Future Generation Computer Systems 64 (2016) 186–197

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

Binary sieves: Toward a semantic approach to user segmentation for behavioral targeting

Roberto Saia *, Ludovico Boratto, Salvatore Carta, Gianni Fenu

Dipartimento di Matematica e Informatica, Università di Cagliari, Via Ospedale 72, 09124 Cagliari, Italy

HIGHLIGHTS

- We propose a novel segmentation approach for user targeting, based on a semantic analysis of the items evaluated by a user.
- Through the semantic analysis we extend the ground truth, to generate non trivial segments.
- With respect to classic segmentation, the advertiser can introduce constraints and atomically model the user segments.

ARTICLE INFO

Article history:
Received 20 March 2015
Received in revised form
24 February 2016
Accepted 9 April 2016
Available online 19 April 2016

Keywords:
User segmentation
Semantic analysis
Behavioral targeting

ABSTRACT

Behavioral targeting is the process of addressing ads to a specific set of users. 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 paper, 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. Experimental results confirm the effectiveness of our approach at producing users segments.

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. 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 [1]. Tu and Lu [2] proposed a user segmentation approach based on a semantic analysis of the queries issued by the users, while Gong et al. [1] proposed a LDA-based semantic segmentation that groups users with similar query and click behaviors.

When dealing with a semantic behavioral targeting approach, several problems remain open.

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 [3–5]. 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 filter them can increase the accuracy of a system [6–8]. 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.

Preference stability. To complicate the previous scenario, there are domains like movies in which the preferences tend to be stable over time [9] (i.e., users tend to watch movies of the same genres

* Corresponding author.
E-mail addresses: roberto.saia@unica.it (R. Saia), ludovico.boratto@unica.it (L. Boratto), salvatore@unica.it (S. Carta), fenu@unica.it (G. Fenu).

http://dx.doi.org/10.1016/j.future.2016.04.006
0167-739X/

or by the same director/actor). This is useful to maintain high-quality knowledge sources, but considering only the items a user evaluated leads to trivial sets of users that represent the target (this problem is known as *overspecialization* [10]).

Interpretability of the segments. The last open problem that has to be faced in this research area is the interpretability of a segment. Indeed, a recent survey on user segmentation (mostly focused on the library domain) [11], highlighted that, in order to create a proper segmentation of the users, it is important to *understand* them. On the one hand, easily interpretable approaches generate trivial segments, and even a partitioning with the k-means clustering algorithm has proven to be more effective than this method [12], while on the other hand, when a larger set of features is combined, the problem of properly understanding and interpreting results arises [13,14]. This is mostly due to the lack of guidance on how to interpret the results of a segmentation [15]. The fact 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.

Our contributions. In this paper, we have moved the item analysis process from the canonical deterministic space model (i.e., that based on strict mathematical criteria) to a more flexible semantic space model that allows us to extend the analysis capability, which in the literature has been highlighted as a challenging topic [16,17]. In particular, we tackle the problem of *defining a semantic user segmentation approach, such that the sources of information used to build it are reliable, the generated segmentation is not trivial and it is easily interpretable*.

The proposed approach 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 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 model in a compact way the information related to these big amounts of data.

The scientific contributions of our proposal are now recapped:

- we introduce a novel data structure, called *semantic binary sieve*, to semantically characterize each class of items;
- we present a semantic user segmentation approach based on reliable sources of 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 solve the overspecialization issues caused by preference stability by building a model for each user that considers her/him 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 us to specify, in a simple but punctual way, the interests that the segment should cover; this algebra, along with the built models, avoids 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 boolean operations. 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 users.

The rest of the paper is organized as follows: we first present the works in the literature related with our approach (Section 2), then we provide a background on the concepts handled by our proposal and the formal definition of the tackled problem (Section 3), we continue with the implementation details (Section 4) and the description of the performed experiments (Section 5), ending with some concluding remarks (Section 6).

2. Related work

In this section we are going to explore the main works in the literature related to the open problems highlighted in the Introduction.

