Computers and Mathematics with Applications 58 (2009) 1498–1519

Contents lists available at ScienceDirect

Computers and Mathematics with Applications

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

An evaluation study of clustering algorithms in the scope of user communities assessment

Pantelis N. Karamolegkos, Charalampos Z. Patrikakis, Nikolaos D. Doulamis, Panagiotis T. Vlacheas, Ioannis G. Nikolakopoulos *

National Technical University of Athens, Telecommunications Laboratory, 9, Heroon Polytechniou str., Zografou, Athens, 157 73, Greece

ARTICLE INFO

Article history:
Received 1 October 2007
Received in revised form 28 April 2009
Accepted 29 May 2009

Keywords:
Spectral clustering
Modeling
Social networking
User profile
Performance evaluation

ABSTRACT

In this paper, we provide the results of ongoing work in Magnet Beyond project, regarding social networking services. We introduce an integrated social networking framework through the definition or the appropriate notions and metrics. This allows one to run an evaluation study of three widely used clustering methods (*k*-means, hierarchical and spectral clustering) in the scope of social groups assessment *and in regard to the cardinality of the profile used to assess users' preferences*. Such an evaluation study is performed in the context of our service requirements (i.e. on the basis of equal-sized *group formation and of maximization of interests' commonalities between users within each social* group). The experimental results indicate that spectral clustering, due to the optimization it offers in terms of normalized cut minimization, is applicable within the context of Magnet Beyond socialization services. *Regarding profile's cardinality impact on the system performance, this is shown to be highly* dependent on the underlying distribution that characterizes the frequency of user preferences appearance. Our work also incorporates the introduction of a heuristic algorithm that assigns new users that join the service into appropriate social groups, once the service has been initialized and the groups have been assessed using spectral clustering. The results clearly show that our approach is able to adhere to the service requirements as new users join the system, without the need of an iterative spectral clustering application that is computationally demanding.

1. Introduction

The recent proliferation of applications that support socialization through the internet such as newsgroups, virtual chat rooms, instant messaging programs and, lately, weblogs, has given rise to a new form of networking concept, called social networking. Socialization has become a driving force of networking applications design and development; many services are now deployed in the scope of bringing together (in networking terms) people with common interests. Explicit expression of interest through customized user profiles [1] and/or implicit inference of user preferences through monitoring of their web activities [2] are used to support the social networking framework. Within such a context, one of the most challenging tasks is the assessment of the exhibited social groups, and the design of appropriate frameworks that exploit the commonalities between users' interests to create virtual communities that promote efficient socialization. However, despite the abundance of virtual meeting points and the relevant services, in the virtual communities, with little sense of the presence of other people, individuals have a difficult time forming cooperative relationships [3]. Finding the right person to contact is still a

* Corresponding author. Tel.: +30 210 772 1513; fax: +30 210 772 2530.
E-mail addresses: pkaramol@telecom.ntua.gr (P.N. Karamolegkos), bpatr@telecom.ntua.gr (C.Z. Patrikakis), ndoulam@cs.ntua.gr (N.D. Doulamis), panvlah@telecom.ntua.gr (P.T. Vlacheas), gnikolakopoulos@telecom.ntua.gr (I.G. Nikolakopoulos).

0898-1221/$ – see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.camwa.2009.05.023

trial and error procedure which is mainly based on intuition and personal experiences [4]. Such efficiency shortage of several methods that are used to promote social networking is complemented by a relevant lack of evaluation studies regarding the performance assessment of algorithms that provide the framework for networked socialization.

In this scope, this paper presents and evaluates an integrated theoretical framework for a social networking service, as this has emerged out of the definition of pilot services of the IST MAGNET Beyond project [5]. MAGNET project and its successor MAGNET Beyond focus on the development of user-centric business model concepts for secure Personal Networks (PNs) in multi-network, multi-device, and multi-user environments, and the design/development of novel personal network applications and services. In the scope of this work, we propose and evaluate the framework of a user profile – based socialization service whose purpose is the automated formation of groups of interests between MAGNET users.

In brief, the service is initialized by an initial pool of users that is registered to a social matching server which collects all the available profiles. At system initialization, the users are partitioned into groups in a way that the following two preconditions are held: (a) all users within each group get as many contact points as possible, which translates to the requirement for the creation of equal sized groups, and (b) the degree of interests commonality within each group is maximized (i.e. the users of each distinct group must have as many interests in common as possible). Then, at each new user's arrival, a proposed algorithm places him at a group in a way such that requirements (a) and (b) are adhered to, during the dynamic process of user joins.

The service introduced herein, named *Icebreaker*, has been selected for implementation during pilot applications deployment, and will be part of a larger MAGNET Beyond testbed. *Icebreaker*'s application scenario incorporates many aspects related to secure Personal Networking; this work focuses on the design and evaluation of mechanisms that pertain to social networking aspects and, namely, it addresses the issues of (i) the clustering algorithm that will be used for partitioning users into social groups (ii) modeling concepts of the user profile structure (iii) a heuristic algorithm that will determine the dynamic system behavior (placement of new users into appropriately selected groups so that the system requirements are adhered to) and (iv) a study of the impact of profile size in relation to the overall system performance.

In terms of user profiling, we focus on modeling aspects (profile cardinality, probability distribution of user-representative keywords), and not on lexical issues (e.g. resolving the presence of more than one synonyms, etc.); such issues have been topics for other areas of research (e.g. stemming algorithms [6,7]) which are also under consideration for future versions of our proposed service. We model the user profile as an unordered set, comprising n distinct keywords that represent user preferences. In the scope of studying the impact of profile cardinality ($|u|$, the set of keywords a profile comprises) we vary $|u|$ and examine the consequences on the system behavior both in static (initial user partitioning) and dynamic (new users' joining the service) behavior. For the clustering quality assessment, we rely on theoretical foundations of spectral clustering, which is our algorithmic framework for the system's initial user partitioning and on the evaluation of certain metrics (Average System Semantic Proximity and standard deviation of users' partitioning distribution) that indicate the performance of the examined clustering method in regard to our service requirements.

We introduce the concepts of *semantic proximity* (p_s), *semantic distance* (d_s) and *semantic centroid* (μ_s) that will be used as inputs to the clustering algorithms which will be evaluated in the scope of static groups' assessment. It is to be noted that the term *semantic proximity* has also been used in other work [8], indicating the degree of common peer users' interests in terms of files. We use the same term herein, so as to indicate the degree of conceptual closeness between two users, on the bases of their interests, as these are expressed in their profiles. We redefine it in the scope of social networking as $p_s^{ij} = a_{ij}/|u|$ (i.e. the ratio of two profiles u_i, u_j common keywords (a_{ij}) to the cardinality of the profile set ($|u|$)). It is evident that semantic proximity takes values between 0 (no common keyword) and 1 (identical profiles). We give the definition of its counterpart metric, *semantic distance* (d_s^{ij}) which is given by $d_s^{ij} = 1 - p_s^{ij}$. As we show in Section 3.2, the latter quantity satisfies all the prerequisites of a metric and varies between 0 (identical profiles) and 1 (maximum distance between two profiles, when they don't share any common keywords). These two metrics are used to transform the profile sets into the relevant metric (distance) and similarity spaces, in which two specific relational clustering algorithms operate: hierarchical and spectral clustering. We also introduce the concept of *semantic centroid* (μ_s) which will help us, on one hand, to adapt k-means clustering in the idiosyncrasies of our semantic framework and, on the other hand, to propose a heuristic algorithm for the dynamic update of clusters during the arrival of new users to the service.

Having expressed the profiles relations into well defined relational (distance/similarity) spaces, the next step of our study is to assess the performance (in terms of adherence to our service requirements) of three very popular clustering algorithms (i.e. *k-means*, *hierarchical clustering* and *spectral clustering*). We use the metric of *Average System Semantic Proximity* ($\overline{P_s}$) which expresses the semantic proximity (p_s) averaged over the members of each cluster and over all existing clusters of the network. In such a context, we substantiate, through experimental results, the applicability of spectral clustering in terms of conformance with our requirements, and we provide a brief explanation of these results using theoretical foundations of the specific algorithm.

The last contribution of this work is the introduction of a heuristic algorithm that performs a very quick and efficient assignment of new users that join the service, without having to go through re-initialization of the system, which would be inefficient, since (i) it would require eigen-decomposition of large matrices (as will be briefly explained in Section 4.3, spectral clustering relies on such an eigen-decomposition) (ii) it would probably create a large scale group reorganization since the outcome of spectral clustering is not deterministic (having users constantly reallocated into different groups would be unacceptable). Our proposed algorithm manages to maintain a dynamic update process that adheres to our service

requirements, and its performance depends largely on the distribution of user's preferences, on the initial user pool on which the system was initialized, and on the cardinality of the profile used to describe users' interests.

