Noname manuscript No.
(will be inserted by the editor)

A Semantic Approach to Remove Incoherent Items From a User Profile and Improve the Accuracy of a Recommender System

Roberto Saia · Ludovico Boratto · Salvatore Carta

Received: 29 July 2015 / Accepted: 31 March 2016

Abstract Recommender systems usually suggest items by exploiting all the previous interactions of the users with a system (e.g., in order to decide the movies to recommend to a user, all the movies she previously purchased are considered). This canonical approach sometimes could lead to wrong results due to several factors, such as a change in user preferences over time, or the use of her account by third parties. This kind of incoherence in the user profiles defines a lower bound on the error the recommender systems may achieve when they generate suggestions for a user, an aspect known in literature as *magic barrier*. This paper proposes a novel dynamic coherence-based approach to define the user profile used in the recommendation process. The main aim is to identify and remove, from the previously evaluated items, those not semantically adherent to the others, in order to make a user profile as close as possible to the user's real preferences, solving the aforementioned problems. Moreover, reshaping the user profile in such a way leads to great advantages in terms of computational complexity, since the number of items considered during the recommendation process is highly reduced. The performed experiments show the effectiveness of our approach to remove the incoherent items from a user profile, increasing the recommendation accuracy.

Keywords User Profiling · Semantic Analysis · Magic Barrier · Accuracy

This work is partially funded by Regione Sardegna under project NOMAD (Next generation Open Mobile Apps Development), through PIA - Pacchetti Integrati di Agevolazione "Industria Artigianato e Servizi" (annualità 2013), and by MIUR PRIN 2010-11 under project "Security Horizons".

Dipartimento di Matematica e Informatica - Università di Cagliari
Via Ospedale 72 - 09124 Cagliari, Italy
E-mail: {roberto.saia, ludovico.boratto, salvatore}@unica.it

1 Introduction

The rapid growth of the number of companies that sell goods through the Word Wide Web generates an enormous amount of valuable information, which can be exploited to improve the quality and efficiency of the sales criteria [48]. Because of the widely-known information overload problem, it became necessary to deal with the large amounts of data available on the Web [57]. *Recommender Systems* (RS) [42] represent an effective response to this problem, by filtering the huge amount of information about the customers in order to get useful elements to produce suggestions to them [7,1,56]. 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 the users. In this paper we address 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, allowing a system to generate effective recommendations.*

The main motivation behind this work is that most of the solutions regarding the *user profiling* in the recommender systems context involve the interpretation of the whole set of items previously evaluated by a user, in order to measure their similarity with those that she did not consider yet, and recommend the most similar ones. Indeed, the recommendation process is usually based on the principle that user 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 [26,28,58]. Therefore, a static approach to user profiling can lead toward wrong results due to various factors, such as a simple change of preferences over time or the temporary use of their own account by other people.

Several works [3,21] have shown that some user ratings might be considered as outliers, due to the fact that the same user may rate the same item with different ratings, at different moments in time. This is a well-known problem, which in the literature is defined as *magic barrier* [10,20,44], a term used to identify the point where, due to the noise in the data, the performance and accuracy of an algorithm cannot be further improved. After the magic barrier has been reached, any improvement in terms of accuracy might mean an overfitting instead of a performance enhancement. Thus, the primary aim of the approach we introduce is to measure the similarity between a single item and the others in the user profile, in order to improve the recommendation process by discarding the items that are highly dissimilar with the others in the user profile. In the literature, the coherence of an item with respect to the user profile is usually measured as the variance in the feature space that defines the items, typically based on the ratings given by the users [10]. 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, in this paper we consider 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 allows us to evaluate the coherence of an item in a more extensive way (i.e., 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 our approach removes them, keeping in the user profiles only those items that are coherent with each other, we can 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, we introduce 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 and the others within the user profile. Moreover, the algorithm takes into account two 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* (as "global" we mean all the items in a user profile). These metrics allow us to remove in a selective way any item that could make the user profiles non-adherent to the real preferences of the users. 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. Through our approach, the evaluation process of the items coherence has been moved from a domain based on rigorous mathematical criteria (i.e., variance of the user ratings in the feature space), to a new semantic domain, which presents a considerable advantage in terms of evaluation flexibility. An important aspect related with the proposed approach is its ability to operate in any domain characterized by a textual description of the items, even in absence of a user rating about them (e.g., implicit user preferences collected from the users' browsing sessions).

In order to validate the capability of our approach to produce accurate user profiles, we are going to include the DCBM algorithm into state-of-the-art recommender systems (i.e., SVD [25] and a User-Based Collaborative Filtering approach [35]) and evaluate the accuracy of the performed recommendations. Since the task of a 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 its capability to remove the incoherent items from the user profiles, increasing the accuracy of the recommendations and reducing the computational complexity of the system, also in the context of the non-semantic approaches as those taken into account (i.e., SVD and User-Based Collaborative Filtering approaches).

The contributions of the paper are summarized as follows:

- we introduce a novel algorithm to remove incoherent items from a user profile, with the aim to improve the recommendation accuracy;
- we integrate our algorithm into state-of-the-art recommender systems, in order to improve their effectiveness and validate our proposal;

– we performed experiments on two real-world datasets and two state-of-the-art recommender systems, which compare the accuracy of a recommender system before and after the use of our approach.

This paper is based on the work presented in [43], which was completely rewritten and extended in the following ways: (*i*) we provide a formal notation and definition of the problem we tackle in this work, (*ii*) we extend the presentation of the background concepts behind this proposal, (*iii*) we provide more details on the proposed approach, (*iv*) we compare our proposal against two state-of-the-art recommender systems and two datasets.

The rest of the paper is organized as follows: Section 2 presents related work on user profiling; in Section 3 we introduce the background on the concepts and the problem handled by our proposal; Section 4 contains the formal definition of our problem, the details of the DCBM algorithm, and its integration into a recommender system; Section 5 presents the experimental framework used to evaluate our approach; Section 6 contains conclusions and future work.

2 Related Work

In order to lead the potential buyers toward a number of well-targeted suggestions, related to the large amount of goods or services available today through the electronic commerce circuits, a *Recommender System* plays a determinant role, since it is able to investigate on the user preferences, suggesting them only potentially interesting items. In order to identify them, a recommender system has to *predict* that an item is worth recommending [42].

The first class of systems that was developed employs the so-called *Collaborative Filtering* approach [21,23,31,47], which is based on the assumption that users have similar preferences on a item, if they already have rated other similar items [54]. *Content-based recommender systems*, instead, suggest to users items whose content is similar to that of the items they previously evaluated [33,38]. The early systems used relatively simple retrieval models, such as the Vector Space Model, with the basic TF-IDF weighting (which is presented in detail in the next section). Examples of systems that employ this type of content filtering are [9,11,29,37]. 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 [13,14].