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 of interaction with the

[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.

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. [18], show that an accurate monitoring of the click-through log of advertisements collected from a commercial search engine can help online advertising. Beales [19] 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. [20] 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 [6,7] or the capabilities of a recommender system [21–23] to improve the effectiveness of the advertising have been proposed, but none of them generates segments of target users.

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* [24] or *commonsense* [25] 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 [12]. On the other hand, *post hoc* [26] approaches (also known as *a posteriori* [24] or *data-driven* [25]) combine a set of features (which are known as *segmentation base* [27]) 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 [13,14]. This is mostly due to the lack of guidance on how to interpret the results of a segmentation [15]. Regarding the literature on behavioral user segmentation, Bian et al. [28] presented an approach to leverage historical user activity on real-world Web portal services to build a behavior-driven user segmentation. Yao et al. [29] 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 characteristics. Zhou et al. [30] 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 [2] and Gong et al. [1] 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. [31] 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.

[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/.

Preference stability. As mentioned in the Introduction, Burke and Ramezani highlighted that some domains are characterized by a stability of the preferences over time [9]. Preference stability leads also to the fact that when users get in touch with diverse items, diversity is not valued [32]. On the one side, users tend to access to agreeable information (a phenomenon known as *filter bubble* [33]) and this leads to the overspecialization problem [10], 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 [34]. The literature presents several approaches that try to reduce this problem, e.g., *NewsCube* [35] operates by offering to the users several points of view, in order to stimulate them to make different and unusual choices.

3. Preliminaries

Background. For many years the item descriptions were analyzed through a word vector space model, where all the words of each item description are processed by TF-IDF [36] 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, because it does not adopt any semantic data model [37], and motivated by the fact that the exploitation of a taxonomy for categorization purposes is an approach recognized in the literature [38], we decided to use 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 155287 words, organized into 117659 synsets, for a total of 206941 word-sense pairs [39]. In a short, the main relation among words in WordNet is the synonymy and the synsets are 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 synsets, in this way each form-meaning pair in WordNet will be 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 a hypernym or *is-a* relationship. 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)*

We use the synsets to perform both the definition of binary filters and the evaluation of the relevance scores of the classes in a user profile.

Problem definition. Here, we define the problem handled by our proposal. A set of definitions will first allow us to introduce the notation used in the problem statement.

Definition 3.1 (*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 3.2 (*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 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. 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 3.3 (*Semantic Item 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 $|d_i| = |BoW|$). 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 $D = \{sd_1, \ldots, sd_M\}$ (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 4.1.

Definition 3.4 (*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 4.3.

Definition 3.5 (*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 4.3.

Definition 3.6 (*User Segment*)**.** Given a set of users U and a (boolean) class c_q, a user segment is a subset of users to target $T \subseteq U$ whose positively evaluated items I_u are semantically related to the items that belong to c_q.

Problem 1. 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 segment) $T \subseteq U$, 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).

4. Applied strategy

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. *Segment definition*: selection of the users characterized by a class or a boolean class.

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

4.1. Text preprocessing

Before extracting the WordNet synsets from the text that describes each item, we need to follow several preprocessing tasks. The first 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* [40]. Then, 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*[11]). After we have determined the lemma of each word using the Java API implementation for WordNet Searching JAWS,[12] 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 found using the Java implementation of the adapted Lesk algorithm provided by the *Denmark Technical University* similarity application [41]. All the collected synsets form the set $S = \{s_1, \ldots, s_W\}$ defined in Section 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 ds_i; each element of the vector $ds_i[w]$ is 1 if the corresponding synset is a part of the item description, or it is 0 otherwise.

4.2. 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

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

order to build m_u, this step performs the following operation on each element w:

$$m_u[w] = \begin{cases} 1, & \text{if } ds_i[w] = 1 \\ m_u[w], & \text{otherwise.} \end{cases} \quad (1)$$

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 $|M| = |U|$).

4.3. 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 the synsets that 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 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.

