A Latent Semantic Pattern Recognition Strategy for an Untrivial Targeted Advertising

Roberto Saia, Ludovico Boratto, Salvatore Carta
Dipartimento di Matematica e Informatica
Università di Cagliari
Via Ospedale 72, 09124 - Cagliari, Italy
Email: {*roberto.saia, ludovico.boratto, salvatore*}*@unica.it*

***Abstract*—Target definition is a process aimed at partitioning the potential audience of an advertiser into several classes, according to specific criteria. Almost all the existing approaches take into account only the explicit preferences of the users, without considering the hidden semantics embedded in their choices, so the target definition is affected by widely-known problems. One of the most important is that easily understandable segments are not effective for marketing purposes due to their triviality, whereas more complex segmentations are hard to understand. In this paper we propose a novel segmentation strategy able to uncover the implicit preferences of the users, by studying the semantic overlapping between the classes of items positively evaluated by them and the rest of classes. The main advantages of our proposal are that the desired target can be specified by the advertiser, and that the set of users is easily described by the class of items that characterizes them; this means that the complexity of the semantic analysis is hidden to the advertiser, and we obtain an interpretable and non-trivial user segmentation, built by using reliable information. Experimental results confirm the effectiveness of our approach in the generation of the target audience.**

***Keywords*-data mining; pattern recognition; user segmentation; advertising targeting; semantic analysis;**

I. Introduction

Behavioral targeting is the strategy used to identify sets of users who share common properties. This process allows the advertisers to address ads toward a specific set of users. In order to choose these sets, 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 user, 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, such as: *the reliability of a semantic query analysis*, since half of the time the users need to reformulate their queries in order to satisfy their information need [3]; *the preference stability*, that afflicts some domains like movies, in which the user preferences tend to be stable over time (this would lead to trivial segments, in which each user is associated to a small and obvious set of segments) [4]; and the *interpretability of the segments*, since an effective segmentation can be built only by understanding the users [5]. In this paper, we tackle the problem of *defining a semantic behavioral targeting approach, such that the sources of information used to build it are reliable, the generated user segmentation is not trivial and it is easily interpretable*. In order to solve the problem of using reliable sources of information, our proposal is based on a semantic analysis of the description of the items positively evaluated by the users, since it the literature it is known that performing a semantic analysis on the description of the items can increase the accuracy of a system [6]. 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 terms characterize it (this creates a unique semantic pattern for each class). Then, for each item positively evaluated by a user, we consider the terms (that as we will explain later, are actually particular semantic entities named *synsets*) that describe it, and use the previously created filters to evaluate a *relevance score* that indicates how relevant is that class for the user; this is done by performing a pattern comparison between the user profile and all the semantic binary sieves, able to detect any latent semantic connection between classes. The relevance scores of each user (stored in a structure called *class path vector*) are filtered by the segmentation algorithm, in order to return all the users characterized by a specified class. The contributions of our proposal are the following:

- definition of the *Semantic Binary Sieves* (SBS) filters, able to weigh the classes relevance in the user profiles;
- creation of the *Class Path Vector* (CPV) model, used to evaluate the interest by class of each user;
- evaluation of the proposed strategy on two real-world datasets, comparing it with the widely used k-means approach and with a baseline *native classification* that do not exploit the semantics behind the user behavior.

The rest of the paper is organized as follows: we first present the works related with our approach (Section II), then we introduce the notation and the problem definition (Section III), continuing with the implementation details (Section IV) and the performed experiments (Section V), and ending with some concluding remarks (Section VI).

II. Related Work

Behavioral targeting. Beales [7] 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. [8] 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 [9], [10] or the capabilities of a recommender system [11], [12], [13] to improve the advertising effectiveness have been proposed, but they do not segment the users.

Segment interpretability and semantic user segmentation. Choosing the right criteria to segment users is a widely studied problem in the literature, and two main classes of approaches exist. On the one hand, the *a priori* [14] 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 [15]. On the other hand, *post hoc* [16] approaches (also known as *a posteriori* [14]) combine a set of features (which are known as *segmentation base*) 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 [5]. Regarding the literature on behavioral user segmentation, Bian et al. [17] presented an approach to leverage historical user activity on real-world Web portal services to build behavior-driven user segmentation. Yao et al. [18] adopted SOM-Ward clustering (i.e., Self Organizing Maps, combined with Ward clustering), to segment a set of customers based on their demographic and behavioral characteristic. Zhou et al. [19] 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 an Latent Dirichlet Allocation-based models, in which users with similar query and click behaviors are grouped together. Similarly, Wu et al. [20] performed a semantic user segmentation by adopting a Probabilistic Latent Semantic Approach on the user queries. To summarize, 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.