When 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. There are several approaches to create user profiles: some of them focus on *short-term* user profiles that capture features of the user's current search context [12,18,50], while others accommodate *long-term* profiles that capture the user preferences over a long period of time [8,15,34]. As shown in [58], compared with the *short-term* user profiles, the use of a *long-term* user profile 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 do not reflect the preferences of the user and that represent a form of "noise".

The most common strategies to get useful information to build the user profiles are two, i.e., explicit or implicit. Explicit profiling strategies directly request to the users different forms of preference information, from categorical preferences [15,34] to simple result ratings [8]. Instead, implicit profiling strategies attempt to infer this information by analyzing the users' behavior, and without a direct interaction with them while they perform actions in a website [15,32,40].

However, the strategy usually adopted is the implicit one, where the user preferences are inferred without a direct interaction with her. This is the common approach because the explicit strategy presents some problems, such as those related with the *privacy* aspects (many users do not like to reveal information about their preferences), and those related with the form filling process (many users do not like to spend their time for this activity, and it is proved that the accuracy of the information depends on the time needed to provide them).

The implicit approach usually requires *long-term* user profiles, where the information about the preferences 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 [50].

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. 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 [26], 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 [58] make use of a *tree-descriptor* model to detect shifts in user interests. Another technique exploits the knowledge captured in an ontology [49] 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 the information in the profiles by collecting the implicit feedbacks of the users during their natural interactions with the system (reading-time, saving, etc.) [24]. However, it should be pointed out that most of the strategies used in this area are effective only in specific contexts, such as for instance [60], where a novel approach to automatically model the user profile, according to the change of her preferences, is designed for the articles recommendation context.

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 into account the correlation between the terms [26,58]. This trivial way to manage the information usually does not lead 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 [39]. In [51,52,53], 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.

It should be noted that there is a common issue that afflicts the recommendation approaches, related to the concept of item incoherence. This is a problem that in the literature is identified as *magic barrier* [20], a term used to define the theoretical boundary for the level of optimization that can be achieved by a recommendation algorithm on transactional data [45]. 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 explored in [4,3], 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 aimed at discussing the concept of reliability of users in terms of rating coherence [21].

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

3 Background

In this section we provide some details concerning two key concepts involved in the context of this work, i.e., the spatial representation of a text document based on the *Vector Space Model*, and the functionalities offered by the *WordNet* environment.

3.1 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 $d_j = \{t_1, t_2, ..., t_N\}, t \in T$ 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 $d_j = \{w_{1j}, w_{2j}, ..., w_{nj}\}$ where w_{nj} represents the weight for term t_n 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, a trivial approach that uses empirical observations of the documents' terms [46].

3.2 WordNet Environment

Due to the fact that an approach based on a simple *bag of words* is not able to perform a semantic disambiguation of the words in an item description, also motivated by the fact that exploiting a taxonomy for categorization purposes is an approach recognized in the literature [2], and by the fact that a semantic analysis is useful to improve the accuracy of a classification [5,6], 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 [17].

3.2.1 WordNet Structure

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 by as many distinct synsets, so that each form-meaning pair in WordNet is unique (e.g., the *fly* noun 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 our work, we consider only the *nouns* and the *verbs*, exploiting the state-of-art semantic-based approach to item recommendation based on the WordNet synsets [39], to evaluate the semantic similarity between the items stored in the user profiles.

4 Item Recommendation with Incoherent Items Removal

The definition of the problem handled by our proposal, the notation used in the problem statement, the details about the implementation of the proposed algorithm, and its integration on a recommender system, are described in the following (Sections 4.1 and 4.2).

4.1 Problem Definition

Here, after introducing the adopted notation, we define the problem handled by our proposal.

Definition 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, $\overline{v} = 3$).

Definition 2 (User items) 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 preference, 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.

Definition 3 (Semantic item 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 $\mid d_i \mid = \mid BoW \mid$). 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 $\mid D \mid = \mid I \mid$). The approach used to extract sd_i from d_i is described in detail in Section 4.2.1.

Definition 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 semantic descriptions of the items in I_u. More formally, $M_u = \{s_w \mid s_w \in sd_m \wedge i_m \in I_u, \forall i_m \in I_u\}$.

Definition 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.

Problem 1 Given a set of items I_u that a user likes, our 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.

4.2 Our 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 preferences of the users, because they are exploited to predict their future interests. For this reason, in this section we propose a novel approach defined *Dynamic Coherence-Based Modeling* (DCBM) that allows us to find and remove the incoherent items from user profiles, regardless of the chosen profiling method. The implementation on a recommender system of the DCBM is described in the following subsections.

4.2.1 Data Preprocessing

Before comparing the similarity between the items in a user profile, 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* [55]. In the second step, we remove 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, and in this work we have used a list of 429 stop words made available with the *Onix Text Retrieval Toolkit*[1]. In the last 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 evaluate the semantic similarity in a precise way. The best sense of each word in a sentence (i.e., the selection of the real meaning of a word in the context where it is used) was found through the Java implementation of the adapted Lesk algorithm provided by the *Denmark Technical University* similarity application [46]. All the collected synsets form the set $S = \{s_1, \ldots, s_W\}$ defined in Section 4.1. 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 0 otherwise.

4.2.2 Semantic Similarity

The most used semantic similarity measures are five, and are those defined by *Leacock and Chodorow* [27], *Jiang and Conrath* [22], *Resnik* [41], *Lin* [30], and *Wu and Palmer* [59]. Each of them evaluates the semantic similarity between two WordNet synsets, and we calculate the semantic similarity by using *Wu and Palmer*'s measure, a method based on the path length between a pair of concepts (WordNet synsets), which in the literature is considered to be the most accurate when generating the similarities [16,13].

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 1.

$$Q = (\langle x_1, y_1 \rangle, \langle x_1, y_2 \rangle, \ldots, \langle x_i, y_j \rangle) \, \forall x \in X, y \in Y \tag{1}$$

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 2).

$$Z = \{\langle x_i, y_j \rangle \mid POS(x_i) = POS(y_j)\} \tag{2}$$

The metric measures the similarity between concepts in an ontology (in our case it is WordNet), as shown in Equation 3.

$$sim_{WP}(x, y) = \frac{2 \cdot A}{B + C + (2 \cdot A)} \tag{3}$$

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 3 we have that

[1] http://www.lextek.com/manuals/onix/stopwords.html
[2] http://lyle.smu.edu/ tspell/jaws/index.html

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.

In the example of Fig. 1, v_4 is the parent (and also ancestor) of v_5, while v_2 is an ancestor of both v_5 and v_3. In this case, the LCS of v_3 and v_5 is v_2, since it is the most specific concept that is an ancestor of both v_3 and v_5. Note that while v_1 is a common subsumer of both v_3 and v_5, it is not the least, since there is still a child of v_1 (in this case it is v_2), which is also a common subsumer of both v_5 and v_3. v_4 is not the least common subsumer since it is not an ancestor of v_3.