4.3.1. 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 $|B| = |C|$, 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., $|b_k| = |S|$). 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:

$$b_k[w] = \begin{cases} 1, & \text{if } ds_i[w] = 1 \wedge i \in c_k, \ \forall i \in I_+ \\ b_k[w], & \text{otherwise.} \end{cases} \quad (2)$$

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.

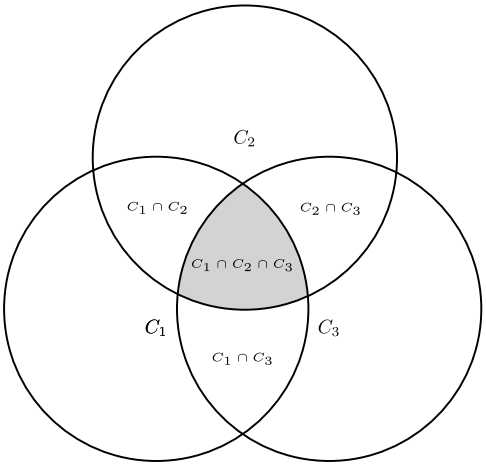


Fig. 1. Inter-class definition.

4.3.2. 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., a 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:

$$b_k[w] \wedge b_q[w] = \begin{cases} 1, & \text{if } b_k[w] = 1 \text{ and } b_q[w] = 1 \\ 0, & \text{otherwise.} \end{cases} \quad (3)$$

This boolean class indicates which synsets characterize all the classes of items involved. We can obtain this result by 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 Fig. 1 is a simple demonstration of what said based on the axiomatic set theory. It describes the effect of a boolean *AND* operation applied 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 segment 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 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* operator. 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. In other words, this SBS has more non-relevant synsets with respect to the original classes (this is represented by the percentage of zero 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.

4.3.3. 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

Table 1
Example of interclass-based SBS considering the two classes with most items.

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 2
Example of superclass-based SBS considering the two classes with most items.

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 3
Example of interclass-based SBS considering the two classes with most items.

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

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:

$$b_k[w] \vee b_q[w] = \begin{cases} 1, & \text{if } b_k[w] = 1 \text{ or } b_q[w] = 1 \\ 0, & \text{otherwise.} \end{cases} \quad (4)$$

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 a union operation $\cup$ of two or more SBSs.

The example in Fig. 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 2 shows an example in which classes 1 and 5 are combined with an *OR* operator. 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 zero occurrences), 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.

4.3.4. 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 genre 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

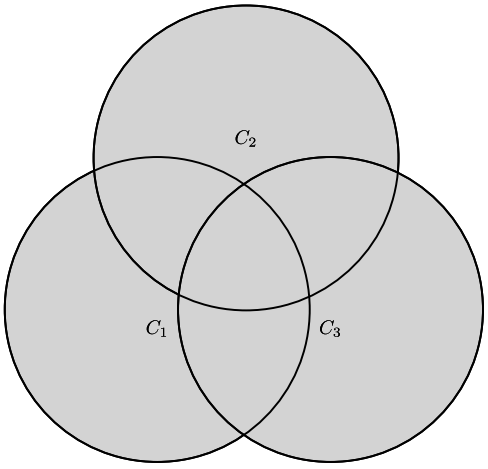


Fig. 2. Superclass definition.

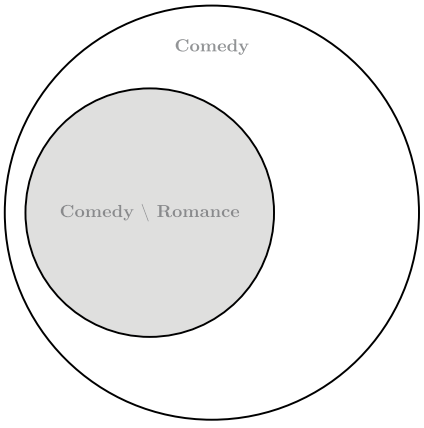


Fig. 3. Sub-class definition.

number of subclasses, applying some operation of the axiomatic set theory. In the example in Fig. 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 modifies the binary value of the destination bits, as shown in Eq. (5).

$$b_k[w] = \begin{cases} b_k[w], & \text{if } b_q[w] = 0 \\ 0, & \text{otherwise.} \end{cases} \quad (5)$$

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 3), in which we combine with a *NOT* operator 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, 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 synsets 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.