Preference stability. As mentioned in the Introduction, Burke and Ramezani highlighted that some domains are characterized by a stability of the preferences over time [4]. Preference stability leads also to the fact that when users get in touch with diverse items, diversity is not valued [21]. On the one side, users tend to access to agreeable information (a phenomenon known as *filter bubble* [22]) and this leads to the overspecialization problem [23], 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 [24]. The literature presents several approaches that try to reduce this problem, e.g., *NewsCube* [25].

III. Notation and Problem Definition

Notation. We consider a set of users $U = \{u_1, \ldots, u_N\}$, a set of items $I = \{i_1, \ldots, i_M\}$, and a set V of values used to express the user preferences (e.g., $V = [1, 5]$ or $V = \{like, dislike\}$). The set of possible preferences expressed is a ternary relation $P \subseteq U \times I \times V$. We denote as $P_+ \subseteq P$ the subset of preferences with a positive value (i.e., $P_+ = \{(u, i, v) \in P | v \geq \overline{v} \vee v = like\}$), where $\overline{v}$ indicates the mean value (in the previous example, $\overline{v} = 3$). Moreover, we denote as $I_+ = \{i \in I | \exists(u, i, v) \in P_+\}$ the set of items for which there is a positive preference, and as as $I_u = \{i \in I | \exists(u, i, v) \in P_+ \wedge u \in U\}$ the set of items a user u likes. Let $C = \{c_1, \ldots, c_K\}$ be a set of classes that 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 $C_u = \{c \in C | \exists(u, i, v) \in P_+ \wedge i \in C_i\}$ the classes associated to the items that a user likes. Let $BoW = \{t_1, \ldots, t_W\}$ be the bag of words used to describe the items in I; we denote as d_i be 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 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 $D = \{sd_1, \ldots, sd_M\}$ (note that we have a semantic description for each item, so $|D| = |I|$).

Problem definition. Given a set of positive preferences P_+ that characterizes the items each user likes, a set of classes C used to classify the items, 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. Each relevance score will be combined into a model CPV_u, defined as follows:

$$CPV_u = (r_u(c_1), \ldots, r_u(c_K)) \tag{1}$$

Each CPV_u must respect the following property $r_u(c_1) \geq r_u(c_2) \geq ... \geq r_u(c_K)$. So, each CPV model contains a list of classes ranked by relevance score. To face the *triviality* problem our aim is to derive a binary model able to describe the distribution of the items by classes. The objective of our approach is to define a function $f : C \rightarrow U$ that, given a class $c \in C$, returns a set of users (user target) $T \subseteq U$ that have c as the most relevant class, i.e., such that $\forall u \in T, c_1 = c$ (where c_1 refers to the order in CPV_u).

IV. Applied strategy

Here we describe the steps performed by our approach:

1) **Textual information**: processing of the textual information of the items, to remove the useless elements;
2) **User model**: creation of a model that contains which synsets are present in the items a user likes;
3) **Semantic Binary Sieve**: definition of the binary filters, named *Semantic Binary Sieves* (SBS), able to estimate which synsets are relevant for a class;
4) **Class Path Vector**: definition of the *Class Path Vector* (CPV) model, adopted to weight the user preferences in terms of classes by means of the SBS;
5) **User Targeting**: selection of the users characterized by a specified class.

A. Text preprocessing

Exploring a taxonomy for categorization purposes is an approach adopted in the literature [26]. In this paper we consider the taxonomy of WordNet, a lexical database of English, where nouns, verbs, adjectives, and adverbs are grouped into sets of cognitive synonyms (synsets) that express a distinct concept. Before extracting the synsets from the text that describes an item, several preprocessing steps are followed: detect the correct *Part-Of-Speech* for each word (by the *Stanford Log-linear Part-Of-Speech Tagger* [27]); remove punctuation marks and *stop-words*; determinine the lemma of each word (by JAWS[1]); detect the best sense of each word (by the adapted Lesk algorithm [28]). The result is the semantic disambiguation of the textual description of each item $i \in I$, which is stored in a binary vector ds_i, where each element $ds_i[w]$ is 1 if the synset is a part of the description, and 0 otherwise.