This paper is structured as follows: Section 2 briefly describes the application scenario of Icebreaker and cites several relevant works in regard to user profile modeling and social groups identification on the basis of clustering methods; Section 3 provides the overall theoretical framework that will be used for the evaluation of our social service, while Section 4 describes the clustering algorithms that will be evaluated in our work. Section 5 introduces our heuristic algorithms, in regard to the dynamic behavior of the system; Section 6 describes the experimental setup and the relevant results; Section 7 concludes the paper by presenting and overview of our work and directions towards upcoming extensions in the scope of MAGNET Beyond social networking services.

2. Related work—Application scenario

One of *Icebreaker's* main goals is the creation of social groups between MAGNET users. The application scenario (regarding the socializing aspect of the service) incorporates a social-matching server, where the users register at the beginning of the session, by submitting to the server their personal information through their profiles; based on these profiles (more details on profile modeling are given in Section 3.1) the server partitions the users into groups, so as to (a) keep the size of the groups equal: assuming an initial number of N users which we want to partition into R groups, the purpose is to create groups whose size is as close to the value N/R as possible and (b) create groups with users sharing as many common interests as possible. Based on this initial users' decomposition into social groups, the system may subsequently accommodate in an effective way new user arrivals (the term effective implying both adherence to our service requirements and low algorithmic complexity).

The requirement for creation of equally-sized groups is based on several criteria that pertain to the specific instantiation of the socialization service. In *Icebreaker*, we intend to incorporate a socialization framework that will suit the idiosyncrasies of the overall application. The service will be deployed in an ad-hoc manner, i.e. on specific occasions, such as business meetings, conferences, workshops, exhibitions; such events are usually attended by people that need to interact with each other, on the basis of common interests. Therefore, we keep in mind that we are addressing short-term user interactions, on a limited physical space. These aspects create some deviations from the social networking concept found in cyberspace, either in the case of p2p paradigm or/and social networking applications per. se (e.g. Facebook). In such a view, it becomes evident that efficient socialization incorporates both the maximization of user interests and the avoidance of great imbalances in group sizes distribution. In a 4–5 hour event, we want to avoid a distribution comprising several groups of 5–10 people (or event the creation of single person groups, i.e. isolated users), and a few others made up of several hundreds of persons.

Furthermore, since *Icebreaker* is intended to be deployed in conference-like kind of events, there is going to be a need for preliminary resources (i.e. facilities) provisioning, for the accommodation of the social groups yielded by the application. Since there will not be any a priori user interests assessment and in the view of provisioning for a viable business scenario, it is more efficient and feasible to provision equal resources for each group.

Requirement *b* is more or less self-evident, since it is easily understood that the users within a group must share as many common interests as possible. In our context, and as will be showcased in Section 6, the notion of user interests maximization is decoupled from the groups' size, by averaging (normalizing) the interests commonality.

Several works have been proposed in the context of user communities' assessment, both in fixed and mobile networks. Seitz et al., [9] present a mechanism that combines mobile clustering and data clustering so as to create groups of mobile users. Information about the users is based in profiles, and the group formation relies on exhibited profile similarities, an approach that will also be undertaken in the work presented in this paper. However, there is an explicit need for an a priori definition of group characteristics and rules that a mobile node must comply with, so as to become member of a group. In our approach, the usage of spectral clustering allows for an automatic recognition of group patterns, without any need for previous definition of group characteristics. Furthermore, we avoid the need for neighbor discovery and spanning tree formation and maintenance, which are essential parts of the aforementioned algorithm. Phatak et al., (2002) [2] address simultaneously both the issues of user and URL clustering. Their approach relies on the discovery of patterns in the web access habits of users so as to create a recommendation mechanism in mobile web clients. Various distance metrics used in clustering are evaluated, using a framework that relies mainly on a metric of entropy that depends only on the original data, i.e. the entries of the incidence matrix that the clustering is based on. The specific work presents several commonalities with our proposed framework: definition of metrics necessary to the clustering process, evaluation of the clustering process, incorporation of relation clustering. However we extend the authors' approach by running a larger scale comparative study about how popular clustering algorithms in communities' assessment behave, under various conditions.

The topics of community structure and community evolution have been in the focus of several other works. In [10], the authors investigate the time dependence of overlapping communities on a large scale, and thus uncover basic relationships characterizing community evolution. They use as case studies, networks that capture the collaboration between scientists and the calls between mobile phone users. The work presented in [11] addresses the issues of detecting and characterizing this community structures exhibited in a variety of social and biological networks. The work deals with the concept of modularity, which is defined as the number of edges (as up to a multiplicative constant) falling within groups minus the expected number in an equivalent network with edges placed at random. Other approaches that target the efficient discovery of communities within several types of networks are presented in [12,13].

A very interesting study on the effect of user profile size in respect to the quality of user communities' organization is performed by Lancieri et al. (2003) [14]. A number of virtual users is characterized based on a set of pre-defined keywords, and are subsequently partitioned in several clusters through an hierarchical agglomerative clustering technique (HAC). The outcome of each clustering process (in respect to a profile size) is then compared with a reference clustering produced by human evaluation. Finally, in another work that is similar with the objectives of this paper, Sotiropoulos et al. (2006) [15] compare different clustering approaches for creating groups between users of common interests, with the purpose of creating a recommendation service. The profile vectors are assembled through the users' interaction with a video-store application. The testbed uses a profile pool of 150 users, and compares the performance of various clustering methods, but without examining the impact of profile size on the clustering quality, which is a central aspect of this work.

3. User profile-Based semantic framework

As was stated in the introduction, the proposed semantic framework for the assessment of user groups relies on three closely interconnected aspects. The first issue that needed to be addressed was the construction of a profile model that was mathematically consistent with the overall framework. Many authors [16,14] have addressed this issue by representing the user profile as a multi-dimensional vector, whose variables correspond to keywords that represent user's preferences. For reasons that will become more apparent in the following subsections, we avoid such a definition and we model the user profile as an *unordered set* of user-representative keywords (tabbed simply as *keywords* in the rest of this paper). The next important step was to perform an evaluation study of the most common clustering algorithms and in this scope we had to provide definitions regarding several metrics and notions used by these algorithms. For example, k-means clustering requires the definition of centroids, which are used at each iteration of the method so as to evaluate the objective function (sum of points—centroids distances within each cluster); spectral clustering and hierarchical clustering require the assessment of similarities and distances respectively between sets to be clustered. We provide definitions for all these notions and create a framework on which these three basic clustering methods are applied. Finally, the last piece of puzzle regarding our proposed framework relates to the definition of a heuristic set of rules that would define the dynamic system's behavior in terms of new user's arrivals that take place after system's initialization. Since new user's arrivals cannot be predicted, the purpose is to introduce a method that provides a compromise between effectiveness and low algorithmic complexity, so to allow the system to have a low response time in regard to new service joins.

3.1. Profile modeling

In order to enable personalization of services, it is imperative that a list of the attributes characterizing the user is made available to the service provider. Such a list is usually provided through standardized user profiles, which according to the definition of ETSI [17] is *the total set of user-related information, preferences, rules and settings, which affects the way in which a user experiences terminals, devices and services*.

Regarding profile extraction–composition, there have been many relevant implementations proposed and deployed. One common approach is the implicit profile construction, through the monitoring of user activities [1], e.g. visited websites [2,18], preferences in consumer goods [15] etc. In such a case, the service provides automated mechanisms that follow a user's habits and construct the relevant profile through semantic interpretation of his interaction with the application. Another methodology relies on the explicit construction of the profile, using textual descriptions that represent his preferences [19]. In the latter case, the main task regarding the creation of a more abstract representation of user preferences is delegated to methods that extract the actual semantic content of the text. Such methods (e.g. vector space model [20,21]) query the document and retrieve the important, i.e. representative words (keywords).

In our study, we do not address issues related to profile information extraction. Instead, we assume that we have a list of words that represent the user's preferences, ignoring the way this list has been acquired and the initial profile structure (i.e. whether it is a plain text or a tree structure composed of keywords). Our main concern focuses on two aspects of profile representation: (a) its cardinality, (i.e. the number of keywords that will be finally used to characterize a user) and (b) the probability distribution of the keywords.

Regarding the study of the profile cardinality, we evaluate its impact on the quality of the social grouping, assuming that each profile u is represented as an unordered set composed of n distinct keywords w_i, $i = 1, 2, \ldots, n$. For the distribution of the keywords frequency of appearances, we adopt the assumptions that the frequency of the keywords follows (i) Zipf's law, which intuitively translates to the fact that there are a few keywords appearing very often, while the majority of the available keywords is rarely used and (ii) Uniform distribution.

The first assumption lies on the basis of Zipf's law extensive usage for user preferences modeling in various relevant studies [22–24]; therefore it is safe to assume that profiles which are generated from monitoring user activities [25–27] also exhibit Zipfian distribution. The Zipf distribution is given by $P(k, a) = 1/[k^a \cdot \zeta(\alpha)]$ and it expresses the probability of appearance of the kth symbol (sorted[1] in order of descending frequency of appearance) of the vocabulary from which

[1] The term *sorted* in this case applies to the relevant frequency of keyword appearances and is not related with their position in the user profile, which is considered as an unordered set of words.

the profiles are assembled; $\zeta(\alpha)$ is the Riemann's zeta function and α the skewness parameter of the distribution. In the experiments we have assumed that without loss of generality that the profiles are constructed from a pool of 5000 keywords and we have also used $\alpha = 1.5$, based on other relevant studies that model distributions of user preferences [27]. In order, however, to test the system into an extreme disperse of users' preferences distribution and to indicate that the system behavior is largely dependent on such a distribution, we perform our experiments assuming also that the appearance of profile keywords are equiprobable, which is substantiated by the adoption of Uniform distribution for keywords frequency.