4.3.5. 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 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 Eq. (6).

$$\begin{aligned} x \wedge y &= \neg(\neg x \vee \neg y) \\ x \vee y &= \neg(\neg x \wedge \neg y). \end{aligned} \tag{6}$$

4.4. 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 the synsets that are relevant for the class c_k). The objective is to build a relevance score $r_u[k]$ that 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*), and this 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 semantics of the operator is shown in Eq. (7).

$$b_k[w] \Theta m_u[w] = \begin{cases} r_u[k]++, & \text{if } m_u[w] = 1 \text{ and } b_k[w] = 1 \\ r_u[k], & \text{otherwise.} \end{cases} \tag{7}$$

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.

4.5. Segment 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:

$$f(c) = \{u \in U \mid r_u(c) \geq \varphi\} \tag{8}$$

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.

5. Experiments

This section describes the experiments performed to validate our proposal. In Section 5.1 we describe the experimental setup and strategy, in Section 5.2 the dataset employed for the evaluation is presented, Section 5.3 illustrates the metrics, and Section 5.4 contains the results.

5.1. Experimental setup and strategy

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).[13] The experimental framework was developed by using a machine with an Intel i7-4510U, quad core (2 GHz $\times$ 4) and a Linux 64-bit Operating System (Debian Jessie) with 4 GBytes of RAM. To validate our proposal, we performed five sets of experiments:

1. *Data overview*. This experiment studies 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 ratings), in order to evaluate how trivial it is to perform a segmentation based on the classes; we also analyze the number of genres with which each item is classified, 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 validate this choice by evaluating the difference between the number of characterizing bits both in a binary vector 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 sets the threshold for each class by employing the elbow method, which evaluates the relevance score of each user for a class and detects the point in which the 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 analyzes 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 evaluate the number of seconds 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 computational efforts, so this analysis allows us to evaluate the performance of the approach from different perspectives.

It should be observed that in order to validate the capability of our proposal to detect users who are not characterized by explicit preferences for a class, we compare with the so-called topic-based approach employed by both Google's AdWords and Facebook's Core Audiences. In order to do so, in the experiments number 4 and 5, we 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.

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

Table 4
Yahoo! Webscope R4 Genres.

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		

5.2. Dataset

The used dataset, i.e., Yahoo! Webscope Movie Dataset (R4), contains a large amount of data related to user preferences expressed by the Yahoo! Movies community that are rated on the base of two different scales, from 1 to 13 and from 1 to 5 (we have chosen to use the latter). The training data is composed by 7642 users ($|U|$), 11915 movies/items ($|I|$), and 211231 ratings ($|R|$). The average user rating ($\overline{R_u} = \frac{\sum_u \overline{r_u}}{|U|}$, *macro-averaged*) is 3.70 and the average item rating (*macro-averaged*) is 3.58. The average number of ratings per user is 27.64 and the average number of ratings per item is 17.73. All users have rated at least 10 items and all items are rated by at least one user. The density ratio ($\delta = \frac{|R|}{|U|*|I|}$) is 0.0023, meaning that only 0.23% of entries in the user-item matrix are filled.

As shown in Table 4, the items are classified by Yahoo in 19 different classes (movie genres), and it is should be noted that each item may be classified in multiple classes.

5.3. Metric

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 score as discriminant), we use the well-known *elbow method*. In other words, we increase the relevance score value and calculate the variance (as shown in Eq. (9), where x denotes the number of users involved, and n is the relevance score) of the users involved: at the beginning we can note a low level of variance, but at some point the level suddenly increases; following the *elbow method* we chose as threshold value the number of synset occurrences used at this point.

$$S^2 = \frac{\sum(x_i - \bar{x})}{n-1}. \tag{9}$$

5.4. Experimental results

This section presents the results of each experiment previously presented.

