaluated by comparing them with the topic-based segmentation (as several state-of-the-art approaches do), based on the real choices of the user.

7.4 Adopted Notation

Definition 7.1 (User preferences) *We are given a set of users $U = \{u_1, \ldots, u_N\}$, a set of items $I = \{i_1, \ldots, i_M\}$, and a set V of values used to express the user preferences (e.g., $V = [1, 5]$ or $V = \{like, dislike\}$). The set of all possible preferences expressed by the users is a ternary relation $P \subseteq U \times I \times V$. We denote as $P_+ \subseteq P$ the subset of preferences with a positive value (i.e., $P_+ = \{(u, i, v) \in P \mid v \geq \overline{v} \vee v = like\}$), where $\overline{v}$ indicates the mean value (in the previous example, in which $V = [1, 5]$, $\overline{v} = 3$).*

Definition 7.2 (User items and classes) *Given the set of positive preferences P_+, we denote as $I_+ = \{i \in I \mid \exists(u, i, v) \in P_+\}$ the set of items for which there is a positive preferences, and as $I_u = \{i \in I \mid \exists(u, i, v) \in P_+ \wedge u \in U\}$ the set of items a user u likes. Let $C = \{c_1, \ldots, c_K\}$ be a set of* primitive classes *used to classify the items; we denote as $C_i \subseteq C$ the set of classes used to classify an item i (e.g., C_i might be the set of genres that a movie i was classified with), and with $C_u = \{c \in C \mid \exists(u, i, v) \in P_+ \wedge i \in C_i\}$ the classes associated to the items that a user likes.*

Definition 7.3 (Item semantic description) *Let $BoW = \{t_1, \ldots, t_W\}$ be the bag of words used to describe the items in I; we denote as d_i the binary vector used to describe each item $i \in I$ (each vector is such that $\mid d_i \mid = \mid BoW \mid$). We define as $S = \{s_1, \ldots, s_W\}$ the set of synsets associated to BoW (that is, for each word used to describe an item, we consider its associated synset), and as sd_i the semantic* *description of i. The set of semantic descriptions is denoted as $D = \{sd_1, \dots, sd_M\}$ (note that we have a semantic description for each item, so $\mid D \mid = \mid I \mid$). The approach used to extract sd_i from d_i is described in detail in Section 5.3.5.*

Definition 7.4 (Semantic Binary Sieve) *Let $D_c \subseteq C$ be the subset of semantic descriptions of the items classified with a class $c \in C$ (i.e., $D_c = \{sd_i \mid c \in C_i\}$). We define as* Semantic Binary Sieve (SBS)*, a binary vector b_c that contains which synsets characterize that class. The algorithm to build a semantic binary sieve is given in Section 7.6.3.*

Definition 7.5 (Boolean class) *Given the set of classes C and a set of boolean operators $\tau = \{\wedge, \vee, \neg\}$, a* boolean class *is a subset of Q classes $C_Q \subseteq C$ combined through a subset of boolean operators $\tau_Q \subseteq \tau$. A boolean class is represented as a semantic binary sieve that defines which synsets characterize the combined classes. The algorithm to build the semantic binary sieve of a boolean class is also given in Section 7.6.3.*

Definition 7.6 (User target) *Given a set of users U and a (boolean) class c_q, a user target is a subset of users $T \subseteq U$ whose positively evaluated items I_u are semantically related to the items that belong to c_q.*

7.5 Problem Definition

Given a set of positive preferences P_+ that characterizes the items each user likes, a set of classes C used to classify the items (possibly combined with a set of boolean operators τ), and a set of semantic descriptions D, our first goal is to assign a relevance score $r_u(c)$ for each user u and each class c, based on the semantic descriptions D. The objective of our approach is to define a function $f : C^K \times \tau \rightarrow U$ that, given a (boolean) class, returns a set of users (user target) $T \subseteq U$, such that $\forall u \in T, r_u(c) \geq \varphi$ (where φ indicates a threshold that defines when a score is relevant enough for the user to be included in the target).

7.6 Approach

In this section we present our strategy, which performs a semantic analysis of the descriptions of the items the users like, in order to model both the users and the classes, and perform the semantic segmentation on the user set. Our approach performs five steps:

1. **Text preprocessing**: processing of the textual information related to all the items, in order to retrieve the synsets;

2. **User Modeling**: creation of a model that contains which synsets are present in the items a user likes;

3. **Semantic Binary Sieve definition**: creation of the *Semantic Binary Sieves* (SBS), i.e., a series of binary filters able to estimate which synsets are relevant for a class; a class can either be a class with which an item was classified, or a *boolean class* that combines primitive classes through boolean

operators (as primitive classes we mean the native classification of the items present in the used dataset);

4. **Relevance score definition**: generation of a relevance score that allows us to weight the user preferences in terms of classes;

5. **User Targeting**: selection of the users characterized by a specified set of classes.

In the following, we will describe in detail how each step works.

7.6.1 Text Preprocessing

Before extracting the WordNet synsets from the text that describes each item, we need to follow several preprocessing steps. The first step is to detect the correct *Part-Of-Speech* (POS) for each word in the text; in order to perform this task, we have used the *Stanford Log-linear Part-Of-Speech Tagger* [103]. In the second step we remove punctuation marks and *stop-words*, which represent noise in the semantic analysis (in this work we have used a list of 429 *stop-words* made available with the *Onix Text Retrieval Toolkit*[1]). In the third step, after we have determined the lemma of each word using the Java API implementation for WordNet Searching JAWS[2], we perform the so-called word sense disambiguation, a process where the correct sense of each word is determined, which permits us to individuate the appropriate synset. The best sense of each word in a sentence was

[1] http://www.lextek.com/manuals/onix/stopwords.html

[2] http://lyle.smu.edu/ tspell/jaws/index.html

found using the Java implementation of the adapted Lesk algorithm provided by the *Denmark Technical University* similarity application [104]. All the collected synsets form the set $S = \{s_1, \ldots, s_W\}$. The output of this step is the semantic disambiguation of the textual description of each item $i \in I$, which is stored in a binary vector ds_i; each element of the vector $ds_i[w]$ is 1 if the corresponding synset is a part of the item description, and 0 otherwise.

7.6.2 User Modeling

For each user $u \in U$, this step considers the set of items I_u she/he likes, and builds a user model m_u that describes which synsets characterize the user profile (i.e., which synsets appear in the semantic description of these items). Each model m_u is a binary vector that contains an element for each synset $s_w \in S$. In order to build the vector, we consider the semantic description ds_i of each item $i \in I_u$ for which the user expressed a positive preference. In order to build m_u, this step performs the following operation on each element w:

$$m_u[w] = \begin{cases} 1, \ if \ ds_i[w] = 1 \\ m_u[w], \ otherwise \end{cases} \tag{7.1}$$

This means that if the semantic description of an item i contains the synset s_w, the synset becomes relevant for the user, and we set to 1 the bit at position w in the user model m_u; otherwise, its value remains unaltered. By performing this operation for all the items $i \in I_u$, we model which synsets are relevant for the user.

The output of this step is a set $M = \{m_1, \ldots, m_N\}$ of user models (note that we have a model for each user, so $| M |=| U |$).

7.6.3 Semantic Binary Sieve Definition

Given a set of classes C, in this step we define a binary vector, called *Semantic Binary Sieve (SBS)*, which describes which synsets characterize each class. Moreover, we are going to present an approach to build the *boolean classes* previously defined, i.e., a semantic binary sieve that describes multiple classes combined through a set of boolean operators $\tau = \{\wedge, \vee, \neg\}$.

Therefore, four types of semantic binary sieves that can be defined:

1. ***Primitive class-based* SBS definition.** Given a primitive class of items c_k, this operation creates a binary vector that contains the synsets that characterize the description of the items classified with c_k;

2. ***Interclass-based* SBS definition.** Given two classes c_k and c_q, we combine the SBSs of the two classes with an *AND* operator, in order to build a new semantic binary sieve that contains the synsets that characterize both the classes;

3. ***Superclass-based* SBS definition.** Given two classes c_k and c_q, we combine the SBSs of the two classes with an *OR* operator, in order to build a new semantic binary sieve that merges their synsets;

4. ***Subclass-based*** **SBS definition.** Given two classes c_k and c_q, we use the SBS of c_q as a binary negation mask on the SBS of c_k, in order to build a new semantic binary sieve that contains the synsets that characterize the first class but do not characterize the second.

Primitive class-based SBS Definition

For each class $c_k \in C$, we create a binary vector that stores which synsets are relevant for that class. These vectors, called *Semantic Binary Sieves*, will be stored in a set $B = \{b_1, \ldots, b_K\}$ (note that $\mid B \mid = \mid C \mid$, since we have a vector for each class). Each vector $b_k \in B$ contains an element for each synset $s_w \in S$ (i.e., $\mid b_k \mid = \mid S \mid$). In order to build the vector, we consider the semantic description ds_i of each item $i \in I_+$ for which there is a positive preference, and each class c_k with whom i was classified. The binary vector b_k stores which synsets are relevant for a class c_k, by performing the following operation on each element $b_k[w]$ of the vector:

$$b_k[w] = \begin{cases} 1, \; if \; ds_i[w] = 1 \wedge i \in c_k \\ b_k[w], \; otherwise \end{cases} \tag{7.2}$$

In other words, if the semantic description of an item i contains the synset s_w, the synset becomes relevant for each class c_k that classifies i, and the semantic binary sieve b_k associated to c_k has the bit at position w set to 1; otherwise, its value remains unaltered. By performing this operation for all the items $i \in I_+$ that are classified with c_k, we know which synsets are relevant for the class. After we processed all the classes $c \in C$ we obtain a description of the primitive classes that allow us to build the filters for the boolean class.

Interclass-based SBS Definition

Starting from the set $B = \{b_1, \ldots, b_K\}$, we can arbitrarily manage the elements $b_k \in B$ to generate *boolean classes*, i.e., the combination of primitive classes by means of a boolean operator. The first type of boolean class we are going to define, named *interclass* is formed by the combination of the binary sieves of the two classes b_k and b_q through an *AND* operator. Considering each element w of the two vectors, which indicates if a synset w is relevant or not for a class, the semantics of the operator is the following:

$$b_k[w] \wedge b_q[w] = \begin{cases} 1, \; if \; b_k[w] = 1 \; and \; b_q[w] = 1 \\ 0, \; otherwise \end{cases} \tag{7.3}$$

This boolean class indicates which synsets characterize all the classes of items involved. We can obtain this result recurring to the axiomatic set theory (i.e., the elementary set theory based on the Venn diagrams); indeed, we can consider each class of items as a set, and create a new interclass that characterizes the common elements of two or more SBSs, using an intersection operation $\cap$;

The example in Figure 7.1 is a simple demonstration of what said based on the axiomatic set theory. It describes the effect of a boolean *AND* operation applied

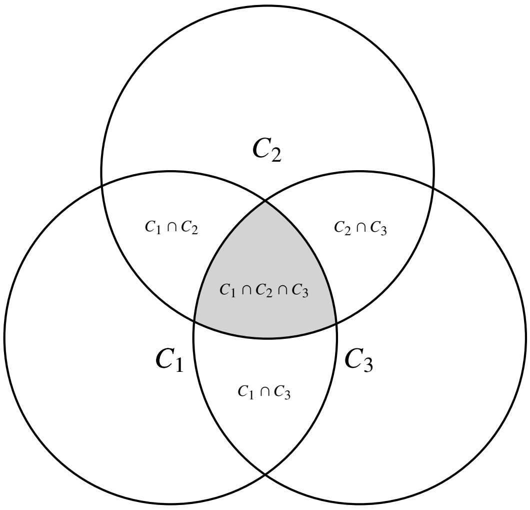

Figure 7.1: Inter-class definition

to the classes C_1, C_2 and C_3: in this case the result of operation $C_1 \cap C_2 \cap C_3$ represents a new interclass that we can use to refer to a precise target of users, in a more atomic way than with the use of the primitive classes.

To provide a more specific presentation of what is the result of an interclass-based SBS, we are going to provide an example (presented in Table 7.1), in which the two classes with most items in the dataset employed in our experiments (i.e., the classes 1 and 5) are combined with an *AND* operation. In the example, the vector has a fixed length and contains 21122 elements, which represent the synsets extracted from the dataset. The results show that when two classes are combined in order to extract the synsets that characterize both, around 15% of synsets that characterize just one class are discarded by the resulting interclass-based SBS.

Class	Num. of 1 occurrences	Num. of 0 occurrences	% of 1 occurrences	% of 0 occurrences
1	14175	6947	67.11	32.89
5	14825	6297	70.19	29.81
1 AND 5	11338	9784	53.68	46.32

Table 7.1: Example of interclass-based SBS considering the two classes with most items

In other words, this SBS has more non-relevant synsets with respect to the original classes (this is represented by the percentage of 0 occurrences), and provides knowledge of which synsets are able to describe both classes of items, allowing a more specific and narrow user segmentation that captures which users are interested in both classes.

Superclass-based SBS Definition

By combining the binary sieves of the two classes b_k and b_q through an *OR* operator, we can generate a new type of boolean class, named *superclass*. Considering each element w of the two vectors, which indicates if a synset w is relevant or not for a class, the semantics of the operator is the following:

$$b_k[w] \vee b_q[w] = \begin{cases} 1, \; if \; b_k[w] = 1 \; or \; b_q[w] = 1 \\ 0, \; otherwise \end{cases} \tag{7.4}$$

This boolean class would allow an advertiser to broaden a target, capturing in a semantic binary sieve the synsets that are characterizing for two or more classes. By using the axiomatic set theory, we can consider each class of items as a set, and create a new superclass that characterizes more primitive classes through an

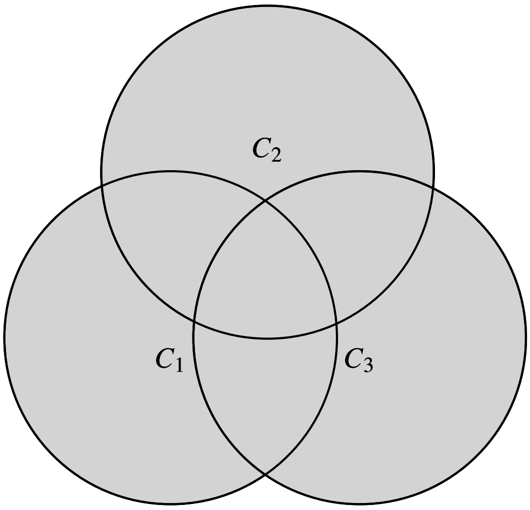

Figure 7.2: Superclass definition

union operation ∪ of two or more SBSs.