3.2. Social networking metrics

Having defined a conceptual model for the user profile, the next issue we tackle relates to the definition of an appropriate metric that will allow us to define relations between different profiles.

In terms of modeling the dissimilarity between user profiles, we choose not to adopt the Euclidian distance which had been extensively used for measuring the dissimilarity between textual documents, since we do not model the profile as a vector. Hence, it is imperative to devise a relevant metric that is applicable within our generic framework in the goal of providing the appropriate substratum for clustering methods to identify our social groups.

In the scope, therefore, of coming up with an appropriate distance metric, we introduce the concept of *semantic distance* between two user profiles u_i and u_j, which is defined as

$$d_s(u_i, u_j) = 1 - \frac{a_{ij}}{|u|} = 1 - p_s^{ij} \tag{1}$$

where α_{ij} indicates the number of common elements between profiles u_i and u_j, $|u|$ expresses the cardinality of the profiles (we assume that at each instance of our framework all profiles are of the same cardinality). This is conformant with the mathematical definition of a metric space since the latter is defined [28] as a tuple (M, s) where M is a set (in our case M is the profile which is a set of keywords) and s is a metric on M, i.e. a function $d_s : M \times M \to R$ such that

(i) $d_s(u_i, u_j) \geq 0$: it is evident from the definition that $d_s^{ij} \in [0, 1]$, with $d_s^{ij} = 0$ if $u_i = u_j$ and $d_s^{ij} = 1$ if the two profiles share no common keyword

(ii) $d_s(u_i, u_j) = d_s(u_j, u_i)$: easily derived from the definition of semantic proximity

(iii) $d_s(u_i, u_j) \leq d_s(u_i, u_k) + d_s(u_k, u_j)$: replacing $d_s(u_i, u_j)$ from Eq. (1) we get

$$\begin{aligned} d_s(u_i, u_j) \leq d_s(u_i, u_k) + d_s(u_k, u_j) &\Rightarrow 1 - \frac{a_{ij}}{|u|} \leq \left[1 - \frac{a_{ik}}{|u|}\right] + \left[1 - \frac{a_{kj}}{|u|}\right] \Rightarrow -a_{ij} \\ &\leq |u| - (a_{ik} + a_{kj}) \Rightarrow (a_{ik} + a_{kj}) - a_{ij} \leq |u| \,. \end{aligned}$$

The last inequality is always true: the left hand side is a sum of a positive $(a_{ik} + a_{kj})$ and a negative $(-a_{ij})$ term, therefore it takes its maximum value when $(a_{ik} + a_{kj})$ is maximized and $|a_{ij}|$ is minimized. The former case occurs when profiles u_i, u_j and u_k have all of their keywords in common, in which case however $a_{ik} = a_{kj} = a_{ij} = |u|$ and $(a_{ik} + a_{kj}) - a_{ij} = |u|$ which is the maximum value of the left hand side of the last inequality.

(iv) $d_s(u_i, u_j) = 0$ if and only if $u_i = u_j$, which goes by the profile definition we provided at the beginning of this section as an unordered set of keywords; it should be noted that had we defined the profile as a vector in a multidimensional space, the specific metric property would not be valid, since a vector defines a default order in the sequence of values.

In the same context of establishing a metric that indicates semantic distance between two profile sets, we introduce its counterpart metric, i.e. *semantic proximity* between two user profiles u_i and u_j which is defined as

$$p_{ij} = a_{ij} / |u| \tag{2}$$

i.e. the ratio of the common profile keywords to the profile cardinality. The definition of such a proximity measure is necessary for the application of spectral clustering that uses, as input, a similarity matrix $\mathbf{S} = [p_{ij}]$, where p_{ij} indicates the similarity between profiles u_i and u_j. Therefore, in the scope of providing a comparison between cluster methods that is as fair as possible, we opt for the definition of a similarity measure that is directly derived from the previously defined semantic distance. Intuitively, this stems from the definition of semantic distance itself; since the latter varies between 0 and 1 (1 being the maximum possible distance between two profiles) and is given by (1), it is evident that the term $a_{ij} / |u|$quantifies the normalized similarity between profiles u_i and u_j.

Finally the last notion we are going to need a definition for, relates to the centroid of a group, which is an essential aspect for k-means and several versions of hierarchical clustering. The common definition of cluster centroid which is used during the clustering process of multidimensional data is attributed to a point whose coordinates equal to the average of the respective group members coordinates. Since our profiles are composed of keywords, the averaging operation cannot be applied in our case; we therefore provide our framework with an alternative definition for a cluster centroid, or *group*[2]

[2] The terms *cluster* and *group* are used interchangeably in this paper.

representative as it will be called in the rest of this paper, based on the notion of semantic distance as was introduced in the previous paragraphs: for each group $k = 1, 2, \ldots, R$, we define as *group representative* μ_k the group member yielded by the following equation:

$$\mu_k = \arg\min_i \sum_{j \in k} d_s^{ij} \tag{3}$$

i.e, the group representative is the point whose sum of semantic distances from the rest of the group members is minimum. As will be evident from the subsequent chapters, such a definition allows us to (i) run an evaluation study for k-means clustering the context of our semantic framework and (ii) to execute a fast heuristic algorithm that places new users into already assessed groups without; the latter algorithm is introduced in a following subsection.

4. Social groups assessment

Various methods have been used in the scope of identifying user communities in networking environments; one of the most common techniques [29,30] is the application of clustering methods that are available in the mathematical literature (i.e. k-means, hierarchical clustering [31] etc) which is the approach that we also follow in this work. We have selected, for our evaluation study, three of the most commonly used clustering algorithms (i.e. k-means, spectral clustering and hierarchical clustering). These clustering algorithms have been selected for comparison on the basis of their widespread adoption in social networking frameworks. Several indicative examples of their applicability can be found in [32,12,33,34].

The following subsections provide a brief introduction to these three clustering techniques. For more details, the reader is referred to the relevant citations.

4.1. k-means clustering

K-means [31] is a simple unsupervised clustering algorithm that uses, as input, the number of clusters and a centroid (central point) for each of these clusters. The algorithm proceeds by iteratively assigning each data point to a centroid (and therefore a cluster), then re-calculating new centroids, and so on, until it converges. The final goal is the minimization of an objective function which expresses the sum of squared distances between the data points and the centroids. The equation to be minimized is given in (4).

$$J = \sum_{j=1}^{R} \sum_{i=1}^{N} \left\| x_i^{(j)} - \mu_j \right\| \tag{4}$$

where, R is the number of clusters, N the number of data points, $x_i^{(j)}$ the ith data point assigned to centroid μ_j and $\| \cdot \|$ represents a chosen distance measure (in our case, semantic distance).

4.2. Hierarchical clustering

Hierarchical clustering [31] is a simple technique that is divided into two major algorithmic subdivisions: divisive and agglomerative methods. The first class of algorithms proceeds by successively decomposing the ensemble of data points (which are initially considered as an integrated group) into smaller and smaller clusters, until each data point composes a stand-alone cluster. The agglomerative methods, which will be used in this work, elaborate the clustering process the other way around, starting by integrating the data points into successively larger clusters, until all of them are merged into one cluster. Agglomerative clustering techniques are further distinguished according to the metric that is used to identify the distances between data points and/or clusters, so as to make the relevant decisions at each step of the algorithm regarding the necessary clusters'/data points' merge. So, for example, when, in order to assess two cluster's distance, the minimum distance between their members is incorporated, the respective method is called single linkage hierarchical clustering; when the average distance between the data points of the two clusters is taken into account, we get the average linkage hierarchical clustering, and so on. In the scope of this work, the distance metric incorporated in our versions of hierarchical clustering is, once again, the semantic distance.

4.3. Spectral clustering

Spectral clustering [35] is a more recently introduced unsupervised learning technique that uses the symmetric similarity matrix of data points and the number of clusters as input to the algorithm. From the vector representation of each data point, a similarity matrix $\mathbf{S} = [s_{ij}]$ is constructed, where s_{ij} indicates the similarity between profiles u_i and u_j, as this was introduced in Section 3.2.