5.4.1. 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, Fig. 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).

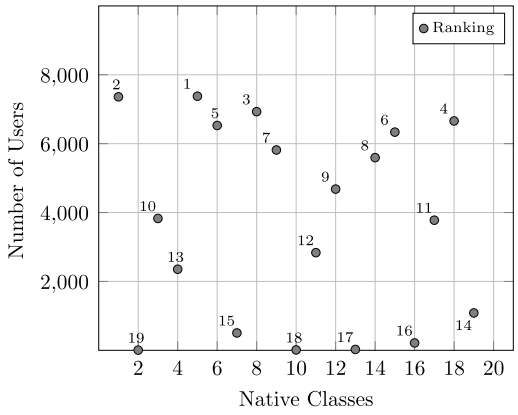


Fig. 4. User distribution for native classes.

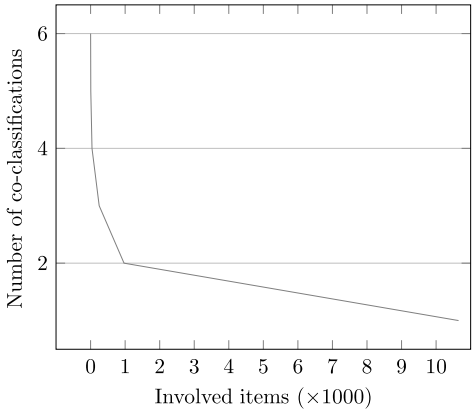


Fig. 5. Number of coclassification for item.

In Fig. 5 we see the number of items that have been classified with multiple genres. The results show that most of the items have 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 synsets contained in an item description characterizes the SBS of just one class (i.e., the SBSs will not be similar, since disjoint sets of items contribute to each binary vector).

5.4.2. Role of the semantics in the SBS data structure

In order to validate our choice to represent a SBS as 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 5 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

Table 5
Synsets and words cardinality.

Words	63772
Synsets	91130
Difference	+30.02%

Table 6
Elbow values.

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

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.

5.4.3. Setting of the φ parameter

In order to set the value of φ that allows us to consider a class as relevant for a user, we adopted the elbow method introduced in Section 5.3. Table 6 shows the threshold values derived from elbow method, 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 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.

5.4.4. Analysis of the segments

In this section, we analyze the produced user segments. For each of the primitive classes, we 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[14]); this allowed us to 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 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 included in 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, *co-classification* 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),[15] 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 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 Fig. 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, regardless of which class is considered. A related and important characteristic of our approach, is its capability to *detect a 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, i.e., 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 method 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 to those reported for the primitive classes. These results confirm the capability of our approach to work well when few 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 (3 times 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).

5.4.5. Performance analysis

Fig. 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.

[14] 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.

[15] 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.

Table 7
Experiments result.

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

Note that to simplify the readability of the results we report just the performance of 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, and presented in Section 4.1, 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).

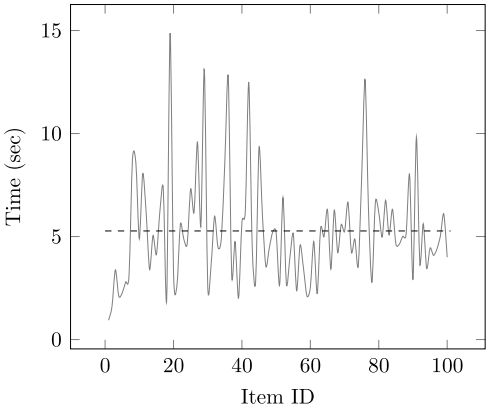


Fig. 6. Execution time.