B. User modeling

For each user $u \in U$, we consider the set of items I_u she/he likes, and build a user model m_u that describes which synsets appear in the semantic description of these items. Each model m_u is a binary vector with an element for each synset $s_w \in S$, and to build the vector we consider the semantic description ds_i of each item $i \in I_u$ (those with a positive rating), then build m_u performing the following operation on each element w:

$$m_u[w] = \begin{cases} 1, \; if \; ds_i[w] = 1 \\ m_u[w], \; otherwise \end{cases} \tag{2}$$

This means that if the semantic description of an item i contains the synset s_w, it 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 $M = \{m_1, \dots, m_N\}$ of user models, with $|M| = |U|$).

C. Semantic Binary Sieve definition

For each class $c \in C$, we create a binary vector that will store which synsets are relevant for that class. These vectors, called *Semantic Binary Sieves*, are stored in a set $B = \{b_1, \dots, b_K\}$ (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 $i \in I$, and each class c_k with whom i was classified. The binary vector b_k will store which synsets are relevant for a class c_k, by performing the following operation on each element $b_k[w]$ of the vector:

$$b_k[w] = \begin{cases} 1, \; if \; ds_i[w] = 1 \wedge i \in c_k \\ b_k[w], \; otherwise \end{cases} \tag{3}$$

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 position w of the binary sieve b_k associated to c_k is set to 1; otherwise, its value remains unaltered. Algorithm 1, summarizes this process.

Algorithm 1 CreateSBS

```
Input:  $c_k \in C$=Class to evaluate, $I$=Items in dataset
Output: $b_k$ = SBS of the class $c_k$, with $|b_k| = |S|$
procedure CreateSBS($c_k, I$)
    for each $i$ in $I$ do
        $ds_i$=GetSemanticDescription($i$)
        for each element $w$ in $ds_i$ do
            if $ds_i[w] == 1$ AND $i \in c_k$ then $b_k[w] = 1$
            end if
        end for
    end for
    return $b_k$
end procedure
```

D. Class Path Vector definition

In this step we compare the sets B and M. 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 stored in each vector b_k, which contains the synsets that are relevant for the class c_k). The aim is to build a relevance score $r_u[k]$, which indicates the relevance of the class c_k for the user u. Each vector in B is used as a filter in order to estimate the relevance of each class for a user. By ordering the relevance scores from the most to the least relevant, we build a model named *Class Path Vector (CPV)*, which is used to perform the targeting. We consider each

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

semantic binary sieve $b_k \in B$ associated to the class c_k and the user model m_u, and define a matching criteria Θ between each synset $m_u[w]$ in the user model and the corresponding synset $b_k[w]$ in the semantic binary sieve, by adding 1 to the relevance score of that class for the user (element $r_u[k]$) if the synset is set to 1 both in the semantic binary sieve and in the user model, and leaving the current value as it is otherwise. The semantic of the operator is shown in (4).

$$b_k[w]\Theta m_u[w] = \begin{cases} r_u[k]++, \; if \; m_u[w] = 1 \wedge b_k[w] = 1 \\ r_u[k], \; otherwise \end{cases} \tag{4}$$

By comparing a user model m_u with each vector $b_k \in B$ (obtained by the Algorithm 1), we get a vector r_u that contains the relevance score of each class for the user (i.e., $|r_u| = |C|$). As shown in the Algorithm 2, the relevance scores of each class for each user are sorted in decreasing order to build the CPV model for a user u, i.e., each model respects the following property: $r_u(c_1) \geq \ldots \geq r_u(c_K)$:

$$CPV_u = (r_u(c_1), \ldots, r_u(c_K)) \tag{5}$$

Algorithm 2 CreateCPV

```
Input: $u \in U$=User to evaluate, $B$=SBSs, with $|B| = |C|$
Output: $r_u$ = CPV of the user $u$
procedure CREATECPV($u$,$B$)
    $m_u$=GetSynsetsInProfile($u$)
    for each $b_k$ in $B$ do
        for each element $w$ in $m_u$ do
            if $m_u[w] == 1$ AND $b_k[w] == 1$ then $r_u[k]++$
            end if
        end for
    end for
    return DescSort($r_u$)
end procedure
```

E. User Targeting

This step defines the set of users that are part of a target. Given a class of items $c \in C$, we build a function $f : C \rightarrow U$ that queries the CPV models previously built and 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. During the partitioning process based on our approach, each user is placed within the class with the highest value of relevance score. This is a choice made to compare our strategy both with the native classification of the items, and with the k-means approach.