An intuitive way of considering spectral clustering is the graph theoretic view; in such a view, data points to be clustered are represented as vertices of a graph and their connections as weighted edges, where the weights correspond to the degree of similarities between the respective vertices. Spectral clustering optimizes the value of the R-way normalized cut; in brief,

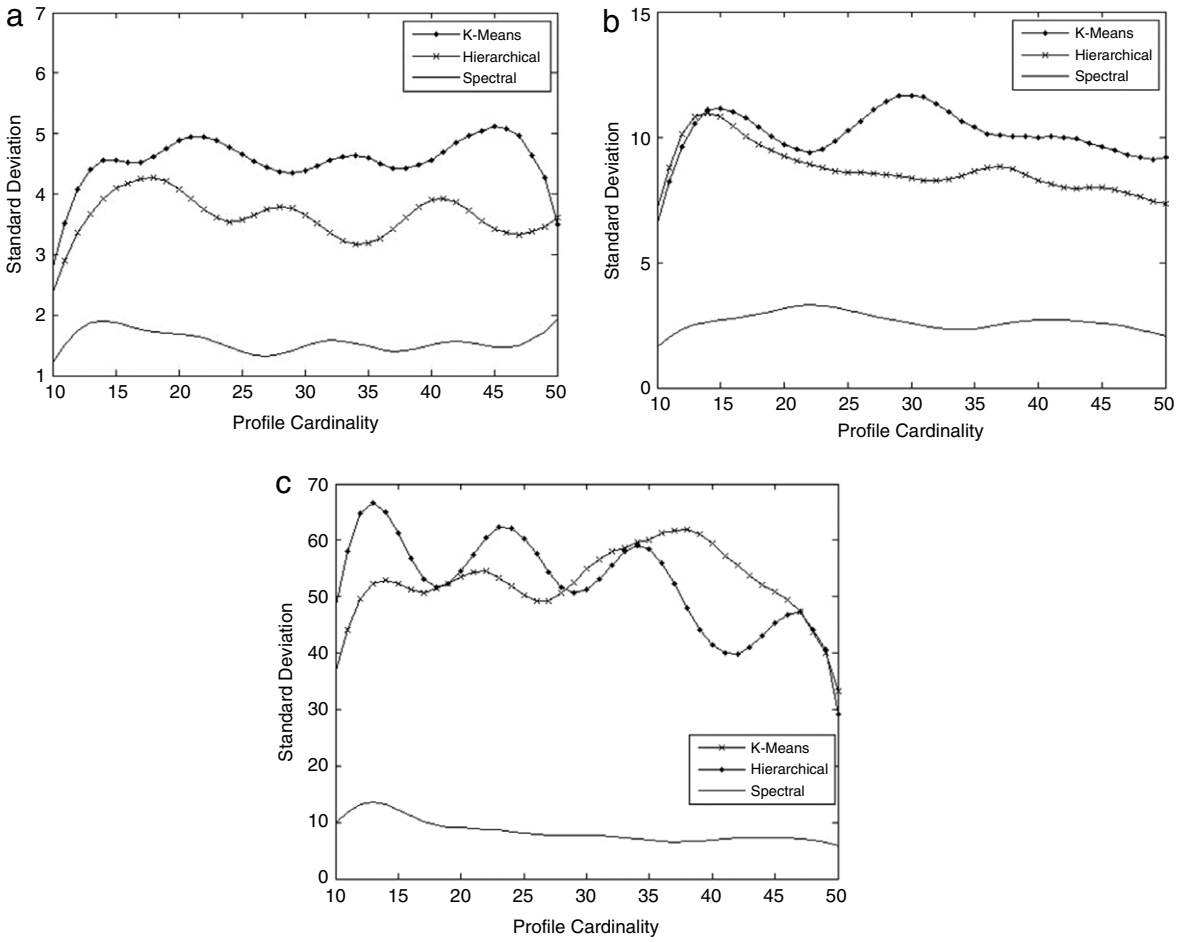

Fig. 1. Standard deviation versus profile cardinality of 10-cluster distributions resulting from application of the three clustering algorithms for Zipfian users' preferences distribution and 50 (a), 100 (b) and 500 (c) users.

for each cluster, the normalized cut equals the ratio between the sum of the edge weights that escape the cluster to the sum of the total weights of the cluster's vertices (the weight of a vertex is the sum of the weights of the edges incident to the specific vertex). The R-way normalized cut is given by the following equation

$$C_R = \sum_{r=1}^{R} \frac{\sum_{i \in A_r, j \notin A_r} s_{ij}}{\sum_{i \in A_r, j \in V} s_{ij}} \tag{5}$$

where A_r represents the rth cluster, V the total of graph's vertices and s_{ij} the similarity metric between vertex i and j. As will be shown in the experimental results, the minimization of (5) by spectral clustering, makes the specific algorithm applicable in the context of the MAGNET Beyond socialware, since it allows for the creation of social groups of similar size, which is one of Icebreaker's main scopes.

Since the emergence of spectral clustering, many algorithmic variations have been proposed [36–39]; the one adopted herein, as a proposal for the assessment of user communities, is the one presented in by Shortreed et al., 2005 [38]; reasons for the selection of the specific version of spectral clustering mainly relate to both its proven efficiency and its ease of deployment. In brief, the steps that will be used to classify our user profiles into groups are summarized below:

- From $\mathbf{S}$, we compute matrix

$$\mathbf{P} = \mathbf{Z}^{-1} \cdot \mathbf{S} \tag{6}$$

  where $\mathbf{Z} = \text{diag}(z_i)$, $z_i = \sum_{j=1}^{N} s_{ij}$, N being the total number of users.
- Subsequently, we compute the first R eigenvectors $v^1, v^2, \ldots, v^R$ of $\mathbf{P}$, where R is the number of clusters.
- Finally, the rows of table $\mathbf{V} = [v^1, v^2, \ldots, v^R]$ are clustered as points in R-dimensional space, using fuzzy c-means clustering.

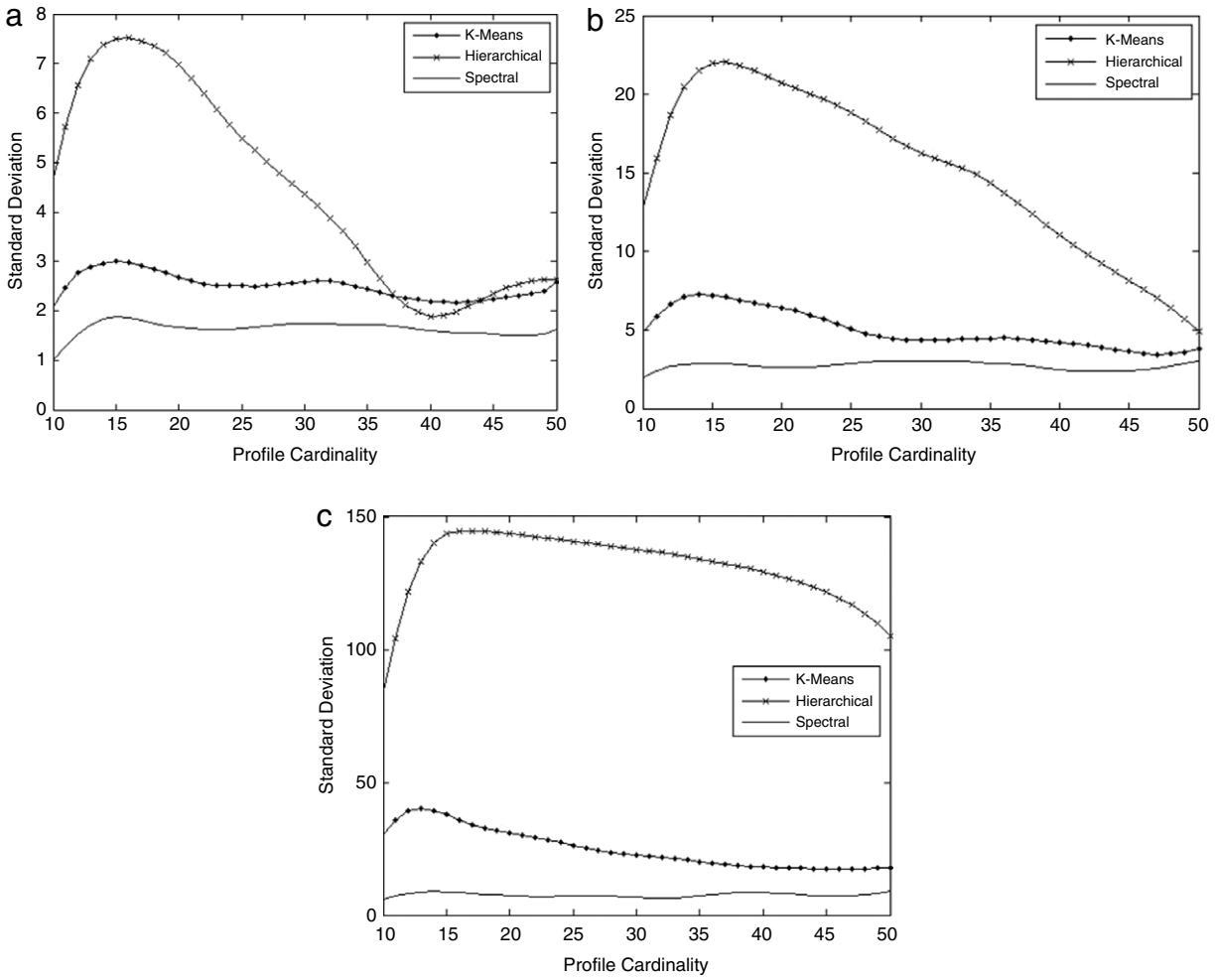

Fig. 2. Standard deviation versus profile cardinality of 10-cluster distributions resulting from application of the three clustering algorithms for Uniform users' preferences distribution and 50 (a), 100 (b) and 500 (c) users.