6. Conclusions and future work

This paper presented a novel semantic user segmentation approach that exploits the description of the items positively evaluated by the users. The detection of the segments is based on the definition of a set of *binary sieves*, new entities that allow us to characterize primitive or boolean classes (i.e., set of classes combined through boolean operations). The experimental results show the ability of our semantic approach to model in an effective way a target of users within the domain taken into account. Future work will test its capability to characterize clusters of users whose purchased items are semantically related. This will allow us to target the users in a different way, e.g., by performing group recommendations to them (i.e., by suggesting items to groups of "semantically similar" users).

Acknowledgments

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".

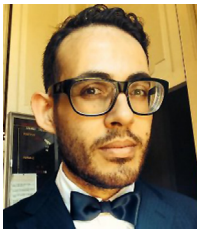
References

[1] X. Gong, X. Guo, R. Zhang, X. He, A. Zhou, Search behavior based latent semantic user segmentation for advertising targeting, in: 2013 IEEE 13th International Conference on Data Mining, ICDM, 2013, pp. 211–220. http://dx.doi.org/10.1109/ICDM.2013.62.

[2] S. Tu, 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, Vol. Part II, ADMA'10, Springer-Verlag, Berlin, Heidelberg, 2010, pp. 259–269. URL http://dl.acm.org/citation.cfm?id=1948448.1948476.