The example in Figure 7.2 shows a demonstration of what said based on the axiomatic set theory. It describes the effect of a boolean *OR* operation applied to the classes C_1, C_2, and C_3 (represented by the grey area).

To provide a more specific presentation of what is the result of a superclass-based SBS, Table 7.2 shows an example in which the same classes previously considered are combined with an *OR* operation. The results show that when two classes are combined in order to extract the synsets that characterize both, around 15% of synsets that characterize just one class are added to the resulting superclass-based SBS. In other words, this SBS has less non-relevant synsets with respect to the original classes (this is represented by the percentage of 0 occur-

Class	Num. of 1 occurrences	Num. of 0 occurrences	% of 1 occurrences	% of 0 occurrences
1	14175	6947	67.11	32.89
5	14825	6297	70.19	29.81
1 OR 5	17662	3460	83.62	16.38

Table 7.2: Example of superclass-based SBS considering the two classes with most items

rences), and provides knowledge of which synsets are able to describe at least one of the classes of items, allowing a more broad user segmentation that captures which users are interested in at least one of the classes.

Subclass-based SBS Definition

Another important entity that we can obtain through the managing of the elements $b \in B$ is the subset of a primitive class. It means that we can extract from a semantic binary sieve a subset of elements that express an atomic characteristic of the source set. For instance, if we consider a dataset where the items are movies, from a subset of genres of classification we can extract several semantic binary sieves that characterize some sub-genres of movies.

More formally, a *subclass* is a partition of a primitive or boolean class, e.g., for the primitive class *Comedy* we can define an arbitrary number of subclasses, applying some operation of the axiomatic set theory. In the example in Figure 7.3, we define a subclass *Comedy* \ *Romance*, in which all the synsets that characterize the *Romance* class are removed from the *Comedy* class. Therefore, only the comedy movies that do contain romance elements are represented through this boolean class.

Given two semantic binary sieves b_k and b_q, we can use b_q as a binary negation mask. For each element w of the vector, this operation inverts the binary value of the destination bits, as shown in Equation (7.5).

$$b_k[w] = \begin{cases} b_k[w], \; if \; b_q[w] = 0 \\ 0, \; otherwise \end{cases} \tag{7.5}$$

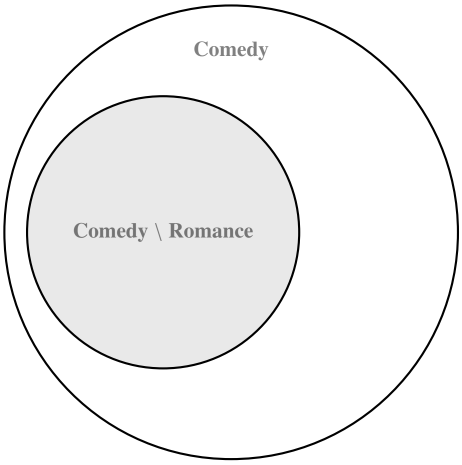

Figure 7.3: Sub-class definition

To provide a more specific presentation of what is the result of a subclass-based SBS, we are going to provide an example (presented in Table 7.3), in which we combine with a *NOT* operation the two classes of the dataset that have been most used to co-classify the items (i.e., the classes 5 and 14). The results show that when two classes are combined in order to extract the synsets that characterize

Class	Num. of 1 occurrences	Num. of 0 occurrences	% of 1 occurrences	% of 0 occurrences
5	14825	6297	70.19	29.81
14	8853	6947	67.11	32.89
5 NOT 14	11338	12269	41.91	58.09

Table 7.3: Example of interclass-based SBS considering the two classes with most items

both, around 30% of synsets that characterize the first class are discarded by the resulting subclass-based SBS. In other words, this SBS has more non-relevant synsets with respect to the first class from which we removed the synset that are relevant for the second, and provides knowledge of which synsets describe the first class of items but not the second, allowing a more specific and narrow user segmentation that captures which users are interested in items of the first class that do not contain in their description synsets of the second class.

Additional Considerations on the Boolean Classes

Given the elementary boolean operations we presented to create a boolean class given two classes and an operator, we can also create a new boolean class using the results of the previous operations, by combining them with a further operations of the same type, e.g., $(b_1 \vee b_2) \wedge (b_2 \neg b_3)$.

It should be also noted that only the *NOT* operation, together with one of the other two operations (*AND* and *OR*) is enough to express all possible combination of classes, as shown in Equation (7.6).

$$\begin{aligned} x \wedge y &= \neg(\neg x \vee \neg y) \\ x \vee y &= \neg(\neg x \wedge \neg y) \end{aligned} \tag{7.6}$$

7.6.4 Relevance Score Definition

This step compares the output of the two previous steps (i.e., the set B of binary vectors related to the *Semantic Binary Sieves*, and the set M of binary vectors related to the *user models*), in order to infer which classes are relevant for a user. The main idea is to consider which synsets are relevant for a user u (this information is stored in the user model m_u) and evaluate which classes are characterized by the synsets in m_u (this information is contained in each vector b_k, which contains which synsets are relevant for the class c_k). The objective is to build a relevance score $r_u[k]$, which indicates the relevance of the class c_k for the user u.

The key concept behind this step is that *we do not consider the items a user evaluated anymore*. Each vector in B is used as a filter (this is why the vectors are called *semantic binary sieves*), which allows us to estimate the relevance of each class for that user. Therefore, the relevance score of a class for a user can be used to generate non trivial segments, since *a user might be associated to classes of items she/he never expressed a preference for, but characterized by synsets that also characterize the user model.*

By considering each semantic binary sieve $b_k \in B$ associated to the class c_k and the user model m_u, we define a matching criteria Θ between each synset $m_u[w]$

in the user model, and the corresponding synset $b_k[w]$ in the semantic binary sieve, by adding 1 to the relevance score of that class for the user (element $r_u[k]$) if the synset is set to 1 both in the semantic binary sieve and in the user model, and leaving the current value as it is otherwise. The semantic of the operator is shown in Equation (7.7).

$$b_k[w]\Theta m_u[w] = \begin{cases} r_u[k]++,\ if\ m_u[w] = 1\ and\ b_k[w] = 1 \\ r_u[k],\ otherwise \end{cases} \tag{7.7}$$

The relevance scores built by this step will be used by our target definition algorithm, in order to infer which users are characterized by a specific class or set of classes.

7.6.5 Target Definition

This step defines the set of users that are part of the target. Given a boolean class of items c, we build a function $f : C^K \times \tau \rightarrow U$, that evaluates the relevance score $r_u(c)$ of each user $u \in U$ for that class, in order to understand if the class is relevant enough for a user to be included in the target. More specifically, the function operates as follows:

$$f(c) = \{u \in U \mid r_u(c) \geq \varphi\} \tag{7.8}$$

where φ is a threshold that defines the minimum value that the score has to take in order to consider the user as relevant for the target.

7.7 Experiments

The experiments have been performed using the Java language with the support of Java API implementation for WordNet Searching (JAWS), and the real-world dataset Yahoo! Webscope Movie dataset (R4)[3]. The experimental framework was developed by using a machine with an Intel i7-4510U, quad core (2 GHz × 4) and a Linux 64-bit Operating System (Debian Jessie) with 4 GBytes of RAM.

Strategy

To validate our proposal, we performed five sets of experiments:

1. **Data overview**. This experiment evaluates the distribution of the classes, by considering for how many users each class is the most relevant (i.e., the one for which a user has given most positive rating), in order to evaluate how trivial it is to perform a segmentation based on the classes; we will also analyze the number of genres with which each item is evaluated, in order to evaluate the capability of a positive rating to characterize a user preference not only in terms of items but also in terms of classes.

2. **Role of the semantics in the SBS data structure.** Our segmentation is based on a semantic data structure, which is built thanks to an ontology and to semantic analysis tools. We will validate this choice by evaluating the difference between the number of characterizing bits both in a binary vector

[3] http://webscope.sandbox.yahoo.com

built by analyzing the original words of the item descriptions and the SBS built thanks to the semantic analysis.

3. **Setting of the φ parameter.** The segmentation is built by putting together all the users with a relevance score higher than a threshold φ. This experiment will evaluate the threshold for each class by employing the elbow criterion, which evaluates the relevance score of each user for a class and detects the point in which the relevance score does not characterize the class anymore, since too many users are included in the segment that represents it.

4. **Analysis of the segments.** This experiment will analyze the segments of users targeted for each class, in order to evaluate the capability of our proposal to include also users who do not express explicit preferences for a class but might be interested in it.

5. **Performance analysis.** Given a new item classified with a class, we will evaluate the number of second it takes to update the SBS data structure (i.e., to perform the semantic disambiguation, evaluate the synsets in the item description, and include this information in the SBS). Note that descriptions of different lengths lead to different a computational effort, so this analysis will allow us to evaluate the performance of the approach from different perspectives.

Note that in order to validate the capability of our proposal to detect users who are not characterized by explicit preferences for a class, we will compare with the so-called topic-based approach employed by both Google's AdWords and Facebook's Core Audiences. For experiments number 4 and 5, we will also build a relevance score for each user and each class, by considering how many movies of a genre a user evaluated (i.e., we are considering a scenario in which the topic of interest is a genre of movies, which is equivalent to our classes). This is done since the companies did not reveal how they associate users to topics, and in order to make a direct comparison between an approach that uses explicit preferences and our semantic approach.

The employed dataset is Yahoo! Webscope Movie dataset (Appendix B). In order to detect the relevance score to take into account during the user segmentation (i.e., the threshold value after which we can consider a synset as discriminant), we use the well-known *elbow criterion* described in Appendix B.

7.7.1 Results

This section will present the results of the user targeting performed by the proposed approach, by studying its behavior on each type of class previously defined. Note that the results of the targeting built by using the primitive class-based semantic binary sieve is presented along with those of the superclass-based approach (i.e., when presenting the results, we will start with a primitive class, then start merging them with an *OR* operator).

Data Overview

In the first experiment we performed a preliminary study on the relation between the users and the native classification of the items in the dataset, in order to analyze the distribution of users with respect to the classes. For each class, Figure 7.4 reports the number of users for which that class is the one with most evaluations. Moreover, above each point, we indicate the ranking of the classes, based on the number of users.

The results show that 15 out of 19 classes have more than 1000 users for which it is the most relevant. Moreover, 6 classes are the most relevant for a number of users between 6000 and 8000. The fact that each class is the most relevant for a lot of users, and it does not exist a unique dominant class that is the most relevant for all the users, ensures that the segmentation process is not trivial (indeed, if all the users could be associated to one class, the relevance scores for that class would be very high and the segmentation would be trivial).

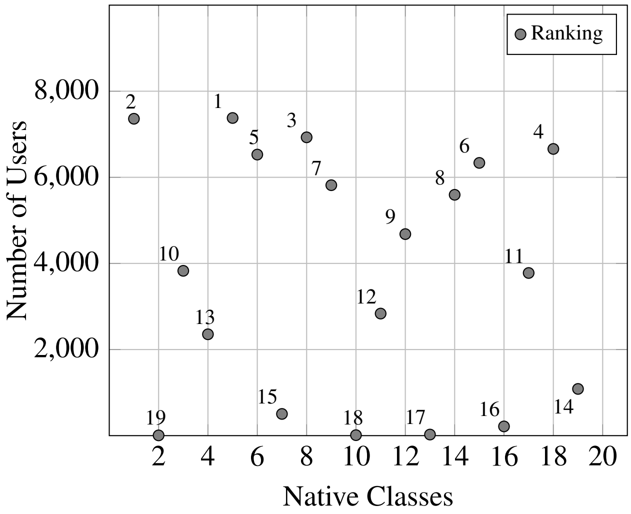

Figure 7.4: User distribution for native classes

In Figure 7.5 we see the number of items that have been classified with multiple genres. The results show that the vast majority of the items has been classified with a single genre and it is rare to find items classified with multiple genres (only one item in the whole dataset has 6 co-classifications). This means that when a user positively evaluates an item, it is possible to derive a preference also in terms of classes, and the synset contained in an item description will characterize the SBS of just one class (i.e., the SBSs will not be similar, since disjoint sets of items will contribute to each binary vector).

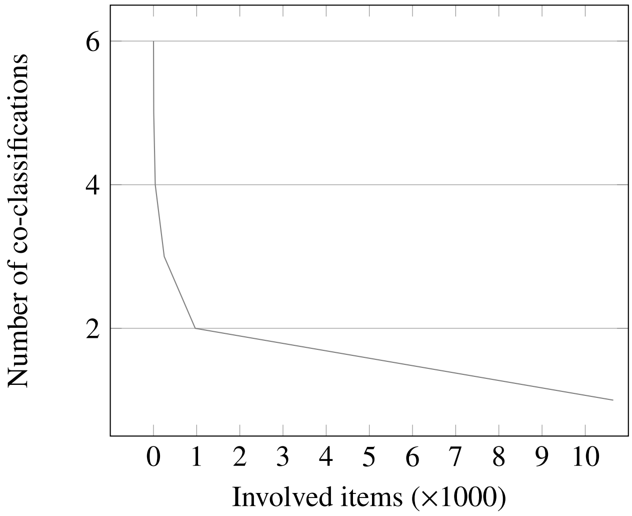

Figure 7.5: Number of co-classification for item

Role of the Semantics in the SBS Data Structure

In order to validate our choice to have a semantic data structure, we built the equivalent of the SBS by considering the original words available in the item descriptions. This means that Wordnet was not employed and no synset was collected, and of course we could not perform a semantic disambiguation of the words. We did this comparison for each class and since 19 classes are involved, in order to facilitate the interpretability of the results, on the one hand we summed the amount of 1 occurrences in the 19 SBSs, while on the other hand we summed the amount of 1 occurrences in the 19 binary vectors containing the words.

The results presented in Table 7.4 show that when considering the words the classes are characterized by 30% less elements, with respect to their semantic counterpart. This shows the high relevance that the employment of the ontology has, and how important it is to perform a semantic disambiguation among the words. Indeed, by associating the correct semantic sense to each word it is possible to avoid phenomena that characterize this area, such as synonymity, and to have more accurate information about what characterizes each class of items.