5. New user arrivals/user departures

After initializing the system through the application of a clustering method, the next step is to provision for the accommodation of new clients that dynamically join the system. One solution is to repetitively run the selected clustering method each time a new user arrives. Since our method of selection (as will be justified in the relevant section) is spectral clustering, such an approach requires the eigendecomposition of matrices of order N (the number of service clients) and it has the advantage that the system would always yield the optimal clustering. The term optimal, in this case, refers to the minimization of (5), which as is proven in the experimental results accommodates also for Icebreaker's service requirements. However, in the context of our service, there is a significant reason that renders such a policy inefficient. It is known that matrix eigendecomposition is a process of high computational complexity, namely of $O(n^3)$. In our case, the fast placement of a new user joining the system is of significantly higher importance than the preservation of the optimal normalized cut value within our group formation. This is much more evident in a crowded event (for which *Icebreaker* is destined to be deployed) with a lot of new service users arriving and creating a highly demanding computational process of decomposing large matrices into their eigenvalues and eigenvectors. Furthermore, another reason for avoiding a recursive eigendecomposition approach relates to the fact that the outcome of the spectral clustering algorithm is not deterministic; in this context, iterating spectral clustering each time a new user arrives would probably result in users' – groups' reallocations, which would make our approach inefficient.

Therefore, instead of adopting the aforementioned solution, we propose a heuristic approach of reduced computational complexity that (as proven in the experimental results) is able to efficiently track the system's progression by providing fast and effective user assignments into groups with the cost of slight deviation from the optimal normalized cut value (which, as was stated, is of secondary importance in our case). Moreover, it places each new user in the most appropriate group without the need of group re-organization, while it keeps both the degree of interests' commonality high within each group and the size of the groups close to N/R.

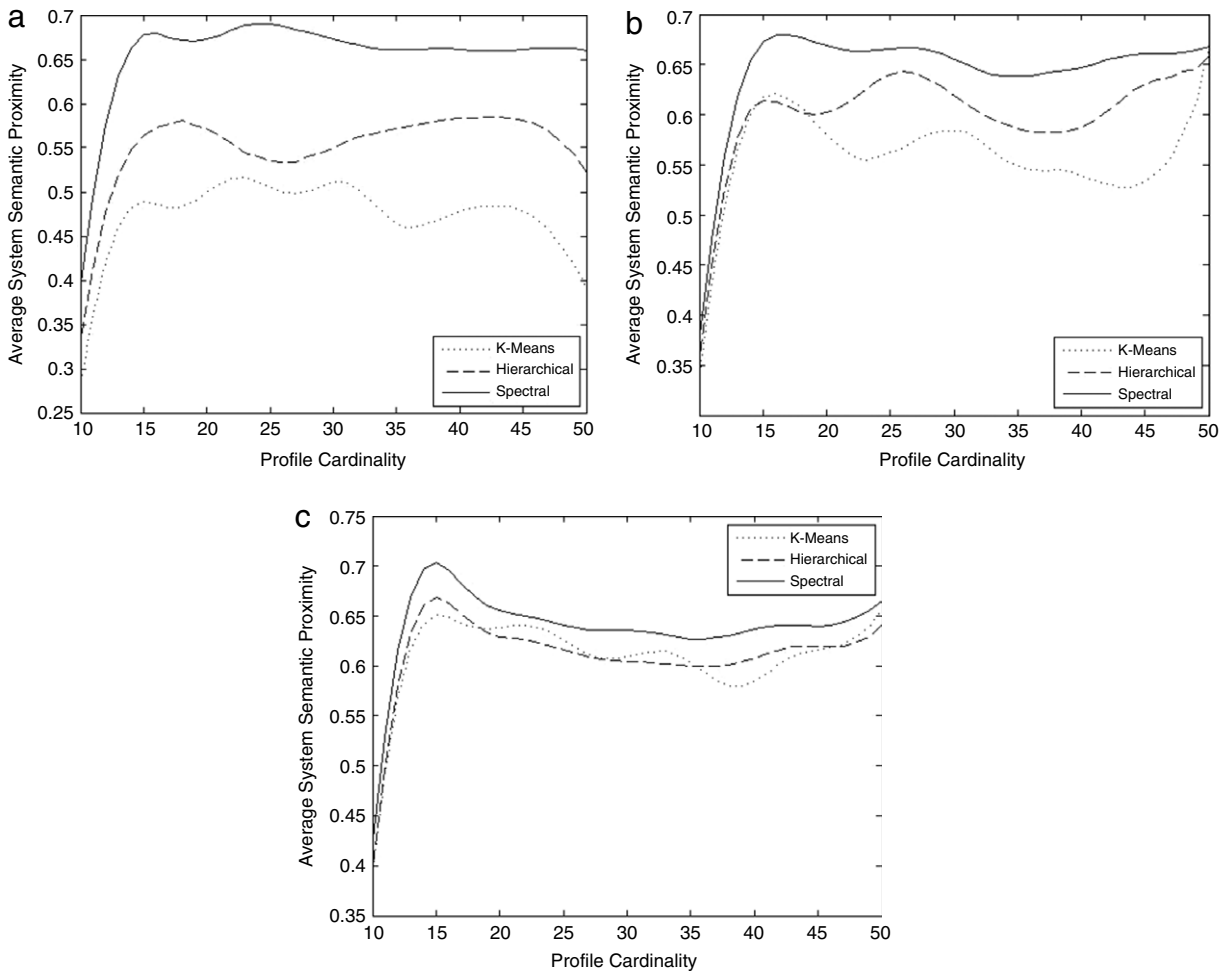

Fig. 3. Average Semantic System Proximity versus profile cardinality of 10-cluster distributions resulting from application of the three clustering algorithms for Zipfian users' preferences distribution and 50 (a), 100(b) and 500 (c) users.

5.1. User groups assignment

Assuming the system has been initialized by decomposing the users into groups through application of spectral clustering, the first step of the dynamic placement algorithm relies on the identification of the group representatives, which are extracted after the assessment of the initial groups; we use the definition of group representatives that was provided in Section 3.2. This process creates a set of representative profiles $\mu_i, i = 1, 2, \ldots, R$, for each group $A_r, r = 1, 2, \ldots, R$. When a new user joins the *Icebreaker* service, two lists of groups are selected:

(a) the list of groups G_1, whose representatives have the minimum semantic distance (see Eq. (1)) from the new user's profile u, i.e. $G_1 = \arg\min_{i=1,2,\ldots,R} d_{\mu_i u}$; in the case more than one groups share a common minimum semantic distance, they are all placed in group G_1.
(b) the list of groups G_2 whose cardinality is closer to the average group cardinality, i.e. $G_2 = \arg\min_{i=1,2,\ldots,R} \left|\overline{A} - A_i\right|$.

The group to which the new user will be assigned is selected randomly from $G_1 \cap G_2$ or from $G_1 \cup G_2$, if the former set is empty; each member of the intersection (or the union) has equal probability of being selected. The randomness is introduced since we do not have any a priori reason for biasing our selection over on group or the other. Table 1 summarizes our proposed algorithm.

5.2. Cluster recomposition/arrivals & departures

In order to address the issue of the changes inflicted in the cluster compositions by users' arrivals and departures into/from the service, we provision for a cluster merging/splitting algorithm whose purpose is to safeguard the semantic cohesion of the groups.