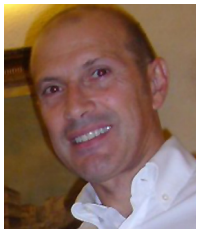
[3] A. Spink, B.J. Jansen, D. Wolfram, T. Saracevic, From e-sex to e-commerce: Web search changes, Computer 35 (3) (2002) 107–109. http://dx.doi.org/10.1109/2.989940.
[4] S.Y. Rieh, H.I. Xie, Analysis of multiple query reformulations on the web: The interactive information retrieval context, Inf. Process. Manage. 42 (3) (2006) 751–768. http://dx.doi.org/10.1016/j.ipm.2005.05.005.
[5] P. Boldi, F. Bonchi, C. Castillo, 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, Vol. 01, WI-IAT '09, IEEE Computer Society, Washington, DC, USA, 2009, pp. 183–190. http://dx.doi.org/10.1109/WI-IAT.2009.34.
[6] G. Armano, A. Giuliani, E. Vargiu, Semantic enrichment of contextual advertising by using concepts, in: J. Filipe, A.L.N. Fred (Eds.), Proceedings of the International Conference on Knowledge Discovery and Information Retrieval, KDIR 2011, Paris, France, 26–29 October, 2011, SciTePress, 2011, pp. 232–237.
[7] G. Armano, A. Giuliani, E. Vargiu, 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, IEEE Computer Society, 2011, pp. 172–176. URL http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=6059238.
[8] R. Saia, L. Boratto, 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.
[9] R.D. Burke, M. Ramezani, Matching recommendation technologies and domains, in: F. Ricci, L. Rokach, B. Shapira, P.B. Kantor (Eds.), Recommender Systems Handbook, Springer, 2011, pp. 367–386. URL http://www.springerlink.com/content/978-0-387-85819-7.
[10] P. Lops, M. de Gemmis, G. Semeraro, Content-based recommender systems: State of the art and trends, in: F. Ricci, L. Rokach, B. Shapira, P.B. Kantor (Eds.), Recommender Systems Handbook, Springer, 2011, pp. 73–105.
[11] C. Gustav Johannsen, Understanding users: from man-made typologies to computer-generated clusters, New Libr. World 115 (9/10) (2014) 412–425. http://dx.doi.org/10.1108/NLW-05-2014-0052.
[12] S.C. Bourassa, F. Hamelink, M. Hoesli, B.D. MacGregor, Defining housing submarkets, J. Hous. Econ. 8 (2) (1999) 160–183. http://dx.doi.org/10.1006/jhec.1999.0246. URL http://www.sciencedirect.com/science/article/pii/S1051137799902462.
[13] A. Nairn, P. Bottomley, Something approaching science? cluster analysis procedures in the crm era, in: H.E. Spotts (Ed.), Proceedings of the 2002 Academy of Marketing Science (AMS) Annual Conference, Developments in Marketing Science: Proceedings of the Academy of Marketing Science, Springer International Publishing, 2003, http://dx.doi.org/10.1007/978-3-319-11882-6_40. 120–120.
[14] S. Dolnicar, K. Lazarevski, Methodological reasons for the theory/practice divide in market segmentation, J. Mark. Manag. 25 (3–4) (2009) 357–373. http://dx.doi.org/10.1362/026725709X429791.
[15] S. Dibb, L. Simkin, A program for implementing market segmentation, J. Bus. Ind. Mark. 12 (1) (1997) 51–65. http://dx.doi.org/10.1108/08858629710157931.
[16] H. Zhuge, Semantic linking through spaces for cyber-physical-socio intelligence: A methodology, Artificial Intelligence 175 (5–6) (2011) 988–1019. http://dx.doi.org/10.1016/j.artint.2010.09.009.
[17] H. Zhuge, Interactive semantics, Artificial Intelligence 174 (2) (2010) 190–204. http://dx.doi.org/10.1016/j.artint.2009.11.014.
[18] J. Yan, N. Liu, G. Wang, W. Zhang, Y. Jiang, Z. Chen, How much can behavioral targeting help online advertising? in: Proceedings of the 18th International Conference on World Wide Web, WWW '09, ACM, New York, NY, USA, 2009, pp. 261–270. http://dx.doi.org/10.1145/1526709.1526745. URL http://doi.acm.org/10.1145/1526709.1526745.
[19] H. Beales, The value of behavioral targeting, Network Advertising Initiative.
[20] Y. Chen, D. Pavlov, J.F. Canny, Large-scale behavioral targeting, in: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '09, ACM, New York, NY, USA, 2009, pp. 209–218. http://dx.doi.org/10.1145/1557019.1557048. URL http://doi.acm.org/10.1145/1557019.1557048.
[21] G. Armano, E. Vargiu, A unifying view of contextual advertising and recommender systems, in: A.L.N. Fred, J. Filipe (Eds.), Proceedings of the International Conference on Knowledge Discovery and Information Retrieval, KDIR 2010, Valencia, Spain, October 25–28, 2010, SciTePress, 2010, pp. 463–466.
[22] A. Addis, G. Armano, A. Giuliani, 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.
[23] E. Vargiu, A. Giuliani, G. Armano, Improving contextual advertising by adopting collaborative filtering, ACM Trans. Web 7 (3) (2013) 13:1–13:22. http://dx.doi.org/10.1145/2516633.2516635. URL http://doi.acm.org/10.1145/2516633.2516635.
[24] J. Mazanee, Market segmentation, in: Encyclopedia of Tourism, Routledge, London, 2000.
[25] S. Dolničar, Beyond "commonsense segmentation": A systematics of segmentation approaches in tourism, J. Travel Res. 42 (3) (2004) 244–250.
[26] J.H. Myers, E.M. Tauber, Market Structure Analysis, American Marketing Association, 1977.