$Words$	63772
$Synsets$	91130
$Difference$	+30.02%

Table 7.4: Synsets and words cardinality

Setting of the φ parameter

In order to set the value of φ that allows to consider a class as relevant for a user, we adopted the elbow criterion described in Appendix B. Table 7.5 shows the threshold values derived from elbow criterion, i.e., for each class we indicate the minimum value the relevance score of a user has to have, in order for a user to be included in the segment of that class. In order to be able to compare our semantic approach to a topic-based segmentation that considers the explicitly expressed preferences, we performed this analysis for both types of vectors that describe a class.

Note that the threshold values for the SBS data structure are much higher with

$Class$	$Topic - based$	$SBS - based$
1	29	1414
2	7	0
3	4	857
4	9	778
5	45	1438
6	8	1195
7	2	287
8	40	1369
9	12	1162
10	1	9
11	4	789
12	12	1112
13	1	47
14	8	1170
15	17	1269
16	3	270
17	15	1033
18	16	1269
19	6	535

Table 7.5: Elbow values

respect to the topic-based values. This means that when the semantics behind the item descriptions are considered (and not just the explicitly expressed preferences), a user is associated to a class many more times, thus showing the capability of our approach to capture latent links between the users and the classes.

Analysis of the segments

In this section, we analyze the produced user segments. For each of the primitive classes, we will present an analysis of the segments generated by both the baseline topic-based approach and by our SBS approach. Regarding the boolean classes, since all the possible ways to combine multiple classes with the three operators are impossible to analyze, we decided to study the segments generated through an interclass- and a superclass-based SBS by combining the two classes with most and least items in the dataset (respectively, classes 1 and 5, and 13 and 10[4]); this allowed us analyze our approach both in a scenario where a lot of information is available and in a case in which the users expressed very little preferences for that class.

The subclass-based segmentation was studied by considering the two classes with which the items were most co-classified (i.e., classes 5 and 14). Table 7.6 presents the obtained results and the columns contain the following information: *Class* contains the identifier of the class that characterizes the interest of the users in it, *Topic-based Segments* and *SBS Segments* report the amount of users added to the segment by the two approaches, *Shared Users* and *Unshared Users* respectively report how many users have been identified by both approaches and how many have been detected with our proposal, *cclass* reports for how many unshared users a class that was relevant for them was also co-classified with the considered class (a positive outcome means that we added a relevant user to the segment of a class, since the class considered in the segment is naturally correlated with a class that is relevant for the user)[5], and column % reports the percentage of relevant unshared users detected by our approach (i.e., those for which a co-classified relevant class was found).

When analyzing the results of the primitive classes, we can notice that the SBS segments contain from 3 to 155 times more users with respect to their Topic-based

[4] Note that class 2 is actually the class with least items, but we will show that its relevance in the dataset is so low that it cannot be managed in practice.

[5] The only exception to this analysis regards the NOT operator, in which we analyzed how many users had a semantic relevance score higher than the threshold in the first class but not in the second.

Class	*Topic − based Segments*	*SBS Segments*	*Shared Users*	*Unshared Users*	*co − classifications*	*%*
1	208	604	206	398	394	98.99
2	0	0	0	0	0	0.00
3	177	940	147	793	786	99.12
4	53	1013	37	976	969	99.28
5	120	590	120	470	466	99.15
6	242	717	200	517	510	98.65
7	40	1518	28	1490	1482	99.46
8	117	622	117	505	499	98.81
9	99	737	92	645	639	99.07
10	0	1026	0	1026	1015	98.93
11	90	1015	77	938	931	99.25
12	87	762	75	687	682	99.27
13	0	1945	0	1945	1930	99.23
14	243	725	214	511	507	99.22
15	185	666	178	488	481	98.57
16	12	1870	9	1861	1848	99.30
17	78	818	66	752	746	99.20
18	196	668	193	475	468	98.53
19	22	1228	20	1208	1200	99.34
5 AND 1	82	640	82	558	552	98.92
5 OR 1	246	559	244	315	311	98.73
13 AND 10	0	3002	0	3002	2971	98.97
13 OR 10	0	1737	0	1737	1724	99.25
5 NOT 14	22	200	19	181	72	36.00

Table 7.6: Experiments result

counterparts. We can also notice that the difference between the amount of users added to a segment is higher for the classes that are the relevant for less users (i.e., classes 3, 4, 7, and 16, which in Figure 7.4 are all associated to the lowest part of the figure).

In addition, we can notice that our approach is able to detect a balanced amount of users for each class; this would allow advertisers to efficiently target users, no matter which class is considered. A related and important characteristic of our approach, is its capability to *detect an homogeneous amount of users no matter how much explicit information about the preferences for the classes are expressed*; indeed, even the less relevant classes can lead to a targeting that considers a high amount of users (note that for the two least relevant classes 10 and 13, the topic-based approach cannot detect any user, while we are able to characterize those classes thanks to the semantics). The only exception to this is class 2 (Adult), which is the least relevant in the dataset and the amount of positive preferences for these items was so little that neither of the two approaches could add users to its segment.

The very relevant classes in the dataset, such as 1 and 5, are not flooded with too many users and elbow criterion has proven to be an effective criterion to choose the threshold.

Regarding the unshared users, detected by our approach but not by the topic-based one, we can notice that more than 98% of them are relevant, since we found another class that is relevant for them when considering the topic-based preferences, and whose items are co-classified with the considered class.

The analysis of the interclass-based segments (AND operator) and of the superclass-based segments (OR operator), show very similar results than those reported for the primitive classes. These results confirm the capability of our approach to work well when little explicit information is available, even when the classes are combined into a boolean one. An interesting result to analyze is the last line of the table, related to the subclass-based segment 5 *NOT* 14, for which 36% of the unshared users that have been detected are relevant.

When looking for users interested by *Comedy* movies (class 5) that do not contain *Romantic* elements (class 14), our approach detected 9 times the users of the topic-based one; out of these 200 detected users, 72 of them (3x the users detected by the topic-based approach) reported a semantic relevance for class 5 but not for class 14. Regarding the remaining users, they do like both Comedy and Romance movies, but this result shows that even if we remove the Romance elements from the Comedy movies, a strong interest for the Comedy genre remains (in other words, they could be targeted as users that might like Comedy movies that do not contain Romance elements).

Performance Analysis

Figure 7.6 reports the number of seconds it takes for our approach to update the SBS of a class once a new item receives a positive rating. Note that to simplify the readability of the results we report just the performance considering the first 100 items of the dataset. The dashed line in the figure represents the average number of seconds considering all the values.

These results show that different items lead to a quite different performance. We inspected on this result furthermore, and we saw that all the different steps performed at the beginning of the computation play a role in the performance of the approach. Indeed, when an item description contains more synsets, the number of seconds necessary to complete the data structure update is higher, but there is not a direct correlation between the number of synsets and the performance (i.e., item 19 is not the one with the highest number of synsets among the 100 items considered, even though it is the one with the lowest performance). Indeed, the other steps, such as the text preprocessing, influence the performance and lead to the different results.

Regarding the performance of the SBS update, which is the core of our approach, it should also be noted that it lends itself well to a processing through grid computing. Indeed, the processing of the individual items might be done on different computers. For example, a possible optimized solution is to use a single computer for the computation of the SBS for a subset of items, so that the computation of the final SBS is distributed over different computers, by employing large scale distributed computing models like MapReduce. It is trivial to notice that the final SBS is a combination of the output of the individual machines through an OR operator (if a synset is relevant for an item, it is relevant for the class).

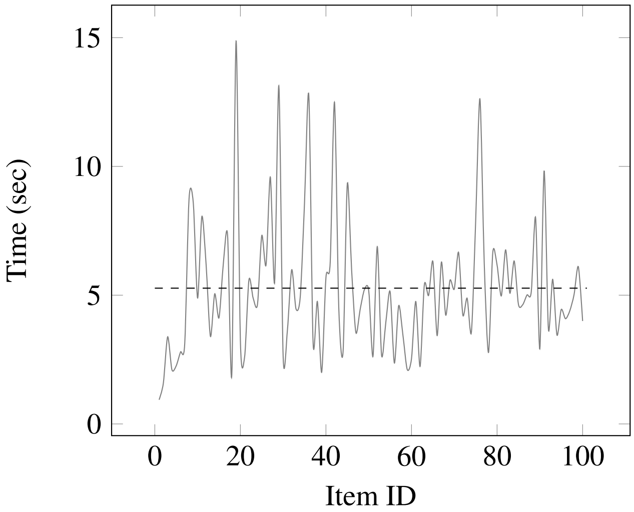

Figure 7.6: Execution time

7.8 Conclusions and Future Work

This thesis presented a novel semantic user segmentation approach that exploits the description of the items positively evaluated by the users. The target detection is based on the definition of a set of *binary sieves*, new entities that allow to characterize primitive or boolean classes (i.e., set of classes combined through boolean operations on the classes). The experimental results show the ability of our approach in order to model in an effective way a target of users within the domain taken into account.

Future work will test the capability of our semantic approach to characterize clusters of users whose purchased items are semantically related. This approach would allow us to target the users in a different way, e.g., by performing group recommendations to them (i.e., by recommending items to groups of “semantically similar” users).

On the Role of Similarity and Diversity in Fraud Detection Systems

Preface

The concepts of similarity/diversity are here exploited to improve the performance of a fraud detection system that operates in the e-commerce environment.

Chapter 8

A Proactive Approach for the Detection of Frauds Attempts

8.0.1 Introduction

In this context we extend the concept of coherence to the fraud detection systems, where a noise detected in the data stream of the financial transactions of a user represents a potential fraud. As discussed in Chapter 4.2.2, considering that the number of fraudulent transactions is typically much smaller than legitimate ones, the distribution of data is highly unbalanced, reducing the effectiveness of many learning strategies used in this field. A fraud detection system can basically operate by adopting a static or dynamic strategy. In the first case, the model used to detect the frauds is completely generated after a certain time period, while in the second case it is generated one time, then updated after a new transaction.

The strategy used in many of the cited approaches is based on the detection of the suspicious changes in the user behavior, a quite trivial approach that in several cases leads toward false alarms. Most of these false alarms are related to the absence of extended criteria during the evaluation of the suspect activities, since numerous state-of-the-art approaches exclude some non numeric data from the evaluation process, due to their incapacity to manage it. This happens because employing machine learning approaches, such as the Random Forests, all the types of data that involve a lot of categories (typically more than 32) cannot be handled. Thinking about real-world transactional data, they usually involve much more than 32 categories (e.g., the places in the transactions).

8.0.2 Overview

The exponential and rapid growth of the E-commerce based both on the new opportunities offered by the Internet, and on the spread of the use of debit or credit cards in the online purchases, has strongly increased the number of frauds, causing large economic losses to the involved businesses.

The design of effective strategies able to face this problem is however particularly challenging, due to several factors, such as the heterogeneity and the non stationary distribution of the data stream, as well as the presence of an imbalanced class distribution. To complicate the problem, there is the scarcity of public datasets for confidentiality issues, which does not allow researchers to verify the new strategies in many data contexts.

Differently from the canonical state-of-the-art strategies, instead of defining a unique model based on the past transactions of the users, we follow a Divide and Conquer strategy, by defining multiple models (user behavioral patterns), which we exploit to evaluate a new transaction, in order to detect potential attempts of fraud. We can act on some parameters of this process, in order to adapt the models sensitivity to the operating environment.

Considering that the proposed models do not need to be trained with both the past legitimate and fraudulent transactions of a user, since they use only the legitimate ones, we can operate in a proactive manner, by detecting fraudulent transactions that have never occurred in the past. Such a way to proceed also overcomes the data imbalance problem that afflicts the machine learning approaches. The evaluation of the proposed approach is performed by comparing it with one of the most performant approaches at the state of the art as Random Forests, using a real-world credit card dataset.

8.0.3 Proposed Solutions

The vision behind this part of of research is to extend the canonical criteria, integrating them the ability to operate with heterogeneous information (i.e., numeric and non numeric data), and by adopting multiple behavioral patterns of the users. This approach reduces the problems previously underlined, related with the scarcity, heterogeneity, non stationary distribution, and presence of an imbalanced

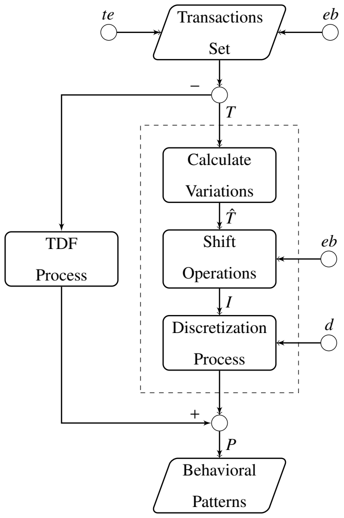

Figure 8.1: System Architecture

class distribution, of the transactions data. This is possible because it takes into account all parts of a transaction, considering more information about it, contrasting the scarcity of information that leads toward an overlapping of the classes of expense. By means of the generation of multiple behavioral models of a user, made by dividing the sequence of transactions in several event-blocks, it faces instead the problem of the non stationarity of data, modeling anyway the user behavior effectively.

The block diagram in Figure 8.1 introduces a high-level architecture of the proposed approach. As shown, the past transactions of a user are processed in order to define a series of behavioral patterns that characterize different parts of the transaction history of the user. Such process takes into account the importance of certain transaction elements in the fraud detection process, such as, for instance, the place where the transaction happens.

The first block in Figure 8.1, labeled *Transactions Set*, contains the initial set of transactions (past transactions of a user) to process in order to define a set of behavioral patterns. Its output depends on the presence of a new transaction *te* to evaluate in the input channel: in absence of it, all transactions will be in the output; otherwise will be only the $eb-1$ transactions (where *eb* denotes the size of event-block), followed by the *te* transaction to evaluate. This happens because in this case we need as output only a single behavioral pattern of size *eb*. As input of the second block (*Calculate Variations*), we have a set of transactions T, composed by the output of the previous block, after the removal of a characterizing field (in this case, the field *place*) designed as *Transaction Determinant Field* (TDF). A TDF is a part of a transaction to which we have decided to give more relevance during the fraud detection process, in accord with the operations of the block *TDF Process*, described in Section 8.0.6.

The set of transactions T is processed by the block *Calculate Variations*, in order to convert it into absolute numeric variations measured between each pair of contiguous transactions, as described in Section 8.0.6. The absolute variations in the set $\hat{T}$ are processed in the *Shift Operations* block, in accord with the value in the input channel *eb*, that defines the size of the *Event-block Shift Vector* (EBSV), as described in Section 8.0.6. The result is a set I of behavioral patterns.