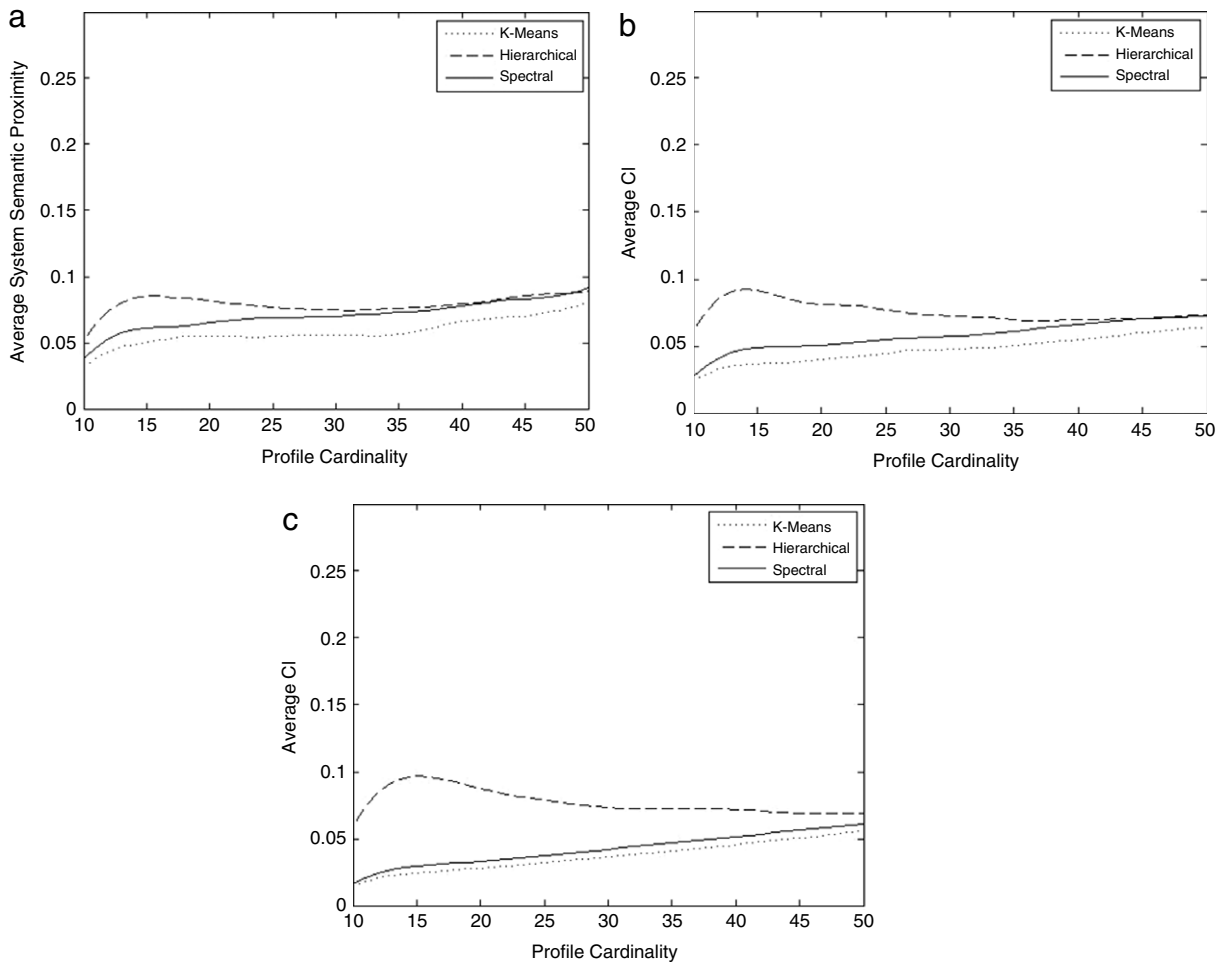

Fig. 4. Average System Semantic Proximity versus profile cardinality of 10-cluster distributions resulting from application of the three clustering algorithms for Uniform users' preferences distribution and 50 (a), 100 (b) and 500 (c) users.

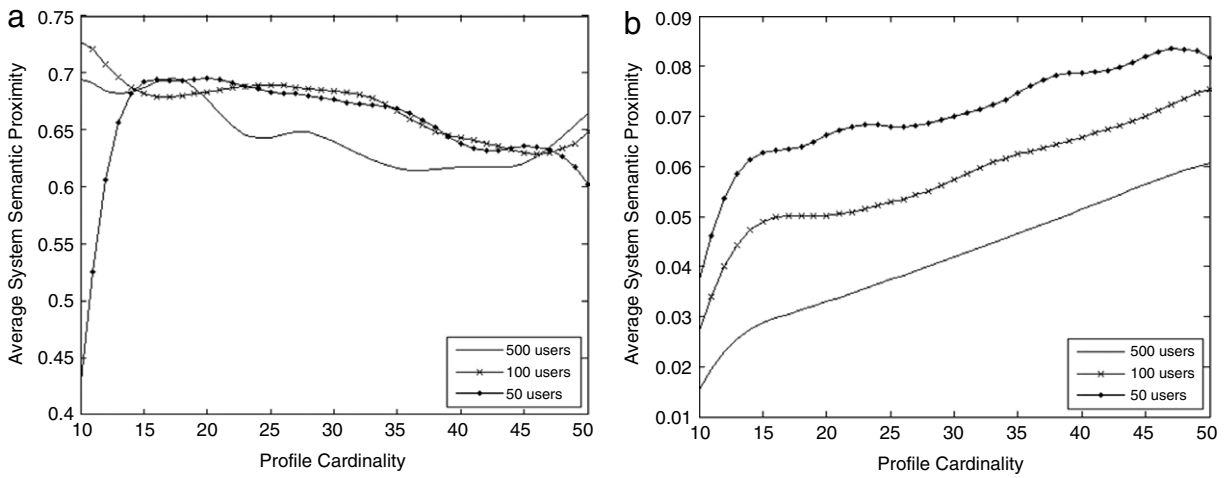

Fig. 5. Average System Semantic Proximity for Zipfian (a) and Uniform (b) users' distributions versus profile cardinality, after the system's initialization through the application of spectral clustering.

Upon an arrival and/or departure, the deviation of the filesets within each cluster is calculated. If this deviation is greater than a threshold, meaning that the content of the cluster's peers starts to become incoherent, a bi-partitioning (splitting)

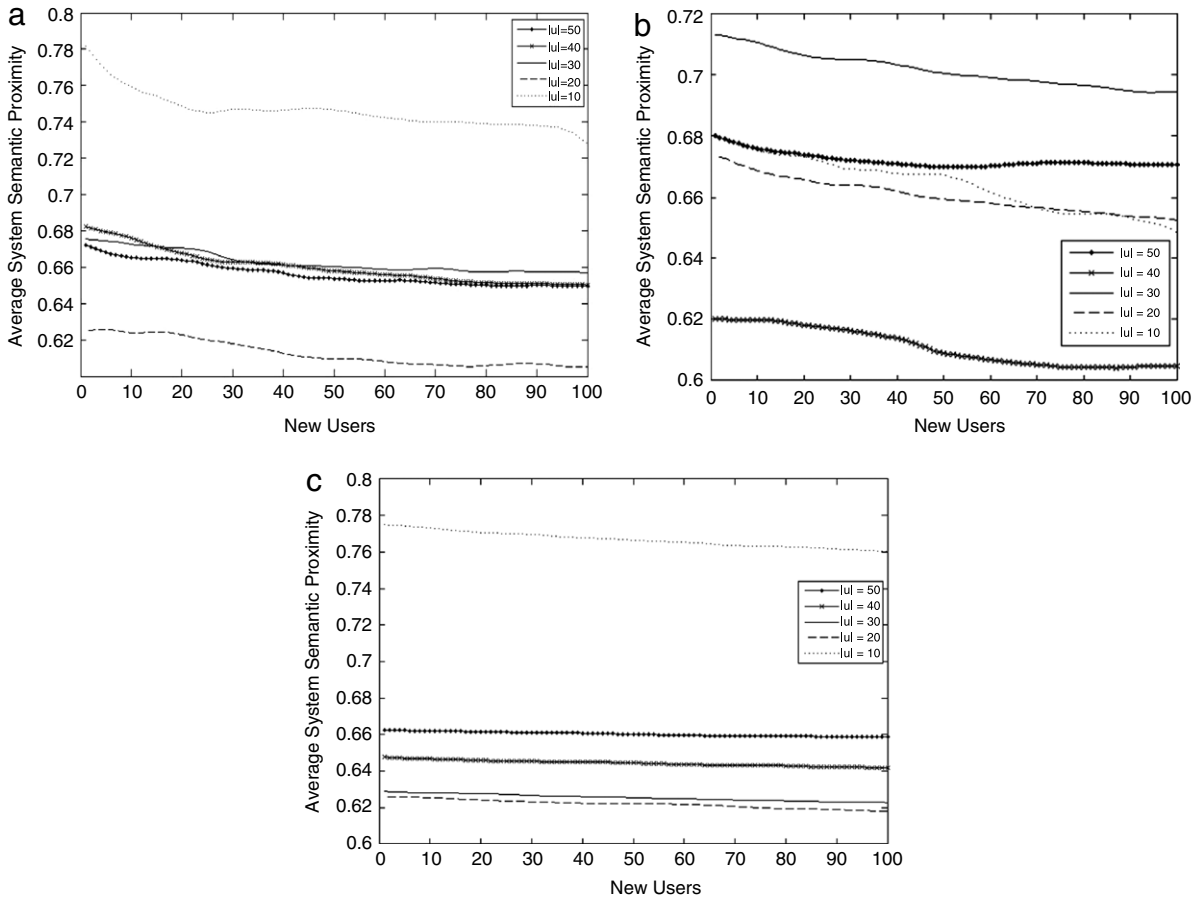

Fig. 6. Average System Semantic Proximity of our proposed heuristic for a system that is initialized by partitioning 50 (a) 100 (b) and 500 (c) users into 10 clusters, assuming Zipfian distribution of users' interests.

Table 1
Our user-group assignment heuristic.