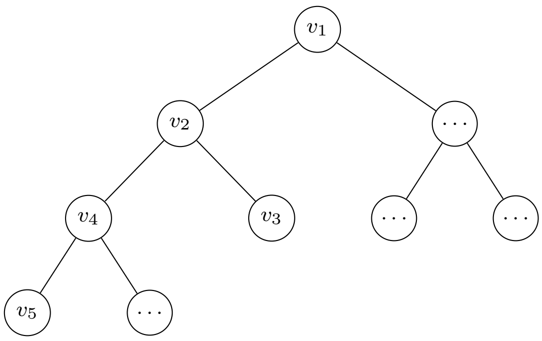

Fig. 1: WordNet Relationships Tree

In order to calculate the Wu and Palmer similarity between v_3 and v_5, we first determine that the least common subsumer of v_3 and v_5 is v_2. Next, we determine that the length of the path from v_3 to v_2 is 1, that the length of the path from v_5 to v_2 is 2, and that the depth of v_2 is 1 (i.e., the distance from v_2 to the root vertex v_1). Now we can determine the similarity between the synsets v_3 and v_5, as shown in Equation 4.

$$sim_{WP}\left(v_3, v_5\right) = \frac{2 \cdot 1}{2 + 1 + (2 \cdot 1)} = 0.40 \tag{4}$$

The similarity between two items is defined as the sum of the similarity score of all the item pairs, divided by its cardinality (the subset Z of WordNet synsets with a common part-of-speech), as shown in Equation 5.

$$sim_{WP}(X, Y) = \frac{\sum\limits_{(x,y)\in Z} sim_{WP}(x, y)}{\mid Z \mid} \tag{5}$$

This similarity metric is employed by our algorithm to compute the coherence of an item with the rest of the semantic user profile.

4.2.3 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 preferences.

In order to identify which items positively evaluated by a user ($i \in I_u$) do not reflect her preference, representing for instance the result of past wrong choices or the use by third parties of her account, the Dynamic Coherence-Based Modeling (DCBM) algorithm measures the *Minimum Global Coherence* (MGC) of each single item description with the set of the other items present in her 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 6 (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).

$$MGC = \underset{i \in I_u}{\operatorname{argmin}} \Big(sim_{WP}(sd_i, M_u \setminus sd_i) \Big) \tag{6}$$

The basic idea is to isolate each individual item i in a user profile, semantically described by sd_i, and then measure its similarity with respect to the other 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 identify 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 1.

Theorem 1 *The Minimum Global Coherence coefficient corresponds to the maximum negative slope.*

Proof Placing 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 7.

$$m = \frac{\triangle y}{\triangle x} = \frac{f(x + \triangle x) - f(x)}{\triangle x} \tag{7}$$

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 $1 - 0 = 1$, $2 - 1 = 1$, ...,

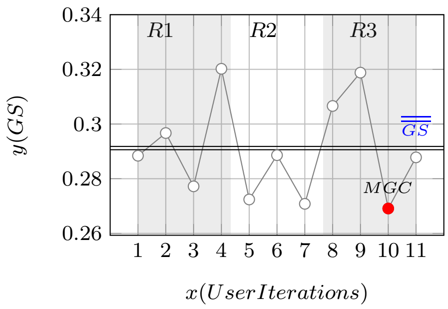

Fig. 2: The maximum negative slope corresponds to the value of MGC

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

Table 1: User profile sample data

$N-(N-1)=1$), the slope value m will always be equal to $f(x+\triangle x)-f(x)$. As Equation 8 shows, where $sim_{WP}(I_u)$ denotes $sim_{WP}(sd_i, M_u \backslash sd_i), \forall i \in I_u$, the maximum negative slope corresponds to the value of MGC.

$$min\left(\frac{\triangle y}{\triangle x}\right) = min\left(\frac{sim_{WP}(I_u)}{1}\right) = MGC \tag{8}$$

In Fig. 2, which displays the data reported in Table 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 preferences of the user or an item not semantically distant enough from the context of the remaining items, the DCBM algorithm considers an item as incoherent and removes it, only if it 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 preferences only when it goes up in the first part of the iterations. This condition is checked thanks 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 Fig. 2, $r = \frac{2}{3} = 0.66$ (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 higher than the *mean value* of the *global similarity.*

Regarding the first requirement, it should be noted that the regions extension is strongly related both to the type of items and to their frequency of fruition, so it depends on the operative scenario. With respect to the second requirement, we prevent the removal of items when their semantic distance with the remaining items is lower than the mean value. 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 9 measures the mean similarity, denoted by $\overline{GS}$, by calculating the average of the *Global Similarity* (GS) values, which are obtained as $sim_{WP}(sd_i, M_u \setminus sd_i), \forall i \in I_u$.

$$\overline{GS} = \frac{1}{\mid I_u \mid} \cdot \sum_{i \in I_u} (sim_{WP}(sd_i, M_u \setminus sd_i)) \tag{9}$$

where $\mid I_u \mid$ represents the total number of items stored in the profile (in the case of sample data shown in Table 1, the $\overline{GS} = 0.2906$). Once this average value is obtained, we can proceed to define the condition ρ to be used to decide when an item has to be removed (1) or not (0), based on the average value $\overline{GS}$ (as shown in Equation 10).

$$\rho = \begin{cases} 1, & \text{if } MGC < \overline{GS} \\ 0, & \text{otherwise} \end{cases} \tag{10}$$

Based on the above considerations, we can now define 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), and a removal area r used to define in which part of the profile an item can be removed. In step 3 we extract the set of synsets M_u (Definition 4) from the description of the items in the user profile I_u (Definition 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 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 without the removed items.

4.2.4 Item Recommendation

After the user profile has been processed by Algorithm 1, this step computes the similarity with all the items not evaluated, and recommends to a user a subset of those with the highest similarity. An interesting aspect to consider is that, thanks to our proposal, a user profile in which both the preferences

Algorithm 1 DCBM Algorithm

Require: I_u=set of items in the user profile, 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 $i < r * size(I_u)$ AND $MGC < a$ **then**
            Remove(i)
            Remove(s)
        end if
    end for
    Return I_u
end procedure

in terms of ratings and in terms of synsets would be available to a recommender system. This would make possible the generation of recommendations with both collaborative filtering and content-based approaches. In this study we consider the two main subclasses of collaborative filtering approaches (i.e., the neighborhood approach and the latent factor models), since collaborative filtering approaches are known to be the most accurate. In particular, we consider SVD [25] for the latent factor model-based approaches and a user-based approach [35] for the neighborhood subclass. This will allow us to evaluate our approach both in a scenario in which the feature space is reduced (SVD), and in which the recommendations are instead produced by considering all the items a user evaluated (user-based approach).

4.2.5 Summary

We can summarize the implementation process of the DCBM algorithm on a recommender system in the following four steps:

1. **Data Preprocessing**: preprocessing of the textual description of all items, in order to remove the useless elements and the items with a 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 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 preferences are removed from the user profile;

4. **Item Recommendation**: the user profile without the incoherent items is employed to filter the items a user has not yet evaluated, in order to select the items to recommend.