The next *Discretization Process* block converts the continuous values present in the set I, output of the previous block, in discrete values, according with the value of discretization defined in the input channel d, as described in Section 8.0.6. The final set of *Behavioral Patterns* P, in the output of the entire process, is built by integrating the output of the block *Discretization Process* block, with the TDF information of the block *TDF Process*.

The level of reliability of a new transaction is evaluated by comparing, through the *cosine similarity* the behavioral pattern P obtained by performing the entire process with the transaction to evaluate applied to input channel *te*, with the set of behavioral patterns generated following the same process without any transaction applied in this channel, as described in Section 8.0.6.

Differently from the canonical machine learning approaches at the state of the art (e.g., the Random Forests approach to which this work is compared), the proposed models do not need to be trained with the fraudulent transactions, because their definition needs only the legitimate ones. This overcomes the problem of data imbalance that afflicts the machine learning approaches. The level of reliability of a new transaction is evaluated by comparing (through the *cosine similarity* measure) its behavioral pattern to each of the behavioral patterns of the user, generated at the end of the previously described process.

This work provides the following main contributions to the current state of the art:

- introduction of a strategy able to manage heterogeneous parts of a financial transaction (i.e., numeric and non numeric), converting them in absolute numeric variations between each pair of contiguous events;
- definition of the *Transaction Determinant Field* (TDF) set, a series of distinct values extracted from a field of the transaction, and used to give more

importance to certain elements of a transaction, during the fraud detection process;

- introduction of the *Event-block Shift Vector* (EBSV) operations, made by sliding a vector of size *eb* (event-block) over the sequence of absolute variations previously calculated, in order to store, in the behavioral patterns of a user, the average values of the variations measured in each event-block;
- definition of a discretization process used to adjust the sensitivity of the system in the fraud detection process, by converting the continuous values in the behavioral patterns in output to the EBSV process, in a number of *d* levels (*discretization*);
- formalization of the process of evaluation of a new transaction, performed by comparing, through the *cosine similarity*, its behavioral pattern with the user behavioral patterns in *P*, in order to assign it a certain level of reliability.

8.0.4 Adopted Notation

Definition 8.1 (Input set) *Given a set of users $U = \{u_1, u_2, \ldots, u_M\}$, a set of transactions $T = \{t_1, t_2, \ldots, t_N\}$, and a set of fields $F = \{f_1, f_2, \ldots, f_X\}$ that compose each transaction t (we denoted as $V = \{v_1, v_2, \ldots, v_W\}$, the values that each field f can assume), we denote as $T_+ \subseteq T$ the subset of legal transactions, and as $T_- \subseteq T$ the subset of fraudulent transactions. We assume that the transactions in the set T are chronologically ordered (i.e., t_n occurs before t_{n+1}).*

Definition 8.2 (Fraud detection) *The main objective of a fraud detection system is the isolation and ranking of the potentially fraudulent transactions [106] (i.e., by assigning a high rank to the potential fraudulent transactions), since in the real-world applications, this allows a service provider to focus the investigative efforts toward a small set of suspect transactions, maximizing the effectiveness of the action, and minimizing the cost. In [106], the average precision (here denoted as* α*) is considered as the correct measure to use in this kind of process. Its formalization is shown in Equation 8.1, where* N *is the number of transactions in the set of data, and* $\Delta R(t_r) = R(t_r) - R(t_r - 1)$*. Denoting as* π *the number of fraudulent transactions in the set of data, out of the percent* t *of top-ranked candidates, denoting as* $h(t) \leq t$ *the hits (i.e., the truly relevant transactions), we can calculate the* $recall(t) = h(t)/\pi$*, and* $precision(t) = h(t)/t$ *values, then the value of* α*.*

$$\alpha = \sum_{r=1}^{N} P(t_r)\Delta R(t_r) \tag{8.1}$$

Lemma 8.1 *The values* $R(t_r)$ *and* $P(t_r)$ *represent, respectively, the recall and precision of the* r^{th} *transaction, then we have* $\Delta R(t_r) = (1/\pi)$ *when the* r^{th} *transaction is fraudulent, and* $\Delta R(t_r) = 0$ *otherwise.*

Corollary 8.1 *When the set processed by the Equation 8.1 is a set composed by a certain number of legitimate transactions, but with only one potential fraudulent transaction to evaluate* $\hat{t}$ *(i.e.,* $T_+ \cup \hat{t}$*), according to the Definition 8.2 we have*

$\pi = 1$ and $t = 1$. Consequently, from the previous Lemma 8.1, we can define a binary classification of the transaction $\hat{t}$, since $\Delta R(t_r) = 1$ when the r^{th} transaction is fraudulent, and $\Delta R(t_r) = 0$ otherwise, which allow us to mark a new transaction as reliable or unreliable.

Definition 8.3 (Performed tasks) *In order to operate with only numeric elements, able to characterize the sequence of transaction events, we transform the set T in the set $\hat{T} = \{\hat{t}_1 = |t_2 - t_1|, \hat{t}_2 = |t_3 - t_2|, \ldots, \hat{t}_N = |t_N - t_{N-1}|\}$, where $|\hat{T}| = (|T| - 1)$, and each subtraction operation is performed on all fields $f \in F$ of the considered transactions, by using a different criterion for each type of data. We also denote as $I = \{i_1, i_2, \ldots, i_Z\}$ the set of behavioral patterns generated at the end of the shift process, performed on the set $\hat{T}$, where the shift operation aims to extract the average value of a certain number (defined by the event-block parameter) of contiguous variations of the set $\hat{T}$. The purpose of this process is the definition of a set of behavioral patterns, which takes into account a series of contiguous events (i.e., the average variation), instead of only one (or all). To uniform all the variations in I in a certain range of values, we define a new set $P = \{p_1, p_2, \ldots, p_Y\}$, with contains the same elements of I, but where the value of each field $f \in F$ is discretized, according to certain number of levels (defined by the discretization parameter d, with $d \geq 2$)). It should be noted that $|I| = |P|$.*

8.0.5 Problem Definition

As previously described in Definition 8.2, an ideal fraud detection approach should have a value of α as close as possible to 1, since it means that all fraudulent transactions π have been ranked ahead the legal ones. The objective is then to maximize the α value, by ordering the new transactions on the basis of their similarity value with the behavioral patterns in P, in order to rank the fraudulent transactions ahead the legal ones:

$$\max_{0 \leq \alpha \leq 1} \alpha = \sum_{r=1}^{N} P(t_r) \Delta R(t_r) \tag{8.2}$$

8.0.6 Approach

The steps needed to implement the proposed strategy, schematically shown in the block diagram in the Introduction (Figure 8.1), can be grouped into the following five steps:

- **Absolute Variation Calculation**: conversion of the transactions set T of a user into a set of absolute numeric variations between two contiguous transactions $t \in T$, adopting a specific criterion for each type of data in the set F;
- **TDF Definition**: creation of a *Transaction Determinant Field* (TDF) set, a series of distinct terms, extracted from the field *place*, used to define a

binary element in each pattern of the set *P*, allowing to give more relevance to this field during the fraud detection process;

- **EBSV Operation**: application of a *Event-block Shift Vector* (EBSV) over the set of absolute numeric variations $\hat{T}$, aimed to calculate the average value of the elements in the event-block *eb*, storing the results as patterns in the set *I*;

- **Discretization Process**: discretization of the average values in the set *I*, in accord with a defined number of levels *d* (discretization). It allows to adjust the sensitivity of the system during the fraud detection process. The result of this operation, along with the result of the TDF query, defines the set of behavioral patterns *P*;

- **Transaction Evaluation**: assignation of a level of reliability to a new transaction, by comparing all patterns in the set *P* with the pattern obtained by inserting the transaction to evaluate as last element of the set T, repeating the process previously described only for the last *eb* transactions.

Absolute Variations Calculation

In order to convert the set of transactions T in the set of absolute variations $\hat{T}$, according with the criterion exposed in Section 5.3.3, we need to define a different kind of operation for each different type of data in the set F (excluding the field *place*, used in the *Transaction Determinant Field*). In this case, in accord with the adopted credit card dataset (described in Appendix B), we need to define three type of operations: numeric absolute variation, temporal absolute variation, and textual absolute variation.

Numeric Absolute Variation. Given a numeric field $f_x \in F$ of a transaction $t_n \in T$ (i.e., in this case the field *amount*), we calculate the Numeric Absolute Variation (NAV) between each pair of fields, that belong to two contiguous transactions (denoted as $f_x^{(t_n)}$ and $f_x^{(t_{n-1})}$), as shown in Equation (8.3). The result is the absolute difference between the values taken into account.

$$NAV = |f_x^{(t_n)} - f_x^{(t_{n-1})}| \tag{8.3}$$

Temporal Absolute Variation. Given a temporal field $f_x \in F$ of a transaction $t_n \in T$ (i.e., in this case the field *date*), we calculate the Temporal Absolute Variation (TAV) between each pair of fields, that belong to two contiguous transactions (denoted as $f_x^{(t_n)}$ and $f_x^{(t_{n-1})}$), as shown in Equation 8.4). The result is the absolute difference in days, between the two dates taken in account.

$$TAV = |days(f_x^{(t_n)} - f_x^{(t_{n-1})})| \tag{8.4}$$

Descriptive Absolute Variation. Given a textual field $f_x \in F$ of a transaction $t_n \in T$ (i.e., in this case the *description* field), we calculate the Descriptive Absolute Variation (DAV) between each pair of fields, that belong to two contiguous transactions (denoted as $f_x^{(t_n)}$ and $f_x^{(t_{n-1})}$), by using the *Levenshtein Distance* metric described in Appendix B, as shown in Equation 8.5). The result is a value in the range from 0 (complete dissimilarity) to 1 (complete similarity).

$$DAV = \text{lev}_{f_x^{(t_n)}, f_x^{(t_{n-1})}} \tag{8.5}$$

TDF Definition

In order to define the *Transaction Determinant Field* (TDF) from a field that we decide to consider as crucial in the fraud detection process (in this case, the field *place*), we extract from the set of transactions all distinct values $v_1, v_2, \ldots, v_W$ of this field, storing them in a new set $\hat{V} = \{\hat{v}_1, \hat{v}_2, \ldots, \hat{v}_W\}_{\neq}$, according with the formalization introduced in Section 5.3.3.

The set $\hat{V}$ will be queried in order to check if the place of the transaction under analysis is a place already used by the user, or not. When it is true, the binary value of the corresponding element of the behavioral pattern (i.e., the field *place* of the behavioral pattern of the transaction to evaluate, defined as described in Section 8.0.6) is set to 1, otherwise to 0. It should be noted that this value is always set to 1 in the behavioral patterns related with the past transactions of the user.

EBSV Operation

After we have converted the set of transaction T into a set of absolute variations $\hat{T}$, adopting the criteria exposed in Section 8.0.6, we operate the shift operation by sliding the *Event-block Shift Vector* over the sequence of absolute variation values stored in $\hat{T}$, one step at a time, extracting the average value of the variations present in the defined event-block eb. Given a event-block $eb = 3$, a set of variations $\hat{T} = \{v_1, v_2, v_3, v_4, v_5, v_6\}$, we can execute a maximum of $|C|$ shift operations, with $|C| = |I| = (|\hat{T}| - |eb| - 1)$, as shown in the Equation 8.6.

$$\begin{aligned}
&\hat{T} = [v_1, v_2, v_3, v_4, v_5, v_6] \\
&\qquad\qquad \Downarrow \\
&c_1 = \tfrac{v_1+v_2+v_3}{|eb|}, c_2 = \tfrac{v_2+v_3+v_4}{|eb|} \\
&c_3 = \tfrac{v_3+v_4+v_5}{|eb|}, c_4 = \tfrac{v_4+v_5+v_6}{|eb|} \\
&\qquad\qquad \Downarrow \\
&I = [c_1, c_2, c_3, c_4]
\end{aligned} \tag{8.6}$$

The sequence of values calculated in each event-block eb, for each considered field (i.e., *description*, *amount*, and *date*), represents the set I of behavioral patterns of the user. It should be observed that we have to discretize the patterns obtained through the shift process, adding to them the binary value determined by querying the *Transaction Determinant Field* set (as described in Section 8.0.6), before using them in the evaluation process of a new transaction.

Discretization process

The continuous values $f \in F$ present in the pattern set I, obtained through the shift operation described in Section 8.0.6), must be transformed in discrete values, in accord with a certain level of *discretization* d. It allow us to determine the level of sensitivity of the system during the fraud detection process.

The result is a set $P = \{p_1, p_2, \ldots, p_Y\}$ of patterns that represent the behavior of a user in different parts of her/his transaction history. Given a discretization value d, and a set of patterns I, each continuous value v_c of a field f (i.e., we process only the fields *description*, *date*, and *amount*, because the field *place* assumes a binary value determined by the TDF process) is transformed in a discrete value v_d, following the process shown in the Equation 8.7.

$$v_d = \left\lceil \frac{v_c}{\left(\frac{max(f) - min(f)}{d}\right)} \right\rceil \tag{8.7}$$

Transaction Evaluation

To evaluate a new transaction, we need to compare each behavioral pattern $p \in P$ with the single behavioral pattern $\hat{p}$ obtained by inserting the transaction to evaluate as last element of the set T, repeating the entire process previously described (variation calculation, shift, and discretization) only for the transactions present in the last event-block (i.e., the event-block composed by the last $|time\text{-}frame|$ transactions of the set T, were the last one element is the transaction to evaluate).

The comparison is performed by using the *cosine similarity* metric (described in Section B.1.3), and the result is a series of values in the range from 0 (transaction completely unreliable) to 1 (transaction completely reliable). It should be noted that the value of the field *place* depends on the result of the query operated on the TDF set, as described in the Section 8.0.6.

The value of similarity is the average of the sum of the minimum and maximum values of cosine similarity $cos(\theta)$, measured between the pattern $\hat{p}$ and all patterns of the set P, i.e., $sim(\hat{p}, P) = (min(\cos(\theta)) + max(\cos(\theta)))/2$. The result is used to rank the new transactions, on the basis of their potential reliability.

8.0.7 Experiments

In order to evaluate the proposed strategy, we perform a series of experiments using a real-world dataset related to one-year (i.e., 2014) of credit card transactions[1]. The proposed EBSV approach was developed in Java, while the implementation of the state-of-the-art approach, used to evaluate its performance, was made in *R*[2], using the *randomForest* package.