System Initialization
- Initialization of the system and assessment of R social groups using spectral clustering
- Identification of group representatives μ_i, $i = 1, 2, \ldots, R$ using Eq. (3)

Dynamic User Placements
When new user (with profile u) arrives:
1. create set of groups G_1, : $G_1 = \arg\min_{i=1,2,\ldots,R} d_{\mu_i u}$, using Eq. (1)
2. create set of groups G_2, : $G_2 = \arg\min_{i=1,2,\ldots,R} \left|A - A_i\right|$ where $|A_i|$ is the group's i cardinality
3.a. if $G_1 \cap G_2 \neq \emptyset$ select (randomly) the group j to assign the user from $G_1 \cap G_2$
3.b. else, select j (randomly) from $G_1 \cup G_2$
4. recalculate μ_j

process is initiated. During the split, the static optimal (spectral) clustering algorithm is used. The process yields two new clusters.

To avoid continuous cluster splitting, a merge process is also activated. More especially, the dynamic patterns of users leaving/joining Icebreaker make probable that progressively several clusters start to become semantically close. In such a context, a periodic comparison of the profiles of groups' representatives is performed. If these profiles are found to have a (semantic) distance which is below a specific threshold, the two clusters merge. These two processes (group merging/splitting) act complementarily so that (a) the initial semantic consistency of the groups is maintained and (b) we avoid over-partitioning users into multiple groups.

5.3. Algorithmic complexity

The algorithm is of much lower computational complexity than the matrix eigendecomposition process incorporated in spectral clustering. More specifically, our algorithm has a complexity of $O(R)$, R being the number of social groups of the

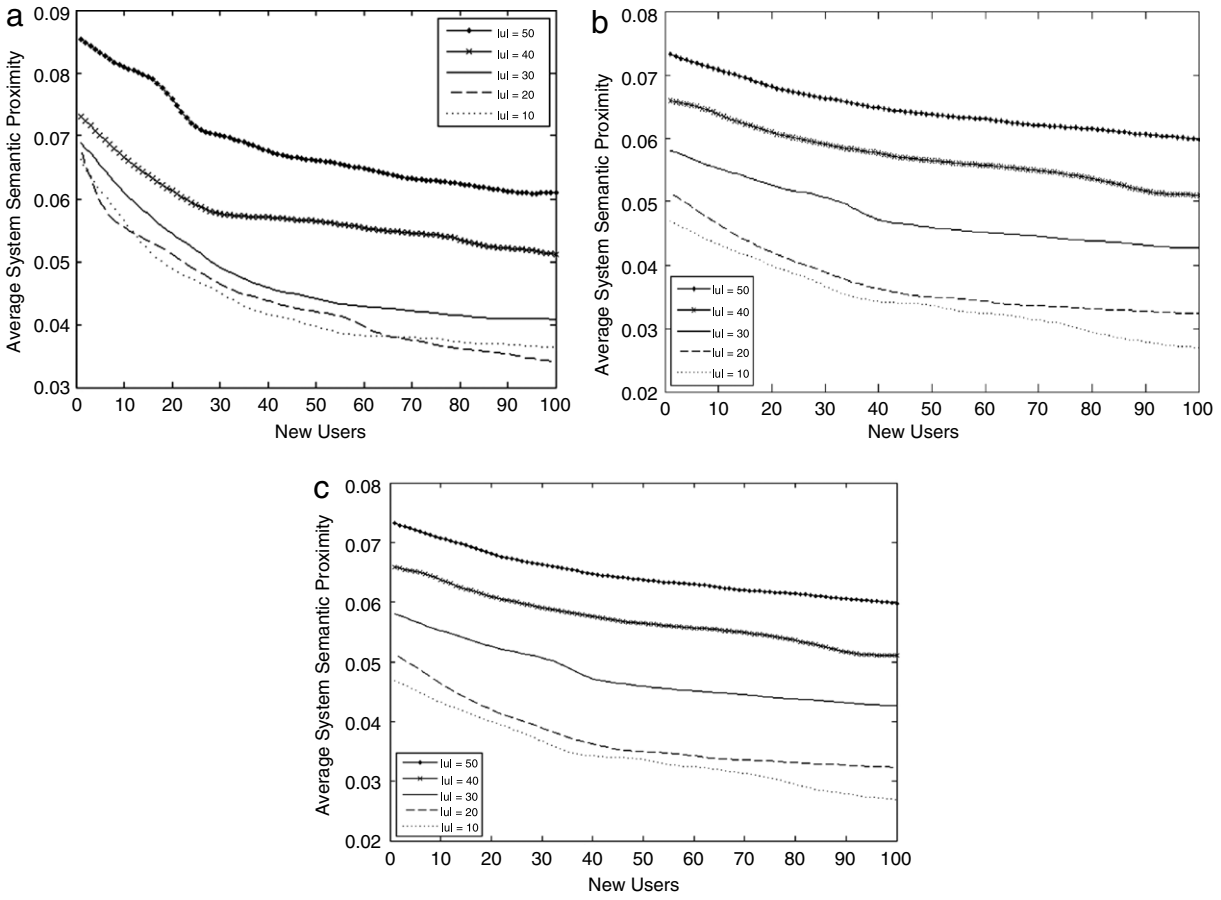

Fig. 7. Average System Semantic Proximity of our proposed heuristic for a system that is initialized by partitioning 50 (a) 100 (b) and 500 (c) users into 10 clusters, assuming Uniform distribution of users' interests.

system while the eigendecomposition is of $O(n^3)$ order, n being the number of Icebreaker's users. Furthermore, it is proven to efficiently track the system progression and user joins, without the need to find eigenvector of successively larger order matrices.

6. Experimental results

6.1. Experimental setup

We consider an infrastructure-based ad hoc network, where a clustering server acquires each user's profile, and provides a decomposition of the population into appropriate social groups, so as to (a) create as consistent groups as possible (the users within a groups should present the highest possible profile similarities) (b) create equal-sized groups. As was also stated in Section 1, the second requirement stems from the fact that in *Icebreaker* service we target an unstructured approach in terms of communities' formation (i.e. there will be no predefined topics of interest around which the users should be clustered). Therefore we have no a priori reason for biasing one group over the other, since the main purpose of the application is the socialization of users, rather than the identification of specific topics of interest, around which the clusters (groups) should be formed. The user's decomposition into social groups consists of two phases: (i) an initial assessment of social groups with the service users that are present at the system's initialization (ii) an on-line algorithm that places new users at the clusters as quickly and efficiently as possible without disturbing the existing groups (merging/splitting groups whenever necessary).

We have used synthetic profiles, with varying cardinality, which varies in the experiments from 10 to 50 keywords; we have assumed a vocabulary of 5000 totally available words, whose frequency distribution follows either Zipf's law with skewness parameter $\alpha = 1.5$ or Uniform distribution (we run the experiments under both assumptions). The concepts of distance, similarity and centroid that are incorporated in the relevant clustering algorithms have been replaced by the respective definitions introduced in Section 3.2.