[27] M. Wedel, W.A. Kamakura, Market Segmentation: Conceptual and Methodological Foundations, in: International Series in Quantitative Marketing, Kluwer Academic Publishers, 2000.
[28] J. Bian, A. Dong, X. He, S. Reddy, Y. Chang, User action interpretation for online content optimization, IEEE Trans. Knowl. Data Eng. 25 (9) (2013) 2161–2174. http://dx.doi.org/10.1109/TKDE.2012.130.
[29] Z. Yao, T. Eklund, 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, ICDMW '10, IEEE Computer Society, Washington, DC, USA, 2010, pp. 639–646. http://dx.doi.org/10.1109/ICDMW.2010.121.
[30] Y.K. Zhou, 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, EC-Web'06, Springer-Verlag, Berlin, Heidelberg, 2006, pp. 11–20. http://dx.doi.org/10.1007/11823865_2.
[31] X. Wu, J. Yan, N. Liu, S. Yan, Y. Chen, Z. Chen, Probabilistic latent semantic user segmentation for behavioral targeted advertising, in: Proceedings of the Third International Workshop on Data Mining and Audience Intelligence for Advertising, ADKDD '09, ACM, New York, NY, USA, 2009, pp. 10–17. http://dx.doi.org/10.1145/1592748.1592751. URL http://doi.acm.org/10.1145/1592748.1592751.
[32] S.A. Munson, P. Resnick, Presenting diverse political opinions: How and how much, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '10, ACM, New York, NY, USA, 2010, pp. 1457–1466. http://dx.doi.org/10.1145/1753326.1753543. URL http://doi.acm.org/10.1145/1753326.1753543.
[33] E. Pariser, The Filter Bubble: What the Internet Is Hiding from You, Penguin Group, The, 2011.
[34] L. Festinger, A Theory of Cognitive Dissonance, Vol. 2, Stanford University Press, 1962.
[35] S. Park, S. Kang, S. Chung, J. Song Jr., Newscube: delivering multiple aspects of news to mitigate media bias, in: D.R. O., R.B. Arthur, K. Hinckley, M.R. Morris, S.E. Hudson, S. Greenberg (Eds.), Proceedings of the 27th International Conference on Human Factors in Computing Systems, CHI 2009, Boston, MA, USA, April 4–9, 2009, ACM, 2009, pp. 443–452. http://dx.doi.org/10.1145/1518701.1518772. URL http://doi.acm.org/10.1145/1518701.1518772.
[36] G. Salton, C. Buckley, Term-weighting approaches in automatic text retrieval, Inf. Process. Manage. 24 (5) (1988) 513–523. http://dx.doi.org/10.1016/0306-4573(88)90021-0.
[37] H. Zhuge, Y. Sun, The schema theory for semantic link network, Future Gener. Comput. Syst. (3) (2010) 408–420. http://dx.doi.org/10.1016/j.future.2009.08.012.
[38] A. Addis, G. Armano, E. Vargiu, Assessing progressive filtering to perform hierarchical text categorization in presence of input imbalance, in: A.L.N. Fred, J. Filipe (Eds.), Proceedings of the International Conference on Knowledge Discovery and Information Retrieval, KDIR 2010, Valencia, Spain, October 25–28, 2010, SciTePress, 2010, pp. 14–23.
[39] C. Fellbaum, WordNet: An Electronic Lexical Database, Bradford Books, 1998.
[40] K. Toutanova, D. Klein, C.D. Manning, 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, Vol. 1, NAACL '03, Association for Computational Linguistics, Stroudsburg, PA, USA, 2003, pp. 173–180. http://dx.doi.org/10.3115/1073445.1073478.
[41] G. Salton, A. Wong, C.S. Yang, A vector space model for automatic indexing, Commun. ACM 18 (11) (1975) 613–620.

Roberto Saia is a Ph.D.c at the Department of Mathematics and Computer Science of the University of Cagliari. He got a Master Degree in Computer Science at the same University. His current research activity is focused on the development of techniques and algorithms able to improve the effectiveness of the user profiling and item recommendation in web-based environments.



Ludovico Boratto is a post-doc at the University of Cagliari, Italy. He graduated with full marks and honor and received his Ph.D. in 2012 at the same University. His research focuses mainly on recommender systems and data mining in social networks.

Salvatore Carta graduated received a Ph.D. in Electronics and Computer Science from the University of Cagliari in 2003. He is Associate Professor in Computer Science at the University of Cagliari since 2014. Recently, he has focused on topics related to the social Web, ubiquitous computing and computational societies. In particular he works on algorithms for social search and recommendation, and on algorithms and strategies in the fields of mobile Human Computer Interaction and fitness games.



Gianni Fenu is an Associate Professor in Computer Science at the University of Cagliari. He has more than 70 scientific papers, published in the proceedings of international conferences and international journals, mainly in the fields of integrated information systems, high speed networks, and distributed systems evaluation criteria. Recently, he has focused on topics related to the Human–Computer interaction, ubiquitous computing, and the social Web.