5 Experimental Framework

This section presents the framework used to evaluate our proposal. The strategy that drove our experiments is first described; then the dataset used and the preprocessing made on the data are introduced; after, the metrics for the evaluation are presented; the last part of the section presents the obtained results.

5.1 Experimental Strategy

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, we used two real-world datasets widely-employed in the recommender systems literature, i.e., Yahoo! Webscope (R4) and Movielens 10M.

The recommender systems used to test the effectiveness of the DCBM algorithms are SVD and a classic User-Based Nearest Neighbors Collaborative Filtering approach. As previously highlighted, this will allow us to evaluate our proposal on recommender systems in which the dimensionality is reduced and in which the feature space is processed as it is.

The Mahout framework was used to implement these recommender systems. In addition to the training set, the framework requires two parameters for SVD: the number of target features and the number of training steps to run (parameter λ). The value of the first parameter can be chosen arbitrarily, on the basis of the number of features that the SVD should target; in our experiments the performance differences measured between our approach and SVD are almost the same by varying this value[3]; we have set it to 19 for the Yahoo dataset and to 18 for the Movielens dataset (i.e., as the number of genres of each dataset); the second parameter, λ, will instead be tested through a set of experiments.

For the user-based approach, it is necessary to set the number of neighbors to consider when predicting the ratings; also this parameter will be set thanks to a specific set of experiments. The distance function chosen for the neighborhood approach is the Pearson correlation, one of the most common measure of correlation to evaluate the similarity between two users [19].

In order to validate our proposal, a comparative analysis has been performed, by considering the values obtained by a recommender system both in a scenario in which the user profile is processed with the DCBM algorithm

[3] The analysis has been omitted, since it did not show significant and interesting results, and in order to facilitate the reading of the paper.

and in a scenario in which the profile is processed as it is. The comparisons have been made by measuring both the Root Mean Squared Error (RMSE) and the Average Difference (AD).

RMSE was chosen as a metric to compare the algorithms because, as the organizers of the Netflix prize highlight[4], it is well-known and widely used, it allows to evaluate a system through a single number, and it emphasizes the presence of large errors (both false positives and false negatives). AD was measured since it is being recently considered by largely employed frameworks (such as Apache's Mahout) to facilitate the interpretability of the results [36].

Three sets of experiments have been performed:

1. **Analysis of the removed items.** We analyze the amount of items removed for each user, in order to analyze the impact of the DCBM algorithm on a user profile.
2. **Recommendation accuracy measurement.** We validate the capability of our approach to remove incoherent items, by analyzing the accuracy of the generated recommendations with and without the incoherent items in the user profiles.
3. **Performance analysis.** We consider the amount of time it takes to process an item, when analyzing if it should be removed or not. This allows us to evaluate our algorithm considering its performance, and to analyze, once an item gets removed, how much time can be saved by using our proposal.

5.2 Datasets

In order to evaluate our strategy, we perform a series of experiments on two different real-world datasets, extracted by two standard benchmarks for recommender systems: Yahoo! Webscope R4[5] and Movielens 10M[6].

Yahoo! Webscope (R4). This dataset contains a large amount of data related to user preferences expressed on the Yahoo! Movies community that are rated on the base of two different scales, from 1 to 13 and from 1 to 5 (we use the latter). The training data is composed by $7,642$ users ($| U |$), $11,915$ movies/items ($| I |$), and $211,231$ ratings ($| P |$). All the users in the training set have rated at least 10 items and all items are rated by at least one user. 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 19 different classes (genres), and it should be noted that an item may be classified with multiple classes. The information of this dataset, training and test data, are sorted in chronological order, and it should be also noted that the test data

[4] http://www.netflixprize.com/faq
[5] http://webscope.sandbox.yahoo.com
[6] http://grouplens.org/datasets/movielens/

were gathered chronologically after the training data.

Movielens 10M. The second dataset used in this work is composed by $71,567$ users ($\mid U \mid$), $10,681$ movies/items ($\mid I \mid$), and $10,000,054$ ratings ($\mid P \mid$). It was extracted at random from 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. 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. In this dataset the users were selected at random for the inclusion in the training and test data.

5.3 Metrics

The accuracy of the predicted ratings was measured through the Root Mean Squared Error (RMSE) and the Average Difference (AD). Both metrics consider the test set and the predicted ratings, by comparing each rating r_{ui}, given by a user u for an item i and available in the test set, with the rating p_{ui} predicted by a recommender system. The formulas are shown below, where n is the number of ratings available in the test set:

$$RMSE = \sqrt{\frac{\sum_{i=0}^{n}(r_{ui} - p_{ui})^2}{n}}$$

$$AD = \frac{\sum_{i=0}^{n} \mid r_{ui} - p_{ui} \mid}{n}$$

5.4 Experimental Results

This section presents the results for each set of experiments presented in the strategy.

5.4.1 Analysis of the Removed Items

Fig. 3 reports the number of removed items and the amount of users for which that amount of items were removed, in the Webscope dataset (Fig. 3a) and in the Movielens dataset (Fig. 3b). Both figures show a power law distribution, which indicates that for the vast majority of the users just a small number of

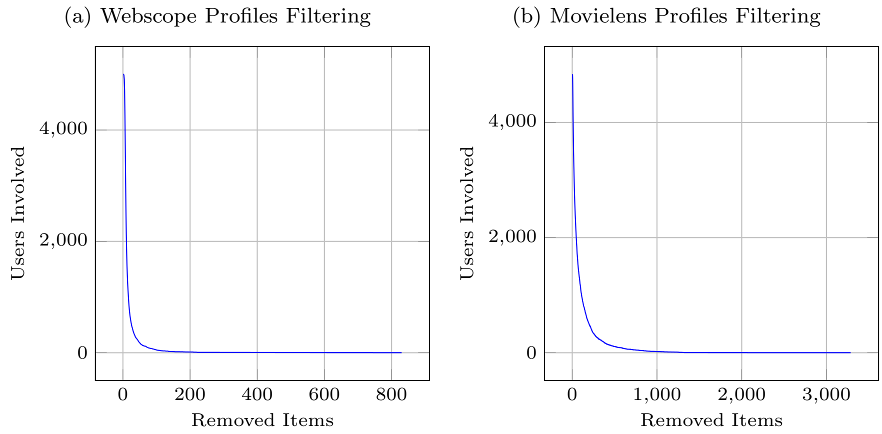

Fig. 3: Removed Items

items gets removed. However, there are cases in which hundreds of items are removed from a single user profile and get classified by our approach as noise.

In the next set of experiments we evaluate the capability of our approach to generate accurate recommendations by employing these profiles without incoherent items.