The dataset used for the training, in order to generate the set of behavioral patterns P, contains one year of data related to the credit card transaction of a user (described in Appendix B).

[1] A private dataset provided by a researcher

[2] https://www.r-project.org/

Strategy

Considering that it has been proved [88] that the *Random Forests* (RF) approach outperforms the other approaches at the state of the art, in this work we chose to compare the proposed EBSV approach only to this one, excluding alternative approaches, such as *Support Vector Machine* (SVM), or *Neural Network* (NNET). For the reason described in Section 5.3.3, we perform this operation by comparing their performance in terms of Average Precision (AP). Since we do not have any real-world fraudulent transactions to use, we first define a synthetic set of data T_-, composed by 10 transactions aimed to simulate several kind of anomalies, as shown in Table 8.1 (they have been marked as *unreliable*, as well as the other ones have been marked as *reliable*).

During the experiments aimed to compare the performance of the proposed EBSV approach, with those of the RF one, we adopt the *k-fold cross-validation* criterion. Regarding the EBSV approach, we first partitioned the entire dataset T_+ into k equal sized subsets (according with the dataset size, we set $k = 3$), which denote as $T_+^{(k)}$. Thus, each single subset $T_+^{(k)}$ is retained as the validation data for testing the model, after adding to it the set of fraudulent transactions T_- (i.e., $T_+^{(k)} \cup T_-$). The remaining $k - 1$ subsets are merged and used as training data to define the user models.

We repeat the same previous steps for the RF approach, with the difference that, in this case, we add the set T_- also to training data. In both cases, we consider as final result the average precision (AP) related to all k experiments. Since the RF approach is not able to operate a textual analysis on the transaction description, and that is well-known that the RF approaches are biased by the categorical variables that generate many levels (such as the *Description* field), we do not use this field in the RF implementation. In addition, in order to work with the same type of data, in the RF implementation we converted the information of the field Date, in time intervals between transactions, expressed in days.

For reasons of reproducibility of the RF experiments, we fix the seed value of the random number generator by the method *set.seed(123)* (the value is not relevant). The RF parameters (e.g., the number of trees to grow) have been defined in experimental way, by researching those that minimized the *error rate* given as output during the RF process. The experiments are articulated in the following two steps:

- definition of the values to assign to the parameters that determine the performance of the EBSV approach (i.e., *event-block* and *discretization*), as described in Section 8.0.7;

- evaluation of the EBSV performance, comparing to the RF approach, by testing the ability to detect a number of $2, 4, \ldots, 10$ fraudulent transactions (respectively, a fraudulent transactions percentage of $2.8\%, 5.5\%, \ldots, 12.8\%$).

In order to evaluate the similarity between the behavioral pattern of a transaction under analysis, and each of the behavioral patterns of the user, generated at the end of the process exposed in Section 8.0.6, we use the cosine similarity

TransactionID		Fields Values (1=anomalous 0=regular)				
From	*To*	*Description*	*Place*	*Date*	*Amount*	*Status*
1	2	1	0	0	0	unreliable
3	4	0	1	0	0	unreliable
5	6	0	0	1	0	unreliable
7	8	0	0	0	1	unreliable
9	10	1	1	1	1	unreliable

Table 8.1: Fraudulent Transactions Set

metric, described in Appendix B. We also use the average precision (AP) metric, described in Appendix B, because it is considered as the correct measure to use in the fraud detection context, as described in Definition 8.2.

Parameters Tuning

Considering that the performance of the proposed approach depends on the parameters *eb* (*event-block*) and *d* (*discretization*), before evaluating its performance, we need to detect their optimal values. To perform this operation we test all pairs of possible values of *eb* and *d*, in a range from 2 to 99 (to be meaningful, both values must be greater than 1).

The criterion applied to choose the best values is the average precision AP, as described in Section 5.3.3. The experiments detected $eb = 41$ as best value of event-block, and $d = 11$ as best value of discretization (i.e., the best performance measured in all subsets involved in the *k-fold cross-validation* process).

Results

As introduced in the Sections 8.0.7 and 8.0.7, we test the proposed EBSV strategy by using a real-world dataset T related to one-year of credit card transactions, where we have added 10 fraudulent transactions, the nature of which is defined in Table 8.1. We adopt the *k-fold cross-validation* criterion, with $k = 3$, during all experiments, as specified in Sections 8.0.7.

The EBSV process generates a set of user behavioral patterns P, which we compare (i.e., using the cosine similarity metric) to the behavioral pattern related to each transaction in the subset of test, in order to retrieve a level of reliability for each of them. The final result is given by the mean value of the results of all experiments performed, in accord with the *k-fold cross-validation* criterion.

As we can observe in Figure 8.2, in spite the awareness that the experiments should be extended to other datasets in order to achieve a strong statistic relevance, the performance of the EBSV approach reachs that of the RF one, and this without train its models with the past fraudulent transactions (as occurs in RF). This is a promising result that shows how EBSV is able to operate in a proactive manner, by detecting fraudulent transactions that have never occurred in the past.

8.0.8 Conclusions and Future Work

This part of research proposed a novel approach able to reduce or eliminate the threats connected with the frauds operated in the electronic financial transactions. Differently from almost all strategies at the state of the art, instead of exploiting a

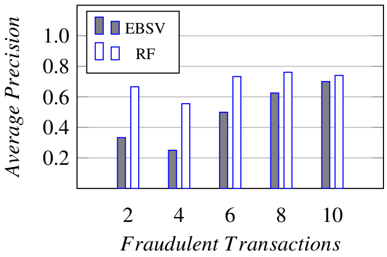

Figure 8.2: Experiment Results

unique model defined on the basis of the past transactions of the users, we adopt multiple models (behavioral patterns), in order to consider, during the evaluation of a new transaction, the user behavioral in different temporal frames of her/his history.

The possibility to adjust the levels of discretization and the size of the temporal frames, give us the opportunity to adapt the detection process to the operating environment characteristics. The most important aspect to consider is however tied to the fact that, in the proposed approach, the building of the behavioral models does not need examples of past fraudulent transactions, but is performed exclusively by exploiting the legitimate cases. This allow us to operate in a proactive manner, by detecting fraudulent transactions that have never occurred in the past, allowing also to overcome the problem of data imbalance, which afflicts the canonical machine learning approaches.

The experimental results show that the performance of the proposed *Event-block Shift Vector* approach reach those of the *Random Forests* (i.e., the state-of-the-art approach, to which we compared), and this without training the proposed models with the past fraudulent transactions (as occurs in *Random Forests*). A possible follow up of this work could be its development and evaluation in scenarios with different kind of financial transaction data, e.g., those generated in an E-commerce environment.

Conclusions

The research presented in this thesis covered three classes of information systems, usually employed in the *e-commerce* environment, whose tasks are based on the concepts of *similarity* and *diversity* (i.e., recommender systems, user segmentation systems, and fraud detection systems). Each part of this work was aimed to investigate about the open problems related with these three areas, by detecting their weaknesses and proposing several novel approaches able to overcome them, with the result to improve their performance. The main idea behind the performed research is that both the concepts of *similarity* and *diversity* must be taken in account, considering that each of them provide us important information that we can exploit in order to improve the approaches at the state of the art.

The experiments performed in the Chapter 5 of this thesis have shown how the diversity represents a primary factor to consider during the user profiling process. The advantages of the proposed approach are twofold: firstly, it moved the evaluation process of the items coherence from a domain based on strict mathematical criteria (i.e., variance of the user's ratings in the feature space) to a more flexible semantic domain. Considering that the removal of all incoherent items from the user profiles leads us toward a considerable reduction of the *magic barrier* problem, the second important result is given by the fact that we can consider each measured improvement as real, instead than a mere overfitting side effect. The experimental results show that this approach is able to reshape the user profiles in a coherent way, obtaining more accurate recommendations and a reduction of the computational complexity given by the reduced number of items in the user profiles.

Moreover, the second part of the experiments, performed in the Chapter 6, showed how it is possible to improve the performance of a recommender system at the state of the art, by post-processing its recommendations on the basis of their similarity/diversity with the most popular items in the domain taken in consideration. The proposed strategy exploits a new hybrid approach of recommendation, based on a novel algorithm called PBSVD++, which is able to extend the state-of-the-art SVD++ strategy, adding it the ability to evaluate two item popularity metrics. The performed experiments have shown both the validity of the adopted indexes, and their ability to improve the performance of the SVD++ approach. This new approach can be used in a wide range of contexts, *in primis* those related to the recommender systems which operate in a commercial environment.

The experiments performed in the user segmentation part of the thesis (Chapter 7) have shown that the introduction of a novel metric of similarity that exploits the latent semantic information about users and items, allows a system to define an effective model to use in order to improve the segmentation process, by performing a non trivial partitioning of the potential audience. The introduction of a new entity, named Semantic Binary Sieve (SBS), allows us to define a set of classes on the basis of the semantic characteristics of the items. These classes can be combined through boolean operations, in order to define in an effective way a precise target of users, simply by weighing their profiles on the basis of these SBSs.

The concepts of similarity and diversity have also been considered in the context of the fraud detection systems (Chapter 8), where they assume even greater importance, since a correct classification as *diverse* of a financial transaction (w.r.t. the past legitimate transactions of the user), allows a system to avoid a potential fraud. In this context, the performed experiments have proved how the introduction of the proposed proactive approach of fraud detection, based on the *divide and conquer* paradigm, which adopts multiple user behavioral models instead of a unique model (i.e., the typical approach at the state of the art), is able to improve the performance of a fraud detection system.

In summary, about the proposed approach for the user profiling, it was tested in the context of a recommender system by using a real-world dataset. The results showed an improvement of the $F_1 - measure$ of up to 13%, compared to a state-of-the-art approach based on the semantic similarity.

In the context of the decision making process, the introduction of the novel $PBSVD++$ algorithm led us towards an improvement of the $F_1 - measure$ of up to 20%, w.r.t. the canonic performance of a state-of-the-art approach such as $SVD++$.

Other significant results have been reported in the context of the user segmentation, where the introduction of novel binary filters (SBS) have allowed us to improve the state of the art as regards the characteristics of the created segments, decreasing their triviality and increasing their understandability. The user segmentation, performed by the proposed approach in the context of the 19 classes of characterization of the items (i.e., by our semantic approach based on the SBSs), have detected $17,464$ users instead of the $1,969$ detectable in the canonic way (i.e., by using the explicit information about the user tastes), thus about 9 times more. These additional users are pertinent in the 93.8% of the cases. Moreover, 86.3% of the additional users are correctly placed in the new classes created by applying some boolean operations between the SBSs (i.e., AND, OR, and NOT).

Even in the last context taken into account, that of the fraud detection systems, we achieved interesting results. The average precision of a fraud detection system based on the proposed approach is up to 70%, almost the same value obtained by using the state-of-the-art approach with which we compared (i.e., random forests). As a matter of fact, in spite of the limitations related with the lack of huge real datasets, the proposed strategy has been able to obtain results very close to those of a canonic state-of-the-art approach. The really important aspect is that this result was achieved by operating in a proactive way, i.e., without knowing the fraudulent transactions occurred in the past.

In conclusion, all the experimental results showed the validity of the ideas that have given life to this thesis, i.e., that an effective exploiting of both similarity and diversity can improve many approaches at the state of the art, related with several types of information systems.

Publications

The research carried out during this thesis has resulted the publications reported in the following:

- **Recommender Systems Context**:
  - **Conference publication**: R. Saia, L. Boratto, S. Carta. *Semantic Coherence-based User Profile Modeling in the Recommender Systems Context*. Proceedings of the 6th International Conference on Knowledge Discovery and Information Retrieval (KDIR), Rome, Italy, 2014.
  - **Journal publication**: R. Saia, L. Boratto, S. Carta. *A Semantic Approach to Remove Incoherent Items From a User Profile and Improve the Accuracy of a Recommender System*. Accepted for publication in Journal of Intelligent Information System, 2015.
  - **Conference publication**: R. Saia, L. Boratto, S. Carta. *Exploiting the Evaluation Frequency of the Items to Enhance the Recommendation*

Accuracy. Proceedings of the International Conference on Computer Applications and Technology (ICCAT), Rome, Italy, 2015.

  - **Journal publication**: R. Saia, L. Boratto, S. Carta. *Popularity Does Not Always Mean Triviality: Introduction of Popularity Criteria to Improve the Accuracy of a Recommender System*. International Conference on Computer Science and Information Technology (ICCSIT-2015), Amsterdam, Netherlands. Accepted for publication in Journal of Computers (JCP), 2015.
  - **Journal publication**: R. Saia, L. Boratto, S. Carta. *Introducing a Weighted Ontology to Improve the Graph-based Semantic Similarity Measures*. 6th International Conference on Networking and Information Technology (ICNIT-2015), Tokyo, Japan. Accepted for publication in International Journal of Signal Processing Systems (IJSPS), 2015.
  - **Journal publication under review**: R. Saia, L. Boratto, S. Carta. *Semantics-Aware Content-Based Recommender Systems: Design and Architecture Guidelines*. Neurocomputing, Special Issue on Recent Advances in Semantic Computing and Personalization, 2015.

- **User Segmentation Systems Context**:
  - **Conference publication**: R. Saia, L. Boratto, S. Carta. *A Latent Semantic Pattern Recognition Strategy for an Untrivial Targeted Adver-*

tising. Proceedings of the 4th IEEE International Congress (BigData), New York, United States of America, 2015.

  - **Conference publication**: R. Saia, L. Boratto, S. Carta. *A New Perspective on Recommender Systems: a Class Path Information Model*. Proceedings of the Science and Information Conference (SAI), London, United Kingdom, 2015.
  - **Journal publication**: R. Saia, L. Boratto, S. Carta. *A Class-based Strategy to User Behavior Modeling*. Accepted for publication in Studies in Computational Intelligence (SCI), Springer, 2015.
  - **Journal publication**: R. Saia, L. Boratto, S. Carta, G. Fenu. *Binary Sieves: Toward A Semantic Approach to User Segmentation for Behavioral Targeting*. Accepted for publication in "Future Generation Computer Systems - Special Issue on Semantics, Knowledge and Grids on Big Data, 2015.

- **Fraud Detection Systems Context**:

  - **Journal publication**: R. Saia, L. Boratto, S. Carta. *A Proactive Time-frame Convolution Vector (TFCV) Technique to Detect Frauds Attempts in E-commerce Transactions*. International Conference on Communication and Information Processing (ICCIP-2015), Tokyo, Japan. Accepted for publication in International Journal of e-Education, e-Business, e-Management and e-Learning (IJEEEE), 2015.