V. Experiments

A. Experimental Setup

To conduct the experiments we adopted the Java language, with the support of Java API implementation for WordNet Searching (JAWS) to perform the semantic analysis. The real-world datasets used during the experiments are the Yahoo! Webscope Movie dataset (R4)[2], and the Movielens 10M[3] dataset. The software *KMlocal*, used to perform the k-means partitioning [29], implements several algorithms (i.e., *Lloyd's*, *Swap*, *Hybrid*, and *EZ-Hybrid*): for our experiments we choose to use the *EZ-Hybrid*, as it is the one that showed the best performance in terms of average distortion. The experiments are organized as follows: in the first part (presented in V-D1) we analyze the composition of the partitions created by the native classification, k-means, and SBS. The aim is to check if our approach is able to detect a number of users comparable with that of the native classification, and how its performance, compared with that of k-means, improves the targeting process; the next experiment (illustrated in V-D2) tests if the detected users (by k-means and SBS) match those of the native partitioning. This to verify if the partitions that we compare largely involve the same set of users (i.e., users with similar characteristics); in the last experiments (presented in V-D3) we conclude the experimentation by studying the semantic characteristics of the users in the SBS partitions. The *Jaccard index* is the metric adopted in the experiments, because it is widely used for comparing the similarity of sample sets. In our experiments we take into account the 5 largest partitions obtained by the different approaches (i.e., native classification, k-means, and our novel SBS approach), because the measurements (by Jaccard index) show that these largely involve the same users, regardless of the method of partitioning used. In this way, we are able to study how the users are aggregated by the different approaches of partitioning, using as a reference their classification in the real-world datasets.

B. Datasets and Data Preprocessing

Yahoo! Webscope (R4). This dataset contains a large amount of data related to users preferences expressed on the Yahoo! Movies community that are rated on the base of two different scales, from 1 to 13 and from 1 to 5 (we use the latter). The training data used in this work is composed by 7,642 users ($|U|$), 11,915 movies/items ($|I|$), and 211,231 ratings ($|R|$). The items are classified in 20 different classes (movie genres), and it should be noted that an item may be classified with multiple classes.

Movielens 10M. The second dataset used in this work is composed by 71,567 users ($|U|$), 10,681 movies/items ($|I|$), and 10,000,054 ratings ($|R|$). 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. The items are classified in 18 different classes (movie genres), and also in this case each item may be classified with multiple classes. Since the Movielens 10M dataset does not

[2] http://www.cs.umd.edu/˜mount/pubs.html

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

contain any textual description of the items, to obtain this information we used a file provided by the Webscope (R4) dataset, where the MovieLens movie IDs are mapped with those of Yahoo. For our experiments we extracted a subset of 10,000 users, 2,816 movies/items, and 1,046,202 ratings.

Data Preprocessing. In order to create a binary sieve for each class used to build a CPV model for every user, we need to define an ontology of synsets based on the descriptions of the items. To perform this operation we considered the description and title of each movie, and since the used algorithm takes into account only the items with a rating above the average, we selected only movies with a rating ≥ 3. In order to perform the k-means partitioning, we need to preprocess the datasets format (i.e., $[User\text{-}ID, Movie\text{-}ID, Rating]$), in order to convert it in the vector format (i.e., $[User\text{-}ID, c_1, c_2, \ldots, c_K], with\ K = |C|$). In the elements c we report the number of movies that belong to that class, and that have been positively evaluated by the user (namely those with a rating ≥ 3).

Native Classification. The *native classification* was created by selecting for each user the class c_k with the highest value, considering the vector format produced in the data preprocessing. This means that in the native classification a user is assigned to a class if most of the positively evaluated items belong to this one, while in the SBS partitioning, a user is assigned to the class indicated by the first position of her/his CPV (the class with the highest rating), the data structure formalized in Section IV-D.

C. Metrics

The evaluation of the results of the partitionings made by k-means and by our approach is performed with the *Jaccard index*. The index measures the similarity of two finite sets A and B, and is defined as the size of their intersection divided by the size of their union, i.e., $Jaccard(A, B) = \frac{|A \cap B|}{|A \cup B|}$. The possible values are in the range between 0 (complete diversity) and 1 (complete similarity). A value of 1 is assumed when both sets are empty.