5.4.2 Recommendation Accuracy Measurement

Fig. 4 shows the results obtained for the SVD recommender system and the two employed metrics, in both the Webscope (Fig. 4a-b) and Movielens (Fig. 4c-d) datasets. The results of the Webscope dataset show that the best results can be obtained with $\lambda = 0.141$, $RMSE = 1.0124$, $AD = 0.7788$ for the unfiltered dataset and $\lambda = 0.171$, $RMSE = 1.0128$, $AD = 0.7301$ for the data filtered with the DCBM algorithm. Regarding the results with the Movielens data, the highest accuracy is obtained with $\lambda = 0.141$, $RMSE = 0.96$, $AD = 0.7788$ for the unfiltered dataset and $\lambda = 0.091$, $RMSE = 0.9751$, $AD = 0.755$ for the data filtered with the DCBM algorithm. We also performed independent-samples two-tailed Student's t-tests, which highlighted that there is no statistical difference between the results ($p > 0.05$). Indeed, we can notice that the RMSE of the filtered data is slightly higher, while the AD is slightly lower. This means that, when the feature space is reduced by the recommendation approach, we can reduce the complexity of the system by processing less items while keeping the same accuracy.

In Fig. 5 we analyzed the other scenario, in which the feature space processed by the recommender system is the entire user profile, by considering the nearest-neighbors user-based filtering. Fig. 5 shows the results with the employed metrics, in both the Webscope (Fig. 5a-b) and Movielens (Fig. 5c-d) datasets. The results of the Webscope dataset show that the best results can be obtained with $neighbors = 8$, $RMSE = 1.058$, $AD = 0.7408$ for the unfil-

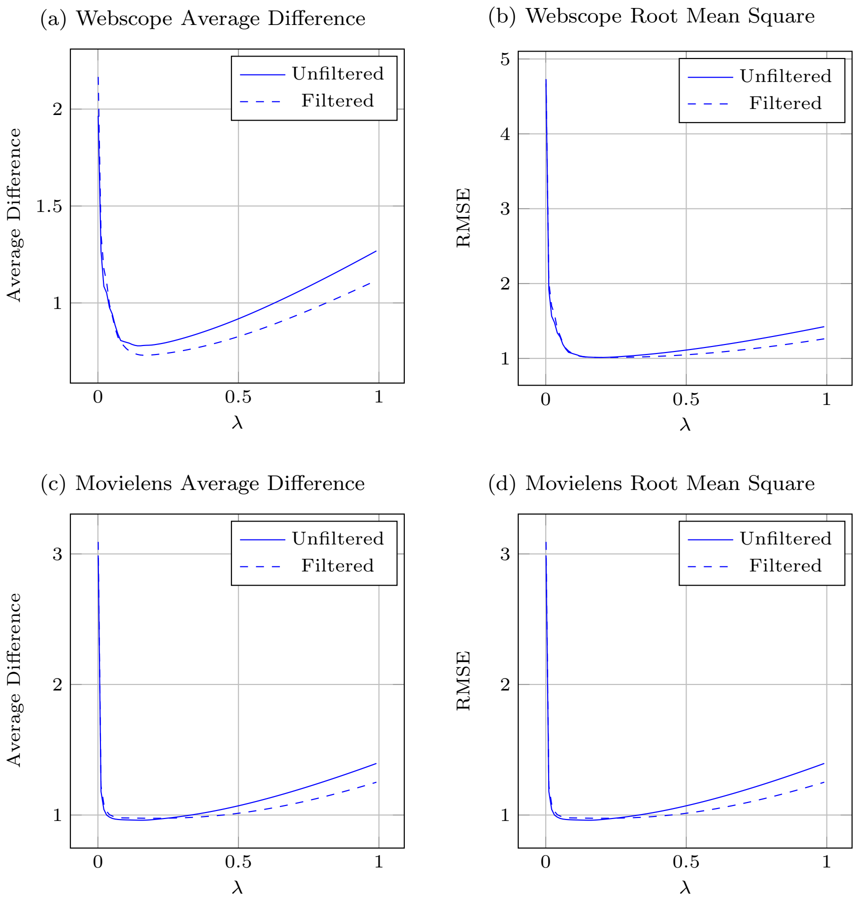

Fig. 4: SVD recommendation accuracy

tered dataset and $neighbors = 7$, $RMSE = 0.8785$, $AD = 0.5877$ for the data filtered with the DCBM algorithm. Regarding the results with the Movielens data, the highest accuracy is obtained with $neighbors = 98$, $RMSE = 1.0485$, $AD = 0.8255$ for the unfiltered dataset and $neighbors = 3$, $RMSE = 1.0035$, $AD = 0.7421$ for the data filtered with the DCBM algorithm. These results show that when the entire feature space is considered (as the collaborative filtering neighborhood-based approach tested in this set of experiments, or the content-based systems), the recommendation accuracy can be significantly improved ($p < 0.05$) by removing the incoherent items from the user profiles.

5.4.3 Performance Analysis

Fig. 6 shows the average amount of time in seconds it takes for the DCBM algorithm to process an item and to decide if it should be removed or not, by

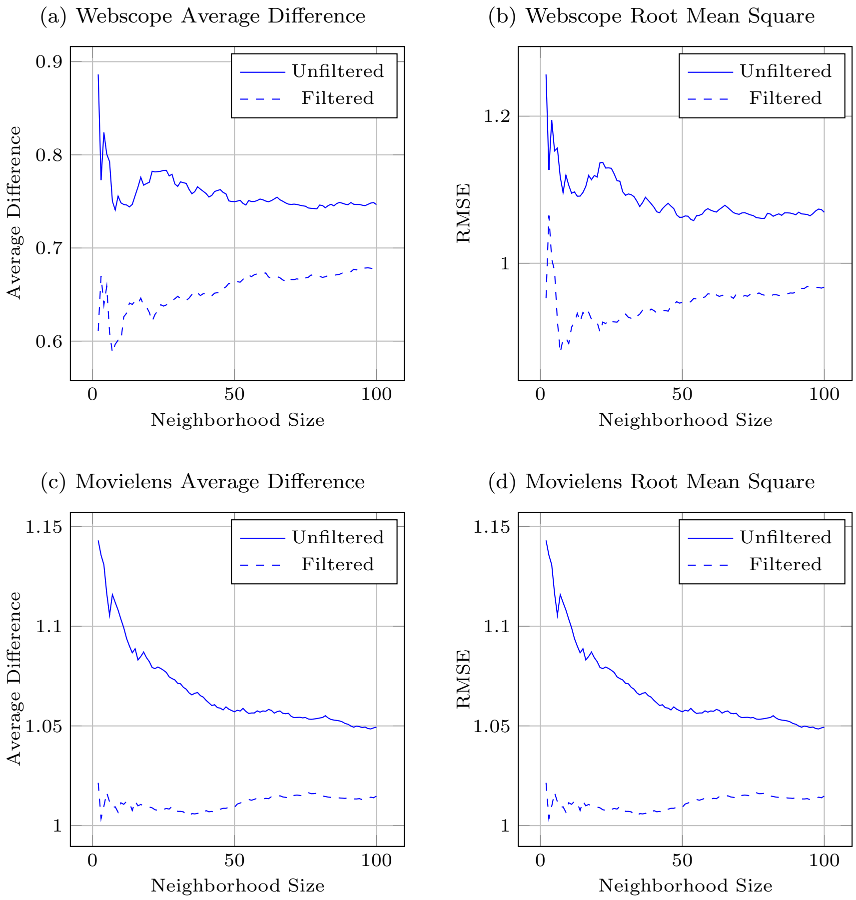

Fig. 5: User-based Approach Recommendation Accuracy

considering the first 1000 user profiles, in order to achieve a statistic relevance. The dashed line in each subfigure indicates the average time considering all the user profiles. In the Webscope dataset (Fig. 6a) the average time to process an item is 0.21 seconds, while for the Movielens dataset (Fig. 6b) it is 0.36 seconds. This means that the algorithm presents a very good performance considering the semantic process behind the DCBM algorithm. Moreover, this quick process to decide if an item should be removed or not leads to great improvements when the recommendations are processed, both in terms of number of items to process and in terms of accuracy when the whole feature space is considered.