- **Conference publication**: R. Saia, L. Boratto, S. Carta. *Multiple Behavioral Models: a Divide and Conquer Strategy to Fraud Detection in Financial Data Streams*. Proceedings of the 7th International Conference on Knowledge Discovery and Information Retrieval (KDIR), Lisbon, Portugal, 2015.

Appendices

Appendix A

Natural Language Processing

A.1 Bag-of-words and Semantic Approaches

With regard to the analysis of information related to user profiles and items, there are several ways to operate and most of them work by using the *bag-of-words* model, an approach where the words are processed without taking account of the correlation between terms [23, 13].

Formalized for the first time in the fifties [107], the bag-of-words model is a representation frequently used in the *Natural Language Processing* (NLP) and *Information Retrieval* (IR) contexts. In accord with this model, a text is represented as the set (bag) of its terms, without taking in account the grammar and the word order, but keeping the multiplicity. In the recent time, this technique has also been used in the context of computer vision [1]. This model is commonly used for the document classification, a context where the frequency of each term is considered in order to train a classifier.

This trivial way to manage the information usually not leads toward good results, and just for this reason there are some more sophisticated alternatives, such as the semantic analysis of the content in order to model the preferences of a user [96]. In [42, 43, 44], the problem of modeling semantically correlated items was tackled, but the authors consider a temporal correlation and not the one between the items and a user profile.

A.2 WordNet Environment

Due to the fact that the approach based on a simple *bag of words* is not able to perform a semantic disambiguation of the words in an item description, and motivated by the fact that exploiting a taxonomy for categorization purposes is an approach recognized in the literature [79] and by the fact that a semantic analysis is useful to improve the accuracy of a classification [53, 54], in order to perform the similarity measures used in this work, we decided to exploit the functionalities offered by the WordNet environment. WordNet is a large lexical database of English, where *nouns*, *verbs*, *adjectives*, and *adverbs* are grouped into sets of cognitive synonyms (synsets), each expressing a distinct concept.

Synsets are interlinked by means of conceptual-semantic and lexical relations. Wordnet currently contains about 155,287 words, organized into 117,659 synsets for a total of 206,941 word-sense pairs [20]. The main relation among words in

WordNet is the synonymy and, in order to represent these relations, the dictionary is based on *synsets*, i.e., unordered sets of grouped words that denote the same concept and are interchangeable in many contexts. Each synset is linked to other synsets through a small number of *conceptual relations*. Word forms with several distinct meanings are represented in as many distinct s ynsets, so that each form-meaning pair in WordNet will be unique (e.g., the *fly* insect and the *fly* verb belong to two distinct synsets). Most of the WordNet relations connect words that belong to the same part-of-speech (POS). There are four POS: *nouns*, *verbs*, *adjectives*, and *adverbs*. Both nouns and verbs are organized into precise hierarchies, defined by hypernym or *is-a* relationships.

For example, the first sense of the word *radio* would have the following hypernym hierarchy, where the words at the same level are synonyms of each other: as shown in the following, some sense of *radio* is synonymous with some other senses of *radiocommunication* or *wireless*, and so on.

1. **POS=*noun***

   (a) *radio, radiocommunication, wireless (medium for communication)*

   (b) *radio receiver, receiving set, radio set, radio, tuner, wireless (an electronic receiver that detects and demodulates and amplifies transmitted signals)*

   (c) *radio, wireless (a communication system based on broadcasting electromagnetic waves)*

2. **POS=*verb***

    (a) *radio (transmit messages via radio waves)*

Each synset has a unique index and shares its properties, such as a gloss or dictionary definition. In the case of *nouns* and *verbs* (the organization of adjectives and adverbs is slightly different) the WordNet hierarchies are organized into several base types (25 primitive groups for the nouns and 15 for the verbs), and all primitive groups ultimately go up to an abstract root node. As we can imagine, the network of nouns is far deeper than that of the other *parts-of-speech*. The verbs instead present a more bushy structure, and the adjectives are distributed into many clusters, as well as the adverbs, since these last are defined in terms of the adjectives (i.e., they are derived from adjectives and thus inherit the structure from them).

Due to the similarity measure chosen for this work, we consider only the *nouns* and the *verbs*. This work exploits the state-of-art semantic-based approach to item recommendation based on the WordNet synsets [96], in order to evaluate the semantic similarity between the items not yet selected and the items already selected by users that are stored in their profiles.

A.3 Vector Space Model

Many content-based recommender systems use relatively simple retrieval models [33], such as the *Vector Space Model* (VSM), with the basic TF-IDF weighting. VSM is a spatial representation of text documents, where each document is represented by a vector in a n-dimensional space, and each dimension is related to a term from the overall vocabulary of a specific document collection.

In other words, every document is represented as a vector of term weights, where the weight indicates the degree of association between the document and the term. Let $D = \{d_1, d_2, ..., d_N\}$ indicate a set of documents, and $d = \{t_1, t_2, ..., t_N\}$, $t \in T$ be the set of terms in a document. The dictionary T is obtained by applying some standard Natural Language Processing (NLP) operations, such as tokenization, *stop-words* removal and stemming, and every document d_j is represented as a vector in a n-dimensional vector space, so $d_j = \left\{w_{1j}, w_{2j}, ..., w_{nj}\right\}$ where w_{kj} represents the weight for term t_k in document d_j .

The major problems during the document representation with the VSM are the weighting of the terms and the evaluation of the similarity of the vectors. The most commonly used way to estimate the term weighting is based on TF-IDF weighting, a trivial approach that uses empirical observations of the documents' text [104].

Appendix B

Metrics and Datasets

B.1 Metrics

B.1.1 $F_1 - measure$

The $F_1 - measure$ [108] is a combined *harmonic mean* of the *precision* and *recall* measures, used to evaluate the accuracy of a recommender system. The harmonic mean is the reciprocal of the arithmetic mean of the reciprocals. Because harmonic mean considers the reciprocals (i.e., for 2, 3 and 4, $HM = \frac{3}{\frac{1}{2}+\frac{1}{3}+\frac{1}{4}} = 2.76923$), it gives a largest weight to the smallest item and the smallest weight to the largest item. Given two sets X_u and Z_u, where X_u denotes the set of recommendations performed for a user u, and Z_u the set of the real choices of the user u in the testset, this metric is defined as shown in Equation (B.1).

$$
\begin{aligned}
&F1\text{-}Measure(X_u, Z_u) = 2\frac{(precision(X_u, Z_u) \cdot recall(X_u, Z_u))}{(precision(X_u, Z_u) + recall(X_u, Z_u))} \\
&\text{with} \\
&precision(X_u, Z_u) = \frac{|Z_u \cap X_u|}{|X_u|}, \quad recall(X_u, Z_u) = \frac{|Z_u \cap X_u|}{|Z_u|}
\end{aligned}
\tag{B.1}
$$

B.1.2 Elbow Criterion

With regard to the experiments of Chapter III, in order to detect the relevance score to take into account during the user segmentation (i.e., the threshold value after which we can consider a synset as discriminant), we use the well-known *elbow criterion*. In other words, we increase the value of the synsets occurrences and calculate the variance (as shown in Equation (B.2), where x denotes the number of users involved, and n is the number of measures performed) of the users involved: at the beginning we can note a low level of variance, but at some point the level suddenly increases (the angle in the graph); following the *elbow criterion* we chose as threshold value the number of synset occurrences used at this point.

$$
S^2 = \frac{\sum(x_i - \overline{x})}{n - 1}
\tag{B.2}
$$

B.1.3 Cosine Similarity

Cosine similarity is a measure of similarity between two vectors of an inner product space. It represents the cosine measure of the angle between them. Considering that the cosine of 0° is 1, and it is less than 1 for any other angle, in two vectors with the same orientation we measure a cosine similarity of 1. The output of this measure is then bounded in $[0, 1]$, with 0 that means complete diversity, and 1 complete similarity. Given two vectors of attributes x and y (i.e., the behavioral patterns), the cosine similarity, $\cos(\theta)$, is represented using a dot product and magnitude as shown in Equation B.3.

$$\text{similarity} = \cos(\theta) = \frac{x \cdot y}{||x||||y||} = \frac{\sum_{i=1}^{n} x_i \times y_i}{\sqrt{\sum_{i=1}^{n} (x_i)^2} \times \sqrt{\sum_{i=1}^{n} (y_i)^2}} \tag{B.3}$$

B.1.4 Levenshtein Distance

The *Levenshtein Distance* is a metric able to measure the difference between two sequences of terms. Given two strings a and b, it indicates the minimal number of insertions, deletions, and replacements, needed to transforming the string a into the string b. Denoting as $|a|$ and $|b|$ the length of the strings a and b, the *Levenshtein Distance* is given by $\text{lev}_{a,b}(|a|, |b|)$, as shown in Equation B.4.

$$\text{lev}_{a,b}(i, j) = \begin{cases} \max(i, j) & \text{if} \min(i, j) = 0 \\ \min \begin{cases} \text{lev}_{a,b}(i-1, j) + 1 \\ \text{lev}_{a,b}(i, j-1) + 1 \\ \text{lev}_{a,b}(i-1, j-1) + 1_{(a_i \neq b_j)} \end{cases} & \text{otherwise} \end{cases} \tag{B.4}$$

Where $1_{(a_i \neq b_j)}$ is the *indicator function* equal to 0 when $a_i = b_j$ and equal to 1 otherwise. It should be noted that the first element in the minimum corresponds to deletion (from a to b), the second to insertion and the third to match or mismatch, depending on whether the respective symbols are the same.

B.1.5 Average Precision

The average precision (AP) is considered as the correct measure to use in the fraud detection context, as described in Definition 8.2. Given N the number of transactions in the dataset, $\Delta Recall(t_r) = Recall(t_r) - Recall(t_r - 1)$, π the number of fraudulent transactions in the dataset (out of the percent t of top-ranked candidates), $h(t) \leq t$ the truly relevant transactions, $Recall(t) = h(t)/\pi$, and $Precision(t) = h(t)/t$, we can obtain the AP value as shown in Equation B.5.

$$AP = \sum_{r=1}^{N} Precision(t_r) \Delta Recall(t_r) \tag{B.5}$$

B.2 Datasets

B.2.1 Yahoo! Webscope Movie Dataset (R4)

The Yahoo! Webscope Movie dataset (R4)[1] is a dataset that contains a large amount of data related to users preferences expressed on the Yahoo! Movies community that are rated on the base of two different scales, from 1 to 13 and from 1

[1] http://webscope.sandbox.yahoo.com

to 5 (we have chosen to use the latter). The training data is composed by $7,642$ users (| U |), $11,915$ movies/items (| I |), and 211,231 ratings (| R |). The test data is composed by $2,309$ users, $2,380$ items, and $10,136$ ratings. There are no test users/items that do not also appear in the training data. All the users in the test set have rated at least one item and all items have been rated by at least one user. The items are classified in 20 different classes (genres), and it should be noted that an item may be classified with multiple classes. As shown in Table B.1, the items are classified by Yahoo in 20 different classes (movie genres), and it is should be noted that each item may be classified in multiple classes.

Class	Genre	Class	Genre
01	Action/Adventure	11	Musical/Performing Arts
02	Adult Audience	12	Other
03	Animation	13	Reality
04	Art/Foreign	14	Romance
05	Comedy	15	Science Fiction/Fantasy
06	Crime/Gangster	16	Special Interest
07	Documentary	17	Suspense/Horror
08	Drama	18	Thriller
09	Kids/Family	19	Western
10	Miscellaneous		

Table B.1: Yahoo! Webscope R4 Genres

B.2.2 Movielens 10M

The Movielens 10M[2] dataset is composed by $71,567$ users ($|U|$), $10,681$ movies/items ($|I|$), and $10,000,054$ ratings ($|P|$). It was extracted at random from MovieLens

[2] http://grouplens.org/datasets/movielens/

(a movie recommendation website). All the users in the dataset had rated at least 20 movies, and each user is represented by a unique ID. The ratings of the items are based on a *5-star* scale, with *half-star* increments. As shown in Table B.2, in this dataset the items are classified in 18 different classes (movie genres), and also in this case each item may be classified with multiple classes (genres). Since the Movielens 10M dataset does not contain any textual description of the items, to obtain this information we used a file provided by the Webscope (R4) dataset, which contains a mapping from the movie IDs used in the dataset to the corresponding movie IDs and titles used in the MovieLens dataset.

Class	Genre	Class	Genre
01	Action	10	Film-Noir
02	Adventure	11	Horror
03	Animation	12	Musical
04	Children's	13	Mystery
05	Comedy	14	Romance
06	Crime	15	Sci-Fi
07	Documentary	16	Thriller
08	Drama	17	War
09	Fantasy	18	Western

Table B.2: Movielens 10M Genres

B.2.3 Credit Card Dataset

It is composed by 204 transactions, operated from January 2014 to December 2014, with amounts in the range from 1.00 to 591.38 Euro, 55 different descrip-

NR	*Field*	*Explanation*	*Type*
1	TID	Transaction ID	Numeric
2	Description	Type of transaction	Textual
3	Place	City of transaction	Textual
4	Date	Date of transaction	Date
5	Amount	Amount in Euro	Currency

Table B.3: Transaction Fields

tions of expense, and 7 places of operation (when the transaction is operated online, the *place* reported is *Internet*). Considering that all transactions in the dataset are legal, we have $T_+ = 204$ and $T_- = 0$. As shown in Table B.3, the fields that compose a transaction are 5, but in this work we do not take in account the *Transaction ID* field (TID), nor any metadata (e.g., mean value of expenditure per week or month).

Bibliography

[1] F. Ricci, L. Rokach, and B. Shapira, "Introduction to recommender systems handbook," in *Recommender Systems Handbook*, F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, Eds. Springer, 2011, pp. 1–35.

[2] J. B. Schafer, J. A. Konstan, and J. Riedl, "Recommender systems in e-commerce," in *Proceedings of the 1st ACM conference on Electronic commerce*, 1999, pp. 158–166.

[3] C. Wei, R. Khoury, and S. Fong, "Recommendation systems for web 2.0 marketing," in *Data Mining for Service*, ser. Studies in Big Data, K. Yada, Ed. Springer Berlin Heidelberg, 2014, vol. 3, pp. 171–196.