D. Experimental Results

Here, we report the results of the experiments presented in the *Experimental Setup* (Section V-A).

1) **Partitions Composition:** Figure 1 shows the users distribution in the 5 largest partitions obtained by the native classification of the users in the used datasets, by k-means, and by our SBS-based approach. We can observe that in the Webscope dataset the partitioning performed by k-means divides the users among all classes in a quite uniform way, while in the Movielens dataset the partitioning is similar to the native classification. This happens because k-means is not able to evaluate the implicit semantic relationships between users, producing a trivial partitioning, based only on the explicit preferences. The results of SBS in both datasets, show instead a strong characterization of the users within few partitions (especially in the first one), virtue of the fact that it takes into account every semantic overlapping between the users preferences, introducing additional users that otherwise would do not be taken into account. The total number of users in each partitioning approach also shows that the SBS strategy involves pretty much the same number of users present in the native classification (4.66% more in the Webscope dataset, and 3.20% more in the Movielens dataset), unlike the k-means approach, which instead involves a much smaller number of users (31.69% less in the Webscope dataset, and 5.75% less in the Movielens dataset).

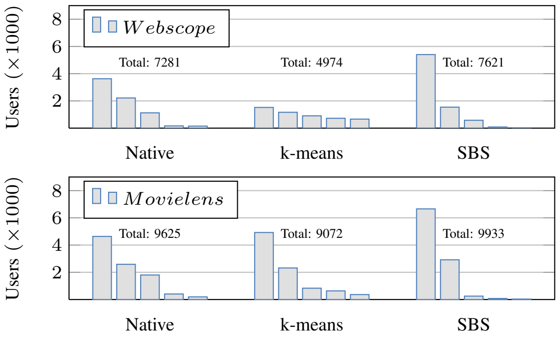

Figure 1: Partitions size

2) **Partitions Similarity:** This experiment aims to determine whether the users within the SBS partitions are relevant, comparing them with those produced by k-means. In other words, we want to know if SBS is able to detect the users of the native classification, adding to them additional semantically relevant users, thus improving the targeting process. For the reasons given in Section V-A, we do this by calculating the Jaccard index for the union of the first 5 largest partitions ($P_1, P_2, \ldots, P_5$), i.e., $1st$=$\{P_1\}$, $2nd$=$\{P_1 \cup P_2\}$,. . .,$5th$=$\{P_1 \cup P_2 \cup P_3 \cup P_4 \cup P_5\}$.

The results of Figure 2 clearly show the presence of a strong overlap between the users in the native classification and those classified by the SBS approach, differently from those classified by k-means.

3) **Segmentation Analysis:** Through this experiment we study the composition of the partitions produced by our SBS approach, i.e., we verify the nature of the additional subset of users identified by SBS in a partition p (denoted as $U^p_{(X)}$) but not present in either the native partitioning, nor in the one operated by k-means. With respect to a partition p, denoting as $U^p_{(N)}$ the set of users related to the native partitioning, as $U^p_{(K)}$ the set of users related to the k-means partitioning, and as $U^p_{(S)}$ the set of users related to the SBS partitioning, we extract the subset $U^p_{(X)}$ (the gray area in Figure 3) through the operation $U^p_{(X)} = (U^p_{(S)} \setminus U^p_{(N)}) \setminus U^p_{(K)}$, where the operator $\setminus$ denotes the subtraction of sets, according to the *set theory*. As shown in Table I and Table II, we detected more users