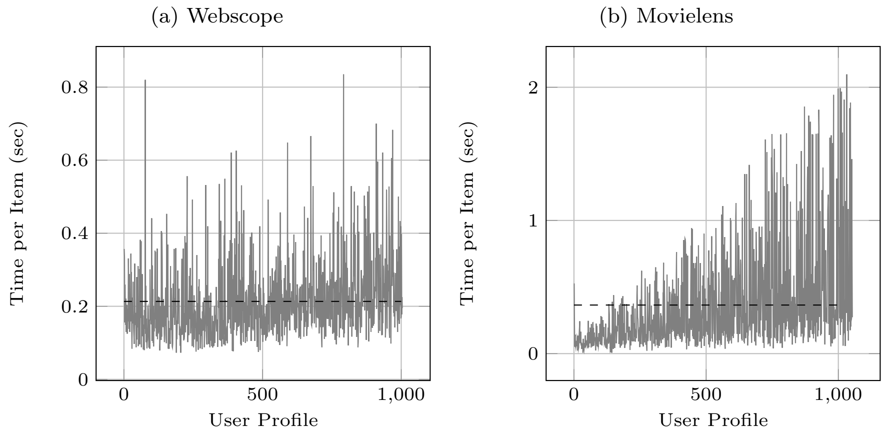

Fig. 6: Computation Time

5.4.4 Discussion

We verified that the proposed approach is actually able to remove the incoherent items from the user profiles, as highlighted in Fig. 3, which shows that there are many cases in which tens of items are removed from a single user profile. Considering that the removed items are not useful to model the real preferences of the users (i.e., they have been classified as incoherent by our approach), we obtain a considerable reduction of the computational load, without any side effect.

The main considerations that arise from the observation of the experimental results are the following: when we adopt a recommendation approach at the state of the art that does not modify the feature space, the accuracy improves significantly, as it happens with the user-based approach, whose results are shown in Fig. 5; otherwise, when we use an approach that reduces this space, as happens by using SVD, the accuracy remains unaltered, as we can observe in Fig. 4.

In this last case, the advantages that come from the use of our approach are to be found in the reduced number of items taken into account in the recommendation process. In fact, considering that the removal of the items from the user profiles does not require much time, as shown in Fig. 6, we have a considerable reduction of the time needed to perform the recommendation process, because it operates by analyzing a number of items that is fewer than the initial one, as reported in Fig. 3.

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. About this aspect, it should be observed that the computational load related to the comparisons performed to detect the MGC do not involve the recommendation process, since it happens in another moment (i.e., when a user evaluates a new item). Even if our approach has a high computational complexity, it is meant to run in background. Indeed, when a new item is evaluated by the user, it would not be removed (even if incoherent) since it is recent, so its similarity with the rest of the user profile can be processed in background. Therefore, the recommendation task, which is the one that has to run fast because it operates online, can benefit from the proposed DCBM algorithm and from the removal of the incoherent items.

In conclusion, our approach improves the performance of a recommender system, both in terms of accuracy and of reduction of the computational time. As shown in the results of the experiments, type, and entity of the improvements is related with the recommendation strategy.

6 Conclusions and Future Work

In this paper we proposed a novel approach able to improve the quality of the user profiling, which takes into account the items related to a user, with the aim of removing those that do not reflect her real preferences. This is useful in many contexts, such as when the system does not allow the users to express their preferences, or when the users do not to make use of this option.

Through it we achieved a twofold result: firstly, we 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 our approach is able to reshape the user profiles in a coherent way.

A further possible extension might involve the use of a large amounts of data also related to contexts different from each other as, for example, a sales platform that gives 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.

References

1. Addis, A., Armano, G., Giuliani, A., Vargiu, E.: 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, pp. 859–861. IEEE (2010)
2. Addis, A., Armano, G., Vargiu, E.: Assessing progressive filtering to perform hierarchical text categorization in presence of input imbalance. In: A.L.N. Fred, J. Filipe (eds.) KDIR 2010 - Proceedings of the International Conference on Knowledge Discovery and Information Retrieval, Valencia, Spain, October 25-28, 2010, pp. 14–23. SciTePress (2010)