[4] G. Armano and E. Vargiu, "A unifying view of contextual advertising and recommender systems," in *KDIR 2010 - Proceedings of the International Conference on Knowledge Discovery and Information Retrieval*, A. L. N. Fred and J. Filipe, Eds. SciTePress, 2010, pp. 463–466.

[5] A. Addis, G. Armano, A. Giuliani, and E. Vargiu, “A recommender system based on a generic contextual advertising approach,” in *Proceedings of the 15th IEEE Symposium on Computers and Communications, ISCC 2010, Riccione, Italy, June 22-25, 2010*. IEEE, 2010, pp. 859–861.

[6] E. Vargiu, A. Giuliani, and G. Armano, “Improving contextual advertising by adopting collaborative filtering,” *ACM Trans. Web*, vol. 7, no. 3, pp. 13:1–13:22, Sep. 2013.

[7] X. Shen, B. Tan, and C. Zhai, “Implicit user modeling for personalized search,” in *Proceedings of the 2005 ACM CIKM International Conference on Information and Knowledge Management, Bremen, Germany, October 31 - November 5, 2005*, O. Herzog, H.-J. Schek, N. Fuhr, A. Chowdhury, and W. Teiken, Eds. ACM, 2005, pp. 824–831.

[8] J. Budzik and K. J. Hammond, “User interactions with everyday applications as context for just-in-time information access,” in *Proceedings of the 5th International Conference on Intelligent User Interfaces*, ser. IUI ’00. New York, NY, USA: ACM, 2000, pp. 44–51.

[9] L. Finkelstein, E. Gabrilovich, Y. Matias, E. Rivlin, Z. Solan, G. Wolfman, and E. Ruppin, “Placing search in context: The concept revisited,” *ACM Trans. Inf. Syst.*, vol. 20, no. 1, pp. 116–131, Jan. 2002.

[10] P. A. Chirita, W. Nejdl, R. Paiu, and C. Kohlschütter, “Using odp metadata to personalize search,” in *Proceedings of the 28th Annual International*

ACM SIGIR Conference on Research and Development in Information Retrieval, ser. SIGIR '05. New York, NY, USA: ACM, 2005, pp. 178–185.

[11] F. A. Asnicar and C. Tasso, "ifweb: a prototype of user model-based intelligent agent for document filtering and navigation in the world wide web," in *Proceedings of Workshop Adaptive Systems and User Modeling on the World Wide Web'at 6th International Conference on User Modeling, UM97, Chia Laguna, Sardinia, Italy*, 1997, pp. 3–11.

[12] Z. Ma, G. Pant, and O. R. L. Sheng, "Interest-based personalized search," *ACM Trans. Inf. Syst.*, vol. 25, no. 1, Feb. 2007.

[13] D. H. Widyantoro, T. R. Ioerger, and J. Yen, "Learning user interest dynamics with a three-descriptor representation," *JASIST*, vol. 52, no. 3, pp. 212–225, 2001.

[14] F. Liu, C. Yu, and W. Meng, "Personalized web search by mapping user queries to categories," in *Proceedings of the Eleventh International Conference on Information and Knowledge Management*, ser. CIKM '02. New York, NY, USA: ACM, 2002, pp. 558–565.

[15] A. Pretschner and S. Gauch, "Ontology based personalized search," in *11th IEEE International Conference on Tools with Artificial Intelligence, ICTAI '99, Chicago, Illinois, USA, November 8-10, 1999*. IEEE Computer Society, 1999, pp. 391–398. [Online]. Available: http://dx.doi.org/10.1109/TAI.1999.809829

[16] D. C. C. Poo, B. Chng, and J. Goh, “A hybrid approach for user profiling,” in *36th Hawaii International Conference on System Sciences (HICSS-36 2003), CD-ROM Abstracts Proceedings, January 6-9, 2003, Big Island, HI, USA*. IEEE Computer Society, 2003, p. 103. [Online]. Available: http://dx.doi.org/10.1109/HICSS.2003.1174242

[17] S. Braynov, “Personalization and customization technologies,” *The Internet Encyclopedia*, 2003.

[18] M. Degemmis, P. Lops, G. Semeraro, and P. Basile, “Integrating tags in a semantic content-based recommender,” in *Proceedings of the 2008 ACM Conference on Recommender Systems, RecSys 2008, Lausanne, Switzerland, October 23-25, 2008*, P. Pu, D. G. Bridge, B. Mobasher, and F. Ricci, Eds. ACM, 2008, pp. 163–170. [Online]. Available: http://doi.acm.org/10.1145/1454008.1454036

[19] R. D. Burke, “Hybrid recommender systems: Survey and experiments,” *User Model. User-Adapt. Interact.*, vol. 12, no. 4, pp. 331–370, 2002. [Online]. Available: http://dx.doi.org/10.1023/A:1021240730564

[20] C. Fellbaum, *WordNet: An Electronic Lexical Database*. Bradford Books, 1998.

[21] G. A. Miller, “Wordnet: A lexical database for english,” *Commun. ACM*, vol. 38, no. 11, pp. 39–41, 1995. [Online]. Available: http://doi.acm.org/10.1145/219717.219748

[22] D. Godoy and A. Amandi, "Hybrid content and tag-based profiles for recommendation in collaborative tagging systems," in *Proceedings of the Latin American Web Conference, LA-WEB 2008, October 28-30, 2008, Vila Velha, Espírito Santo, Brasil*, R. A. Baeza-Yates, W. M. Jr., and L. A. O. Santos, Eds. IEEE Computer Society, 2008, pp. 58–65. [Online]. Available: http://dx.doi.org/10.1109/LA-WEB.2008.15

[23] W. Lam, S. Mukhopadhyay, J. Mostafa, and M. J. Palakal, "Detection of shifts in user interests for personalized information filtering," in *SIGIR*, 1996, pp. 317–325.

[24] D. H. Widyantoro, T. R. Ioerger, and J. Yen, "An adaptive algorithm for learning changes in user interests," in *Proceedings of the 1999 ACM CIKM International Conference on Information and Knowledge Management, Kansas City, Missouri, USA, November 2-6, 1999*. ACM, 1999, pp. 405–412. [Online]. Available: http://doi.acm.org/10.1145/319950.323230

[25] V. Schickel-Zuber and B. Faltings, "Inferring user's preferences using ontologies," in *Proceedings, The Twenty-First National Conference on Artificial Intelligence and the Eighteenth Innovative Applications of Artificial Intelligence Conference, July 16-20, 2006, Boston, Massachusetts, USA*. AAAI Press, 2006, pp. 1413–1418.

[26] D. Kelly and J. Teevan, "Implicit feedback for inferring user preference: a bibliography," *SIGIR Forum*, vol. 37, no. 2, pp. 18–28, 2003.

[27] M. Zeb and M. Fasli, “Adaptive user profiling for deviating user interests,” in *Computer Science and Electronic Engineering Conference (CEEC), 2011 3rd*, July 2011, pp. 65–70.

[28] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. Riedl, “Evaluating collaborative filtering recommender systems,” *ACM Trans. Inf. Syst.*, vol. 22, no. 1, pp. 5–53, 2004.

[29] A. Said, B. J. Jain, S. Narr, T. Plumbaum, S. Albayrak, and C. Scheel, “Estimating the magic barrier of recommender systems: a user study,” in *The 35th International ACM SIGIR conference on research and development in Information Retrieval, SIGIR ’12, Portland, OR, USA, August 12-16, 2012*, W. R. Hersh, J. Callan, Y. Maarek, and M. Sanderson, Eds. ACM, 2012, pp. 1061–1062.

[30] W. C. Hill, L. Stead, M. Rosenstein, and G. W. Furnas, “Recommending and evaluating choices in a virtual community of use,” in *Human Factors in Computing Systems, CHI ’95 Conference Proceedings, Denver, Colorado, USA, May 7-11, 1995.*, I. R. Katz, R. L. Mack, L. Marks, M. B. Rosson, and J. Nielsen, Eds. ACM/Addison-Wesley, 1995, pp. 194–201.

[31] X. Amatriain, J. M. Pujol, N. Tintarev, and N. Oliver, “Rate it again: increasing recommendation accuracy by user re-rating,” in *Proceedings of the 2009 ACM Conference on Recommender Systems, RecSys 2009, New York,*

NY, USA, October 23-25, 2009, L. D. Bergman, A. Tuzhilin, R. D. Burke, A. Felfernig, and L. Schmidt-Thieme, Eds. ACM, 2009, pp. 173–180.

[32] X. Amatriain, J. M. Pujol, and N. Oliver, “I like it... I like it not: Evaluating user ratings noise in recommender systems,” in *User Modeling, Adaptation, and Personalization, 17th International Conference, UMAP 2009, formerly UM and AH, Trento, Italy, June 22-26, 2009. Proceedings*, ser. Lecture Notes in Computer Science, G. Houben, G. I. McCalla, F. Pianesi, and M. Zancanaro, Eds., vol. 5535. Springer, 2009, pp. 247–258.

[33] P. Lops, M. de Gemmis, and G. Semeraro, “Content-based recommender systems: State of the art and trends,” in *Recommender Systems Handbook*, F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, Eds. Springer, 2011, pp. 73–105.

[34] M. J. Pazzani and D. Billsus, “Content-based recommendation systems,” in *The Adaptive Web*, P. Brusilovsky, A. Kobsa, and W. Nejdl, Eds. Berlin, Heidelberg: Springer-Verlag, 2007, pp. 325–341. [Online]. Available: http://dl.acm.org/citation.cfm?id=1768197.1768209

[35] M. Balabanovic and Y. Shoham, “Content-based, collaborative recommendation,” *Commun. ACM*, vol. 40, no. 3, pp. 66–72, 1997.

[36] D. Billsus and M. J. Pazzani, “A hybrid user model for news story classification,” in *Proceedings of the Seventh International Conference*

on User Modeling, ser. UM '99. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 1999, pp. 99–108. [Online]. Available: http://dl.acm.org/citation.cfm?id=317328.317338

[37] H. Lieberman, "Letizia: An agent that assists web browsing," in *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 1*, ser. IJCAI'95. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1995, pp. 924–929. [Online]. Available: http://dl.acm.org/citation.cfm?id=1625855.1625975

[38] M. Pazzani, J. Muramatsu, and D. Billsus, "Syskill &#38; webert: Identifying interesting web sites," in *Proceedings of the Thirteenth National Conference on Artificial Intelligence - Volume 1*, ser. AAAI'96. AAAI Press, 1996, pp. 54–61. [Online]. Available: http://dl.acm.org/citation.cfm?id=1892875.1892883

[39] M. Capelle, F. Frasincar, M. Moerland, and F. Hogenboom, "Semantics-based news recommendation," in *Proceedings of the 2Nd International Conference on Web Intelligence, Mining and Semantics*, ser. WIMS '12. New York, NY, USA: ACM, 2012, pp. 27:1–27:9.

[40] M. Capelle, F. Hogenboom, A. Hogenboom, and F. Frasincar, "Semantic news recommendation using wordnet and bing similarities," in *Proceedings of the 28th Annual ACM Symposium on Applied Computing*, ser. SAC '13. New York, NY, USA: ACM, 2013, pp. 296–302.

[41] P. Basile, C. Musto, M. de Gemmis, P. Lops, F. Narducci, and G. Semeraro, "Content-based recommender systems + dbpedia knowledge = semantics-aware recommender systems," in *Semantic Web Evaluation Challenge - SemWebEval 2014 at ESWC 2014, Anissaras, Crete, Greece, May 25-29, 2014, Revised Selected Papers*, ser. Communications in Computer and Information Science, V. Presutti, M. Stankovic, E. Cambria, I. Cantador, A. D. Iorio, T. D. Noia, C. Lange, D. R. Recupero, and A. Tordai, Eds., vol. 475. Springer, 2014, pp. 163–169.

[42] G. Stilo and P. Velardi, "Time makes sense: Event discovery in twitter using temporal similarity," in *Proceedings of the 2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT) - Volume 02*, ser. WI-IAT '14. Washington, DC, USA: IEEE Computer Society, 2014, pp. 186–193.

[43] ——, "Temporal semantics: Time-varying hashtag sense clustering," in *Knowledge Engineering and Knowledge Management*, ser. Lecture Notes in Computer Science. Springer International Publishing, 2014, vol. 8876, pp. 563–578.

[44] ——, "Efficient temporal mining of micro-blog texts and its application to event discovery," *Data Mining and Knowledge Discovery*, 2015.

[45] P. Cremonesi, Y. Koren, and R. Turrin, "Performance of recommender algorithms on top-n recommendation tasks," in *Proceedings of the 2010 ACM*

Conference on Recommender Systems, RecSys 2010, X. Amatriain, M. Torrens, P. Resnick, and M. Zanker, Eds. ACM, 2010, pp. 39–46.

[46] L. Iaquinta, M. de Gemmis, P. Lops, G. Semeraro, M. Filannino, and P. Molino, "Introducing serendipity in a content-based recommender system," in *HIS*. IEEE Computer Society, 2008, pp. 168–173.

[47] C. V. Yehuda Koren, Robert M. Bell, "Matrix factorization techniques for recommender systems," *IEEE Computer*, vol. 42, no. 8, pp. 30–37, 2009.

[48] Y. Koren and J. Sill, "Ordrec: an ordinal model for predicting personalized item rating distributions," in *Proceedings of the 2011 ACM Conference on Recommender Systems, RecSys 2011*, B. Mobasher, R. D. Burke, D. Jannach, and G. Adomavicius, Eds. ACM, 2011, pp. 117–124.

[49] Y. Koren, "Factorization meets the neighborhood: a multifaceted collaborative filtering model," in *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Y. Li, B. Liu, and S. Sarawagi, Eds. ACM, 2008, pp. 426–434.

[50] J. Yan, N. Liu, G. Wang, W. Zhang, Y. Jiang, and Z. Chen, "How much can behavioral targeting help online advertising?" in *Proceedings of the 18th International Conference on World Wide Web*, ser. WWW '09. New York, NY, USA: ACM, 2009, pp. 261–270.

[51] H. Beales, “The value of behavioral targeting,” *Network Advertising Initiative*, 2010.

[52] Y. Chen, D. Pavlov, and J. F. Canny, “Large-scale behavioral targeting,” in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD ’09. New York, NY, USA: ACM, 2009, pp. 209–218.