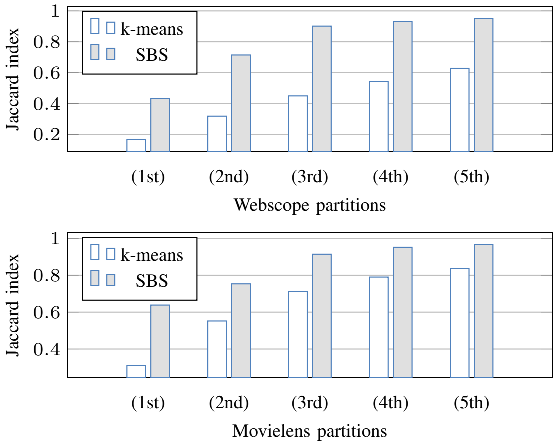

Figure 2: Jaccard index comparison

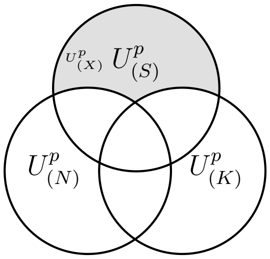

Figure 3: $U^p_{(X)}$ subset

top-p	$U_{(S)}$	$U_{(K)}$	$U_{(N)}$	$U_{(S)} \setminus U_{(N)}$	$U_{(X)}$
1	5404	1520	3628	2671	2152
2	1542	1164	2217	1005	841
3	578	904	1123	573	435
4	84	724	164	82	77
5	13	662	149	12	11
$\|U\|$	7621	4974	7281	4283	3516

Table I: Webscope Top Partitions Elements

top-p	$U_{(S)}$	$U_{(K)}$	$U_{(N)}$	$U_{(S)} \setminus U_{(N)}$	$U_{(X)}$
1	6657	4924	4636	2257	1191
2	2919	2318	2587	1029	803
3	249	832	1801	15	12
4	74	634	406	26	23
5	34	364	195	34	33
$\|U\|$	9933	9072	9625	3361	2062

Table II: Movielens Top Partitions Elements

Datasets	Webscope classes			Movielens classes		
top-p	*Native*	*SBS*	*K-means*	*Native*	*SBS*	*K-means*
1	1	8	-	8	8	-
2	5	5	-	5	5	-
3	8	1	-	1	1	-
4	17	19	-	16	16	-
5	19	13	-	2	15	-

Table III: Predominant Classes

in the largest partition (those in the subset $U^1_{(S)}$, with $|U^1_{(S)}| = 5404$, against the subset $U^1_{(N)}$, with $|U^1_{(N)}| = 3628$ in the Webscope dataset, and with $|U^1_{(S)}| = 9933$, against the subset $U^1_{(N)}$, with $|U^1_{(N)}| = 9625$) in the Movielens dataset. This result, as previously mentioned, is related to our semantic approach that is able to discover the latent semantic connections between the users that already exist in a partition and the new ones.

At this step we need to know if the users that appear in the SBS segmentation, but not in the native one, are semantically related with the others in the partition. To get this information we calculate the Jaccard index between each pair of involved SBS. In other words, considering that each class is characterized by a binary vector (SBS), through the Jaccard index we measure the shared synsets between a pair of SBS, namely the semantic relationships between the related classes. As explained in Section V-B, the SBS length (i.e., the number of distinct synsets $s \in S$) depends on the synset ontology: for Webscope (R4) this value is $|S| = 20,776$, while for Movielens 10M it is $|S| = 13,521$. It should be noted that we know the class of belonging of each user in the partitions created by the native classification and SBS approach (shown in Table III). This is possible because both approaches (unlike k-means that does not take in account the class of belonging of the users, operating by mere mathematical criteria) are based on the class predominance.

The first consideration about the information reported in Table III, is related with the CPV data structure, which reports the user preferences by class, deducting these through pure semantic criteria, i.e., regardless of the real classification of the items in the user profiles. The data show two scenarios: in the first of them, the users detected belong to different classes w.r.t. those assigned by the native classification of the dataset (5 out of 10 cases), while in the other one, they belong to the same classes (the remaining 5 cases). In all cases, however, our strategy is either able to detect a larger number of users, or it is capable to add additional semantically relevant users. The semantic relationship has been verified by calculating the semantic overlapping between the pairs of classes. For instance, in the first partition of the Webscope dataset, our approach detects a larger number of users than those in the native partitioning. The classification is different (i.e., the native class is 1, and the SBS class is 8), but a strong semantic similarity between them exists, with a Jaccard index of 0.7 (we get the same result with the native class 17 and the SBS class 19, and with the classes 19 and 13). Also in the Movielens dataset, we have one case of different classification but with an high index of semantic similarity (the native class 2 and the SBS class 15, which have a Jaccard index of 0.8). In the other scenario, when the classification is the same, and the Jaccard index is obviously equal to 1, we still improve the targeting process, introducing new users that otherwise are not taken into account. In every case, the results clearly confirms the existence of a strong semantic biunivocal relation between the native classes of assignment, and the classification operated by the SBS approach.