3. Amatriain, X., Pujol, J.M., Oliver, N.: I like it... I like it not: Evaluating user ratings noise in recommender systems. In: G. Houben, G.I. McCalla, F. Pianesi, M. Zancanaro (eds.) User Modeling, Adaptation, and Personalization, 17th International Conference, UMAP 2009, formerly UM and AH, Trento, Italy, June 22-26, 2009. Proceedings, *Lecture Notes in Computer Science*, vol. 5535, pp. 247–258. Springer (2009)
4. Amatriain, X., Pujol, J.M., Tintarev, N., Oliver, N.: Rate it again: increasing recommendation accuracy by user re-rating. In: L.D. Bergman, A. Tuzhilin, R.D. Burke, A. Felfernig, L. Schmidt-Thieme (eds.) Proceedings of the 2009 ACM Conference on Recommender Systems, RecSys 2009, New York, NY, USA, October 23-25, 2009, pp. 173–180. ACM (2009)
5. Armano, G., Giuliani, A., Vargiu, E.: Semantic enrichment of contextual advertising by using concepts. In: J. Filipe, A.L.N. Fred (eds.) KDIR 2011 - Proceedings of the International Conference on Knowledge Discovery and Information Retrieval, Paris, France, 26-29 October, 2011, pp. 232–237. SciTePress (2011)
6. Armano, G., Giuliani, A., Vargiu, E.: Studying the impact of text summarization on contextual advertising. In: F. Morvan, A.M. Tjoa, R. Wagner (eds.) 2011 Database and Expert Systems Applications, DEXA, International Workshops, Toulouse, France, August 29 - Sept. 2, 2011, pp. 172–176. IEEE Computer Society (2011)
7. Armano, G., Vargiu, E.: A unifying view of contextual advertising and recommender systems. In: A.L.N. Fred, J. Filipe (eds.) KDIR 2010 - Proceedings of the International Conference on Knowledge Discovery and Information Retrieval, Valencia, Spain, October 25-28, 2010, pp. 463–466. SciTePress (2010)
8. Asnicar, F.A., Tasso, C.: 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, pp. 3–11 (1997)
9. Balabanovic, M., Shoham, Y.: Content-based, collaborative recommendation. Commun. ACM **40**(3), 66–72 (1997). DOI 10.1145/245108.245124. URL `http://doi.acm.org/10.1145/245108.245124`
10. Bellogín, A., Said, A., de Vries, A.P.: The magic barrier of recommender systems - no magic, just ratings. In: V. Dimitrova, T. Kuflik, D. Chin, F. Ricci, P. Dolog, G. Houben (eds.) User Modeling, Adaptation, and Personalization - 22nd International Conference, UMAP 2014, Aalborg, Denmark, July 7-11, 2014. Proceedings, *Lecture Notes in Computer Science*, vol. 8538, pp. 25–36. Springer (2014)
11. Billsus, D., Pazzani, M.J.: A hybrid user model for news story classification. In: Proceedings of the Seventh International Conference on User Modeling, UM '99, pp. 99–108. Springer-Verlag New York, Inc., Secaucus, NJ, USA (1999). URL `http://dl.acm.org/citation.cfm?id=317328.317338`
12. Budzik, J., Hammond, K.J.: User interactions with everyday applications as context for just-in-time information access. In: Proceedings of the 5th International Conference on Intelligent User Interfaces, IUI '00, pp. 44–51. ACM, New York, NY, USA (2000)
13. Capelle, M., Frasincar, F., Moerland, M., Hogenboom, F.: Semantics-based news recommendation. In: Proceedings of the 2Nd International Conference on Web Intelligence, Mining and Semantics, WIMS '12, pp. 27:1–27:9. ACM, New York, NY, USA (2012)
14. Capelle, M., Hogenboom, F., Hogenboom, A., Frasincar, F.: Semantic news recommendation using wordnet and bing similarities. In: Proceedings of the 28th Annual ACM Symposium on Applied Computing, SAC '13, pp. 296–302. ACM, New York, NY, USA (2013)
15. Chirita, P.A., Nejdl, W., Paiu, R., Kohlschütter, C.: Using odp metadata to personalize search. In: Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '05, pp. 178–185. ACM, New York, NY, USA (2005)
16. Dennai, A., Benslimane, S.M.: 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), pp. 197–207. The Society of Digital Information and Wireless Communication (2013)
17. Fellbaum, C.: WordNet: An Electronic Lexical Database. Bradford Books (1998)

18. Finkelstein, L., Gabrilovich, E., Matias, Y., Rivlin, E., Solan, Z., Wolfman, G., Ruppin, E.: Placing search in context: The concept revisited. ACM Trans. Inf. Syst. **20**(1), 116–131 (2002)
19. Herlocker, J.L., Konstan, J.A., Borchers, A., Riedl, J.: An algorithmic framework for performing collaborative filtering. In: SIGIR, pp. 230–237. ACM (1999)
20. Herlocker, J.L., Konstan, J.A., Terveen, L.G., Riedl, J.: Evaluating collaborative filtering recommender systems. ACM Trans. Inf. Syst. **22**(1), 5–53 (2004)
21. Hill, W.C., Stead, L., Rosenstein, M., Furnas, G.W.: Recommending and evaluating choices in a virtual community of use. In: I.R. Katz, R.L. Mack, L. Marks, M.B. Rosson, J. Nielsen (eds.) Human Factors in Computing Systems, CHI '95 Conference Proceedings, Denver, Colorado, USA, May 7-11, 1995., pp. 194–201. ACM/Addison-Wesley (1995)
22. Jiang, J.J., Conrath, D.W.: Semantic similarity based on corpus statistics and lexical taxonomy. arXiv preprint cmp-lg/9709008 (1997)
23. Karypis, G.: Evaluation of item-based top-n recommendation algorithms. In: Proceedings of the 2001 ACM CIKM International Conference on Information and Knowledge Management, Atlanta, Georgia, USA, November 5-10, 2001, pp. 247–254. ACM (2001). DOI 10.1145/502585.502627. URL `http://doi.acm.org/10.1145/502585.502627`
24. Kelly, D., Teevan, J.: Implicit feedback for inferring user preference: a bibliography. SIGIR Forum **37**(2), 18–28 (2003)
25. Koren, Y., Bell, R.M.: Advances in collaborative filtering. In: F. Ricci, L. Rokach, B. Shapira (eds.) Recommender Systems Handbook, pp. 77–118. Springer (2015). DOI 10.1007/978-1-4899-7637-6_3. URL `http://dx.doi.org/10.1007/978-1-4899-7637-6_3`
26. Lam, W., Mukhopadhyay, S., Mostafa, J., Palakal, M.J.: Detection of shifts in user interests for personalized information filtering. In: SIGIR, pp. 317–325 (1996)
27. Leacock, C., Chodorow, M.: Combining local context and wordnet similarity for word sense identification. In: C. Fellbaum (ed.) WordNet: An Electronic Lexical Database, pp. 305–332. MIT Press (1998)
28. Li, L., Yang, Z., Wang, B., Kitsuregawa, M.: Dynamic adaptation strategies for long-term and short-term user profile to personalize search. In: G. Dong, X. Lin, W. Wang, Y. Yang, J.X. Yu (eds.) Advances in Data and Web Management, Joint 9th Asia-Pacific Web Conference, APWeb 2007, and 8th International Conference, on Web-Age Information Management, WAIM 2007, Huang Shan, China, June 16-18, 2007, Proceedings, *Lecture Notes in Computer Science*, vol. 4505, pp. 228–240. Springer (2007)
29. Lieberman, H.: Letizia: An agent that assists web browsing. In: Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 1, IJCAI'95, pp. 924–929. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1995). URL `http://dl.acm.org/citation.cfm?id=1625855.1625975`
30. Lin, D.: An information-theoretic definition of similarity. In: J.W. Shavlik (ed.) Proceedings of the Fifteenth International Conference on Machine Learning (ICML 1998), Madison, Wisconsin, USA, July 24-27, 1998, pp. 296–304. Morgan Kaufmann (1998)
31. Linden, G., Smith, B., York, J.: Industry report: Amazon.com recommendations: Item-to-item collaborative filtering. IEEE Distributed Systems Online **4**(1) (2003). URL `http://dsonline.computer.org/0301/d/wp1lind.htm`
32. Liu, F., Yu, C., Meng, W.: Personalized web search by mapping user queries to categories. In: Proceedings of the Eleventh International Conference on Information and Knowledge Management, CIKM '02, pp. 558–565. ACM, New York, NY, USA (2002)
33. Lops, P., de Gemmis, M., Semeraro, G.: Content-based recommender systems: State of the art and trends. In: F. Ricci, L. Rokach, B. Shapira, P.B. Kantor (eds.) Recommender Systems Handbook, pp. 73–105. Springer (2011)
34. Ma, Z., Pant, G., Sheng, O.R.L.: Interest-based personalized search. ACM Trans. Inf. Syst. **25**(1) (2007)
35. Ning, X., Desrosiers, C., Karypis, G.: A comprehensive survey of neighborhood-based recommendation methods. In: F. Ricci, L. Rokach, B. Shapira (eds.) Recommender Systems Handbook, pp. 37–76. Springer (2015). DOI 10.1007/978-1-4899-7637-6_2. URL `http://dx.doi.org/10.1007/978-1-4899-7637-6_2`
36. Owen, S., Anil, R., Dunning, T., Friedman, E.: Mahout in Action. Manning Publications Co., Greenwich, CT, USA (2011)