[53] G. Armano, A. Giuliani, and E. Vargiu, “Semantic enrichment of contextual advertising by using concepts,” in *KDIR 2011 - Proceedings of the International Conference on Knowledge Discovery and Information Retrieval, Paris, France, 26-29 October, 2011*, J. Filipe and A. L. N. Fred, Eds. SciTePress, 2011, pp. 232–237.

[54] ——, “Studying the impact of text summarization on contextual advertising,” in *2011 Database and Expert Systems Applications, DEXA, International Workshops, Toulouse, France, August 29 - Sept. 2, 2011*, F. Morvan, A. M. Tjoa, and R. Wagner, Eds. IEEE Computer Society, 2011, pp. 172–176.

[55] A. Spink, B. J. Jansen, D. Wolfram, and T. Saracevic, “From e-sex to e-commerce: Web search changes,” *Computer*, vol. 35, no. 3, pp. 107–109, Mar. 2002.

[56] S. Y. Rieh and H. I. Xie, “Analysis of multiple query reformulations on the web: The interactive information retrieval context,” *Inf. Process. Manage.*, vol. 42, no. 3, pp. 751–768, May 2006.

[57] P. Boldi, F. Bonchi, C. Castillo, and S. Vigna, “From ”dango” to ”japanese cakes”: Query reformulation models and patterns,” in *Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology - Volume 01*, ser. WI-IAT ’09. Washington, DC, USA: IEEE Computer Society, 2009, pp. 183–190.

[58] R. Saia, L. Boratto, and S. Carta, “Semantic coherence-based user profile modeling in the recommender systems context,” in *Proceedings of the 6th International Conference on Knowledge Discovery and Information Retrieval, KDIR 2014, Rome, Italy, October 21-24, 2014*. SciTePress, 2014, pp. 154–161.

[59] J. Mazanee, “Market Segmentation,” in *Encyclopedia of Tourism*. London: Routledge, 2000.

[60] S. Dolničar, “Beyond “commonsense segmentation”: A systematics of segmentation approaches in tourism,” *Journal of Travel Research*, vol. 42, no. 3, pp. 244–250, 2004.

[61] S. C. Bourassa, F. Hamelink, M. Hoesli, and B. D. MacGregor, “Defining housing submarkets,” *Journal of Housing Economics*, vol. 8, no. 2, pp. 160 – 183, 1999.

[62] J. H. Myers and E. M. Tauber, *Market Structure Analysis*. American Marketing Association, 1977.

[63] M. Wedel and W. A. Kamakura, *Market Segmentation: Conceptual and Methodological Foundations (International Series in Quantitative Marketing)*. Kluwer Academic Publishers, 2000.

[64] A. Nairn and P. Bottomley, "Something approaching science? cluster analysis procedures in the crm era," in *Proceedings of the 2002 Academy of Marketing Science (AMS) Annual Conference*, ser. Developments in Marketing Science: Proceedings of the Academy of Marketing Science. Springer International Publishing, 2003, pp. 120–120.

[65] S. Dolnicar and K. Lazarevski, "Methodological reasons for the theory/practice divide in market segmentation," *Journal of Marketing Management*, vol. 25, no. 3-4, pp. 357–373, 2009.

[66] S. Dibb and L. Simkin, "A program for implementing market segmentation," *Journal of Business & Industrial Marketing*, vol. 12, no. 1, pp. 51–65, 1997.

[67] J. Bian, A. Dong, X. He, S. Reddy, and Y. Chang, "User action interpretation for online content optimization," *IEEE Trans. on Knowl. and Data Eng.*, vol. 25, no. 9, pp. 2161–2174, Sep. 2013.

[68] Z. Yao, T. Eklund, and B. Back, "Using som-ward clustering and predictive analytics for conducting customer segmentation," in *Proceedings of the 2010 IEEE International Conference on Data Mining Workshops*, ser. ICDMW '10. Washington, DC, USA: IEEE Computer Society, 2010, pp. 639–646.

[69] Y. K. Zhou and B. Mobasher, "Web user segmentation based on a mixture of factor analyzers," in *Proceedings of the 7th International Conference on E-Commerce and Web Technologies*, ser. EC-Web'06. Berlin, Heidelberg: Springer-Verlag, 2006, pp. 11–20.

[70] S. Tu and C. Lu, "Topic-based user segmentation for online advertising with latent dirichlet allocation," in *Proceedings of the 6th International Conference on Advanced Data Mining and Applications - Volume Part II*, ser. ADMA'10. Berlin, Heidelberg: Springer-Verlag, 2010, pp. 259–269.

[71] X. Gong, X. Guo, R. Zhang, X. He, and A. Zhou, "Search behavior based latent semantic user segmentation for advertising targeting," in *Data Mining (ICDM), 2013 IEEE 13th International Conference on*, Dec 2013, pp. 211–220.

[72] X. Wu, J. Yan, N. Liu, S. Yan, Y. Chen, and Z. Chen, "Probabilistic latent semantic user segmentation for behavioral targeted advertising," in *Proceedings of the Third International Workshop on Data Mining and Audi-*

ence Intelligence for Advertising, ser. ADKDD ’09. New York, NY, USA: ACM, 2009, pp. 10–17.

[73] R. D. Burke and M. Ramezani, “Matching recommendation technologies and domains,” in *Recommender Systems Handbook*, F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, Eds. Springer, 2011, pp. 367–386. [Online]. Available: http://www.springerlink.com/content/978-0-387-85819-7

[74] S. A. Munson and P. Resnick, “Presenting diverse political opinions: How and how much,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI ’10. New York, NY, USA: ACM, 2010, pp. 1457–1466. [Online]. Available: http://doi.acm.org/10.1145/1753326.1753543

[75] E. Pariser, *The Filter Bubble: What the Internet Is Hiding from You*. Penguin Group , The, 2011.

[76] L. Festinger, *A theory of cognitive dissonance*. Stanford university press, 1962, vol. 2.

[77] S. Park, S. Kang, S. Chung, and J. Song, “Newscube: delivering multiple aspects of news to mitigate media bias,” in *Proceedings of the 27th International Conference on Human Factors in Computing Systems, CHI 2009, Boston, MA, USA, April 4-9, 2009*. ACM, 2009.

[78] G. Salton and C. Buckley, “Term-weighting approaches in automatic text retrieval,” *Inf. Process. Manage.*, vol. 24, no. 5, pp. 513–523, Aug. 1988. [Online]. Available: http://dx.doi.org/10.1016/0306-4573(88)90021-0

[79] A. Addis, G. Armano, and E. Vargiu, “Assessing progressive filtering to perform hierarchical text categorization in presence of input imbalance,” in *KDIR 2010 - Proceedings of the International Conference on Knowledge Discovery and Information Retrieval, Valencia, Spain, October 25-28, 2010*, A. L. N. Fred and J. Filipe, Eds. SciTePress, 2010, pp. 14–23.

[80] C. Assis, A. M. Pereira, M. de Arruda Pereira, E. G. Carrano, C. Phua, V. C. S. Lee, K. Smith-Miles, and R. W. Gayler, “Using genetic programming to detect fraud in electronic transactions,” in *A Comprehensive Survey of Data Mining-based Fraud Detection Research*, C. V. S. Prazeres, P. N. M. Sampaio, A. Santanchè, C. A. S. Santos, and R. Goularte, Eds., vol. abs/1009.6119, 2010, pp. 337–340.

[81] C. Phua, V. C. S. Lee, K. Smith-Miles, and R. W. Gayler, “A comprehensive survey of data mining-based fraud detection research,” *CoRR*, vol. abs/1009.6119, 2010.

[82] R. J. Bolton and D. J. Hand, “Statistical fraud detection: A review,” *Statistical Science*, pp. 235–249, 2002.

[83] S. Bhattacharyya, S. Jha, K. K. Tharakunnel, and J. C. Westland, "Data mining for credit card fraud: A comparative study," *Decision Support Systems*, vol. 50, no. 3, pp. 602–613, 2011.

[84] G. E. A. P. A. Batista, A. C. P. L. F. de Carvalho, and M. C. Monard, "Applying one-sided selection to unbalanced datasets," in *MICAI 2000: Advances in Artificial Intelligence, Mexican International Conference on Artificial Intelligence, Acapulco, Mexico, April 11-14, 2000, Proceedings*, ser. Lecture Notes in Computer Science, O. Cairó, L. E. Sucar, and F. J. Cantu, Eds., vol. 1793. Springer, 2000, pp. 315–325.

[85] N. Japkowicz and S. Stephen, "The class imbalance problem: A systematic study," *Intell. Data Anal.*, vol. 6, no. 5, pp. 429–449, 2002.

[86] C. Drummond, R. C. Holte *et al.*, "C4. 5, class imbalance, and cost sensitivity: why under-sampling beats over-sampling," in *Workshop on learning from imbalanced datasets II*, vol. 11. Citeseer, 2003.

[87] M. Wasikowski and X. Chen, "Combating the small sample class imbalance problem using feature selection," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1388–1400, 2010. [Online]. Available: http://dx.doi.org/10.1109/TKDE.2009.187

[88] A. D. Pozzolo, O. Caelen, Y. L. Borgne, S. Waterschoot, and G. Bontempi, "Learned lessons in credit card fraud detection from a practitioner perspective," *Expert Syst. Appl.*, vol. 41, no. 10, pp. 4915–4928, 2014.

[89] H. Wang, W. Fan, P. S. Yu, and J. Han, "Mining concept-drifting data streams using ensemble classifiers," in *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, August 24 - 27, 2003*, L. Getoor, T. E. Senator, P. M. Domingos, and C. Faloutsos, Eds. ACM, 2003, pp. 226–235.

[90] J. Gao, W. Fan, J. Han, and P. S. Yu, "A general framework for mining concept-drifting data streams with skewed distributions," in *Proceedings of the Seventh SIAM International Conference on Data Mining, April 26-28, 2007, Minneapolis, Minnesota, USA*. SIAM, 2007, pp. 3–14.

[91] M. Lek, B. Anandarajah, N. Cerpa, and R. Jamieson, "Data mining prototype for detecting e-commerce fraud," in *Proceedings of the 9th European Conference on Information Systems, Global Co-operation in the New Millennium, ECIS 2001, Bled, Slovenia, June 27-29, 2001*, S. Smithson, J. Gricar, M. Podlogar, and S. Avgerinou, Eds., 2001, pp. 160–165.

[92] A. J. Hoffman and R. E. Tessendorf, "Artificial intelligence based fraud agent to identify supply chain irregularities," in *IASTED International Conference on Artificial Intelligence and Applications, part of the 23rd Multi-Conference on Applied Informatics, Innsbruck, Austria, February 14-16, 2005*, M. H. Hamza, Ed. IASTED/ACTA Press, 2005, pp. 743–750.

[93] M. J. Lenard and P. Alam, "Application of fuzzy logic fraud detection," in *Encyclopedia of Information Science and Technology (5 Volumes)*,

M. Khosrow-Pour, Ed. Idea Group, 2005, pp. 135–139.

[94] D. G. Whiting, J. V. Hansen, J. B. McDonald, C. C. Albrecht, and W. S. Albrecht, "Machine learning methods for detecting patterns of management fraud," *Computational Intelligence*, vol. 28, no. 4, pp. 505–527, 2012.

[95] R. C. Holte, L. Acker, and B. W. Porter, "Concept learning and the problem of small disjuncts," in *Proceedings of the 11th International Joint Conference on Artificial Intelligence. Detroit, MI, USA, August 1989*, N. S. Sridharan, Ed. Morgan Kaufmann, 1989, pp. 813–818.

[96] T. Pedersen, S. Patwardhan, and J. Michelizzi, "Wordnet::similarity: Measuring the relatedness of concepts," in *Demonstration Papers at HLT-NAACL 2004*, ser. HLT-NAACL–Demonstrations '04. Stroudsburg, PA, USA: Association for Computational Linguistics, 2004, pp. 38–41.

[97] C. Leacock and M. Chodorow, "Combining local context and wordnet similarity for word sense identification," in *WordNet: An Electronic Lexical Database*, C. Fellbaum, Ed. MIT Press, 1998, pp. 305–332.

[98] J. J. Jiang and D. W. Conrath, "Semantic similarity based on corpus statistics and lexical taxonomy," *CoRR*, vol. cmp-lg/9709008, 1997.

[99] P. Resnik, "Using information content to evaluate semantic similarity in a taxonomy," in *Proceedings of the 14th International Joint Conference on*

Artificial Intelligence - Volume 1, ser. IJCAI'95. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1995, pp. 448–453.

[100] D. Lin, "An information-theoretic definition of similarity," in *Proceedings of the Fifteenth International Conference on Machine Learning (ICML 1998)*, J. W. Shavlik, Ed. Morgan Kaufmann, 1998, pp. 296–304.

[101] Z. Wu and M. Palmer, "Verbs semantics and lexical selection," in *Proceedings of the 32Nd Annual Meeting on Association for Computational Linguistics*, ser. ACL '94. Stroudsburg, PA, USA: Association for Computational Linguistics, 1994, pp. 133–138.

[102] A. Bellogín, A. Said, and A. P. de Vries, "The magic barrier of recommender systems - no magic, just ratings," in *User Modeling, Adaptation, and Personalization - 22nd International Conference, UMAP 2014, Aalborg, Denmark, July 7-11, 2014. Proceedings*, ser. Lecture Notes in Computer Science, V. Dimitrova, T. Kuflik, D. Chin, F. Ricci, P. Dolog, and G. Houben, Eds., vol. 8538. Springer, 2014, pp. 25–36.

[103] K. Toutanova, D. Klein, C. D. Manning, and Y. Singer, "Feature-rich part-of-speech tagging with a cyclic dependency network," in *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, ser. NAACL '03. Stroudsburg, PA, USA: Association for Computational Linguistics, 2003, pp. 173–180.

[104] G. Salton, A. Wong, and C. S. Yang, “A vector space model for automatic indexing,” *Commun. ACM*, vol. 18, no. 11, pp. 613–620, 1975.

[105] A. Dennai and S. M. Benslimane, “Toward an update of a similarity measurement for a better calculation of the semantic distance between ontology concepts,” in *The Second International Conference on Informatics Engineering & Information Science (ICIEIS2013).* The Society of Digital Information and Wireless Communication, 2013, pp. 197–207.

[106] G. Fan and M. Zhu, “Detection of rare items with target,” *Statistics and Its Interface*, vol. 4, pp. 11–17, 2011.

[107] Z. S. Harris, “Distributional structure.” *Word*, 1954.

[108] R. A. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1999.