E. Discussion

The results of the first experiment, shown in Figure 1, indicate that the proposed approach is able to perform a strong characterization of the users, grouping them in partitions larger than those made with the other approaches. The experiment presented in Figure 2 also indicates that, not only SBS detects a larger number of users, but it also identifies other pertinent users, in addition to those already natively classified as relevant. The analysis of the new users detected by SBS (i.e., those not included in the partitions obtained through the native classification, or by the k-means process), reported in Table I and Table II, shows as our strategy is able to discover the latent semantic connections between the users already present in a partition and the new ones. The semantic overlapping has been verified by calculating the Jaccard index between the pairs of classes.

VI. Conclusions and Future Work

This paper proposed a novel semantic behavioral targeting approach, based on a latent semantic pattern recognition process able to uncover the implicit preferences of the users. Through the use of a set of *binary sieves*, we can investigate the semantic overlapping between the classes of items positively evaluated by a user and the rest of classes. In this work we compared our strategy only with the k-means approach (as it is one of most used methods), but we can state that also using others approaches that like k-means perform their partitioning applying *a priori* strategy, which is based on the explicit attributes of the users, the bad results would be the same. This is related to the consideration that to get good and non trivial results in the partitioning, we must necessarily adopt *a posteriori* approach, that allows us to exploit the implicit attributes of the users. The approach used in our strategy builds the knowledge about the users by taking in account the latent semantic connections between them. The experiments performed using two real-world datasets show the effectiveness of our model within the targeted advertising domain, but it could be also applied to any other domain characterized by a textual description of the items. The SBS approach improves the state of the art, overcoming the well-known problems related with the interpretability of the semantic strategies but, above all, the triviality of the results. Future work will test the capability of the proposed approach to characterize clusters of users whose purchased items are semantically related. This approach would allow us to target the users in a different way, e.g., by performing group recommendations to them (i.e., by recommending items to groups of "semantically similar" users).

Acknowledgment

This work is partially funded by Regione Sardegna under project SocialGlue, through PIA - Pacchetti Integrati di Agevolazione "Industria Artigianato e Servizi" (annualità 2010), and by MIUR PRIN 2010-11 under project "Security Horizons".

References

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

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

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

[4] R. D. Burke and M. Ramezani, "Matching recommendation technologies and domains," in *Recommender Systems Handbook*. Springer, 2011, pp. 367–386.

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

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

[7] H. Beales, "The value of behavioral targeting," *Network Advertising Initiative*, 2010.

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

[9] G. Armano, A. Giuliani, and E. Vargiu, "Semantic enrichment of contextual advertising by using concepts," in *KDIR 2011 - Proceedings of the International Conference on Knowledge Discovery and Information Retrieval, Paris, France, 26-29 October, 2011*. SciTePress, 2011, pp. 232–237.

[10] G. Armano, A. Giuliani, and E. Vargiu, "Studying the impact of text summarization on contextual advertising," in *2011 Database and Expert Systems Applications, DEXA, International Workshops, Toulouse, France, August 29 - Sept. 2, 2011*. IEEE Computer Society, 2011, pp. 172–176.

[11] G. Armano and E. Vargiu, "A unifying view of contextual advertising and recommender systems," in *KDIR 2010 - Proceedings of the International Conference on Knowledge Discovery and Information Retrieval, Valencia, Spain, October 25-28, 2010.* SciTePress, 2010, pp. 463–466.

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

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

[14] J. Mazanee, "Market Segmentation," in *Encyclopedia of Tourism*. London: Routledge, 2000.

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

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

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

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

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

[20] X. Wu, J. Yan, N. Liu, S. Yan, Y. Chen, and Z. Chen, "Probabilistic latent semantic user segmentation for behavioral targeted advertising," in *Proceedings of the Third International Workshop on Data Mining and Audience Intelligence for Advertising*, ser. ADKDD '09. New York, NY, USA: ACM, 2009, pp. 10–17.

[21] S. A. Munson and P. Resnick, "Presenting diverse political opinions: How and how much," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '10. New York, NY, USA: ACM, 2010, pp. 1457–1466.

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

[23] P. Lops, M. de Gemmis, and G. Semeraro, "Content-based recommender systems: State of the art and trends," in *Recommender Systems Handbook.* Springer, 2011, pp. 73–105.

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

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

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

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

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

[29] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, "An efficient k-means clustering algorithm: Analysis and implementation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 881–892, 2002.