37. Pazzani, M., Muramatsu, J., Billsus, D.: Syskill & webert: Identifying interesting web sites. In: Proceedings of the Thirteenth National Conference on Artificial Intelligence - Volume 1, AAAI'96, pp. 54–61. AAAI Press (1996). URL `http://dl.acm.org/citation.cfm?id=1892875.1892883`
38. Pazzani, M.J., Billsus, D.: Content-based recommendation systems. In: P. Brusilovsky, A. Kobsa, W. Nejdl (eds.) The Adaptive Web, pp. 325–341. Springer-Verlag, Berlin, Heidelberg (2007). URL `http://dl.acm.org/citation.cfm?id=1768197.1768209`
39. Pedersen, T., Patwardhan, S., Michelizzi, J.: Wordnet::similarity: Measuring the relatedness of concepts. In: Demonstration Papers at HLT-NAACL 2004, HLT-NAACL–Demonstrations '04, pp. 38–41. Association for Computational Linguistics, Stroudsburg, PA, USA (2004)
40. Pretschner, A., Gauch, S.: Ontology based personalized search. In: 11th IEEE International Conference on Tools with Artificial Intelligence, ICTAI '99, Chicago, Illinois, USA, November 8-10, 1999, pp. 391–398. IEEE Computer Society (1999). DOI 10.1109/TAI.1999.809829. URL `http://dx.doi.org/10.1109/TAI.1999.809829`
41. Resnik, P.: Using information content to evaluate semantic similarity in a taxonomy. In: Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 1, IJCAI'95, pp. 448–453. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1995)
42. Ricci, F., Rokach, L., Shapira, B.: Introduction to recommender systems handbook. In: F. Ricci, L. Rokach, B. Shapira, P.B. Kantor (eds.) Recommender Systems Handbook, pp. 1–35. Springer (2011)
43. Saia, R., Boratto, L., Carta, S.: 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, pp. 154–161. SciTePress (2014)
44. Said, A., Jain, B.J., Narr, S., Plumbaum, T.: Users and noise: The magic barrier of recommender systems. In: J. Masthoff, B. Mobasher, M.C. Desmarais, R. Nkambou (eds.) User Modeling, Adaptation, and Personalization - 20th International Conference, UMAP 2012, Montreal, Canada, July 16-20, 2012. Proceedings, *Lecture Notes in Computer Science*, vol. 7379, pp. 237–248. Springer (2012)
45. Said, A., Jain, B.J., Narr, S., Plumbaum, T., Albayrak, S., Scheel, C.: Estimating the magic barrier of recommender systems: a user study. In: W.R. Hersh, J. Callan, Y. Maarek, M. Sanderson (eds.) The 35th International ACM SIGIR conference on research and development in Information Retrieval, SIGIR '12, Portland, OR, USA, August 12-16, 2012, pp. 1061–1062. ACM (2012)
46. Salton, G., Wong, A., Yang, C.S.: A vector space model for automatic indexing. Commun. ACM **18**(11), 613–620 (1975)
47. Sarwar, B.M., Karypis, G., Konstan, J.A., Riedl, J.: Item-based collaborative filtering recommendation algorithms. In: V.Y. Shen, N. Saito, M.R. Lyu, M.E. Zurko (eds.) Proceedings of the Tenth International World Wide Web Conference, WWW 10, Hong Kong, China, May 1-5, 2001, pp. 285–295. ACM (2001). DOI 10.1145/371920.372071. URL `http://doi.acm.org/10.1145/371920.372071`
48. Schafer, J.B., Konstan, J.A., Riedl, J.: Recommender systems in e-commerce. In: Proceedings of the 1st ACM conference on Electronic commerce, pp. 158–166 (1999)
49. Schickel-Zuber, V., Faltings, B.: 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, pp. 1413–1418. AAAI Press (2006)
50. Shen, X., Tan, B., Zhai, C.: Implicit user modeling for personalized search. In: O. Herzog, H.J. Schek, N. Fuhr, A. Chowdhury, W. Teiken (eds.) Proceedings of the 2005 ACM CIKM International Conference on Information and Knowledge Management, Bremen, Germany, October 31 - November 5, 2005, pp. 824–831. ACM (2005)
51. Stilo, G., Velardi, P.: Temporal semantics: Time-varying hashtag sense clustering. In: Knowledge Engineering and Knowledge Management, *Lecture Notes in Computer Science*, vol. 8876, pp. 563–578. Springer International Publishing (2014)
52. Stilo, G., Velardi, P.: 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, WI-IAT '14, pp. 186–193. IEEE Computer Society, Washington, DC, USA (2014). DOI 10.1109/WI-IAT.2014.97
53. Stilo, G., Velardi, P.: Efficient temporal mining of micro-blog texts and its application to event discovery. Data Mining and Knowledge Discovery (2015). DOI 10.1007/s10618-015-0412-3
54. Su, X., Khoshgoftaar, T.M.: A survey of collaborative filtering techniques. Adv. Artificial Intellegence **2009** (2009). DOI 10.1155/2009/421425. URL `http://dx.doi.org/10.1155/2009/421425`
55. Toutanova, K., Klein, D., Manning, C.D., Singer, Y.: 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, NAACL '03, pp. 173–180. Association for Computational Linguistics, Stroudsburg, PA, USA (2003)
56. Vargiu, E., Giuliani, A., Armano, G.: Improving contextual advertising by adopting collaborative filtering. ACM Trans. Web **7**(3), 13:1–13:22 (2013)
57. Wei, C., Khoury, R., Fong, S.: Recommendation systems for web 2.0 marketing. In: K. Yada (ed.) Data Mining for Service, *Studies in Big Data*, vol. 3, pp. 171–196. Springer Berlin Heidelberg (2014)
58. Widyantoro, D.H., Ioerger, T.R., Yen, J.: Learning user interest dynamics with a three-descriptor representation. JASIST **52**(3), 212–225 (2001)
59. Wu, Z., Palmer, M.: Verbs semantics and lexical selection. In: Proceedings of the 32Nd Annual Meeting on Association for Computational Linguistics, ACL '94, pp. 133–138. Association for Computational Linguistics, Stroudsburg, PA, USA (1994)
60. Zeb, M., Fasli, M.: Adaptive user profiling for deviating user interests. In: Computer Science and Electronic Engineering Conference (CEEC), 2011 3rd, pp. 65–70 (2011). DOI 10.1109/CEEC.2011.5995827