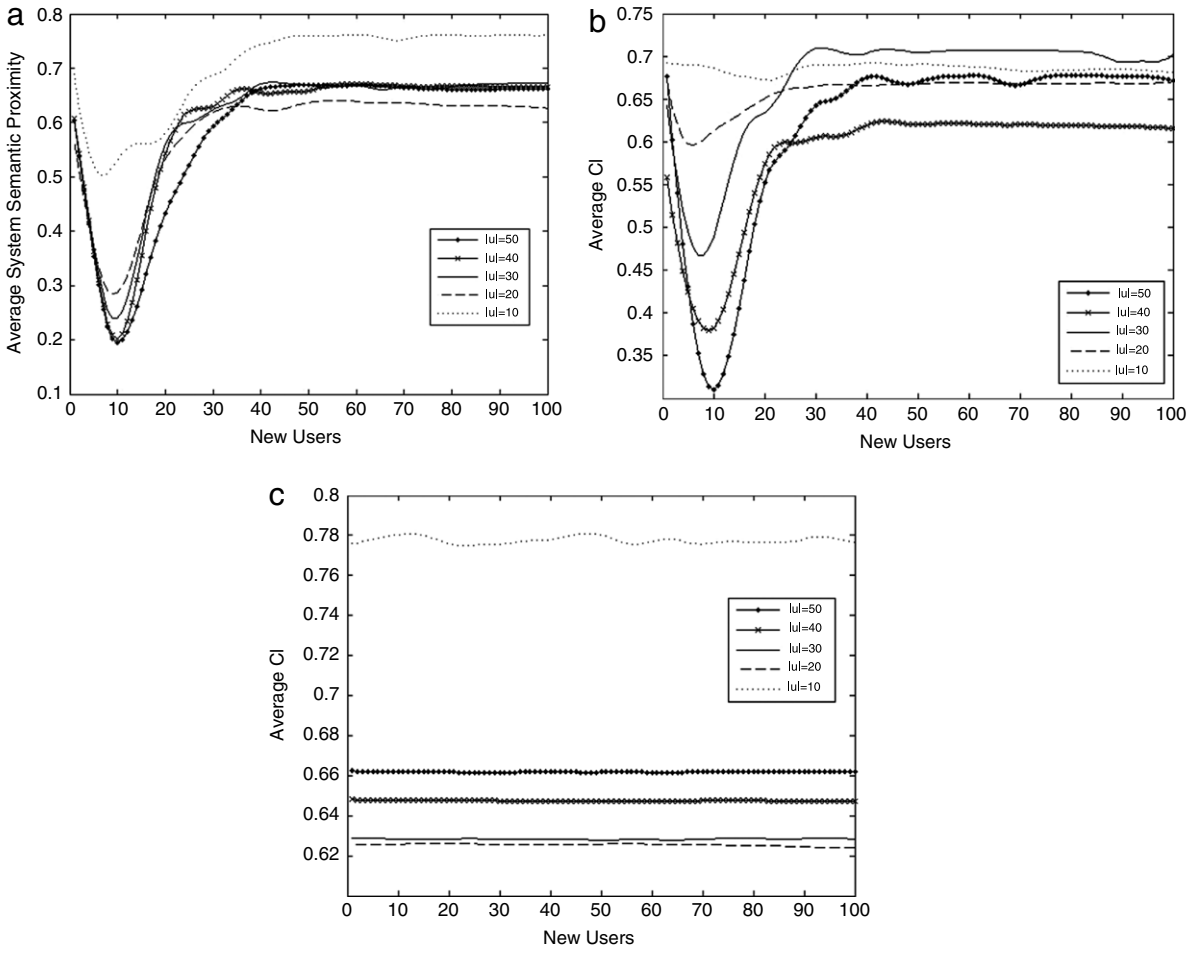

Fig. 8. Average System Semantic Proximity that is yielded after iterative spectral clustering application at each new user's arrival, for a system that is initialized by partitioning 50 (a) 100 (b) and 500 (c) users into 10 clusters, assuming Zipfian distribution of users' interests.

We should note here that there will not be any dictionary-based restriction on the users, i.e. there will not be a specific pool of keywords from which the service clients should compose their profile. The 5000 words restriction is just an assumption made for the sake of experimentation. Furthermore, as was also stated in the previous section, the present study does not address issues such as word disambiguation, presence of one or more synonyms/hyponyms etc. within the profiles. Such aspects have been the topics of other extended studies [6,7] whose results we intend to incorporate in the final instantiation of our framework. However, several studies [40,41] indicate that a span of 5000 words, with each one of them standing for a distinct concept (e.g. after appropriate stemming techniques have been applied), constitute an adequate synthetic dataset.

Assessing the performance of a social networking service is clearly not a trivial task. Human interactions with the system and the socialization aspect of such an application render its evaluation inherently subjective. In the scope, however, of providing a metric that is as objective as possible, we introduce the *Average System Semantic Proximity* ($\overline{P_S}$) metric, given by the following equation:

$$\overline{P_S} = \frac{1}{R}\sum_{k=1}^{R}\frac{1}{|A_k|}\sum_{i,j\in A_k}\frac{a_ij}{|u|} = \frac{1}{R}\sum_{k=1}^{R}\frac{1}{|A_k|}\sum_{i,j\in A_k}p_s^{ij} \tag{7}$$

where R is the number of clusters, a_{ij} the number of common elements between profiles i and j, $|u|$ the profile cardinality and $|A_k|$ the cardinality of the kth group. $\overline{P_S}$ provides an (objective) indication about the average number of common elements shared by each service user, averaged over all clusters. In order to use this measure also in the scope of evaluating the impact of profile cardinality variations, we choose to use a normalized metric (the number of common elements is divided by the profile cardinality); otherwise an increase of $|u|$ would also yield a respective increase of $\overline{P_S}$. Such a metric is indicative of the overall system's performance and is not subject to user-dependent perceptions about the service; we will use it to evaluate

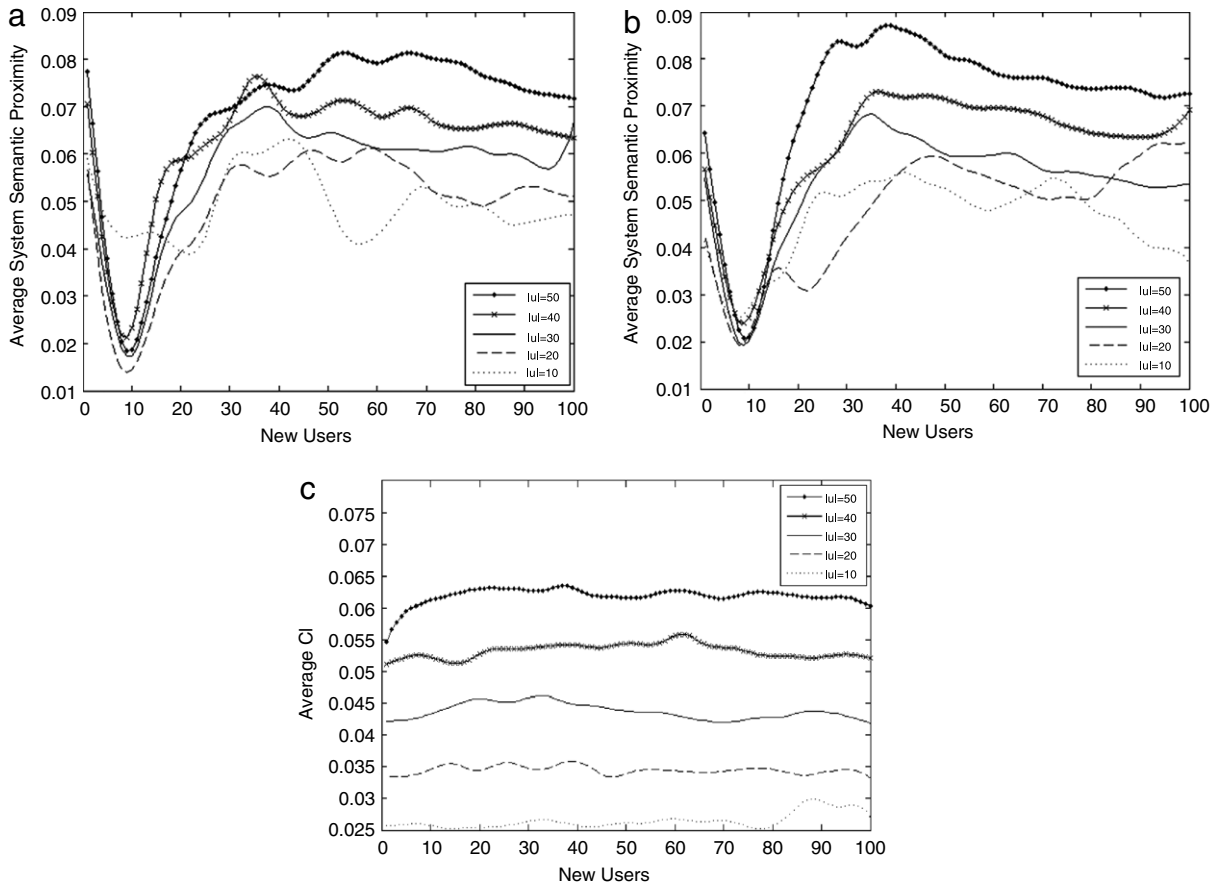

Fig. 9. Average System Semantic Proximity that is yielded after iterative spectral clustering application at each new user's arrival, for a system that is initialized by partitioning 50 (a) 100 (b) and 500 (c) users into 10 clusters, assuming Uniform distribution of users' interests.

(i) the performance of the clustering algorithms we compare (ii) the impact of profile cardinality variations and (iii) our heuristic method of dynamic user placement into social groups.

The first section of experiments is devoted to an evaluation of the static system behavior (i.e. how the system responds in the initial users' partitioning under various profile cardinalities used and under the assumptions of both Zipfian and Uniform preferences distribution). The second section assesses the performance of the service in a dynamic context of new users' joins/departures.

The system is evaluated under different initial user pools, and the results clearly indicate that the cardinality of the initializing set is a determining factor for the system progression. The metrics that are used to evaluate the service performance are (a) the standard deviation of the resulting distribution (a lower standard deviation indicates more equal-sized groups) and (b) the Average System Semantic Proximity.

6.2. Static system view

We compare the performance of three widely used clustering algorithms – k-means, hierarchical and spectral clustering – in terms of their conformance with our service requirements.

The adopted k-means implementation is the one described in Section 4.1. The initial group representatives are R (the number of clusters) randomly selected profiles around which the other user profiles are grouped: each profile is assigned to the group representative from which it has the minimum (semantic) distance; when this initial group assessment is completed, the actual group representatives are established and the distances of each profile from the respective groups are recalculated; the iteration proceeds until the algorithm converges. Regarding hierarchical clustering, we have chosen to compare the performance of the complete linkage version, since it gave the best results (in terms of conformance with our service requirements) compared to other implementations of this algorithm (i.e. average linkage, single linkage, centroid and median methods). Finally, the spectral clustering method that was tested is the one presented in Section 4.3. The similarity matrix $\mathbf{S}$ that was used as input to the spectral method encompassed entries p_{ij} which are given by (2).

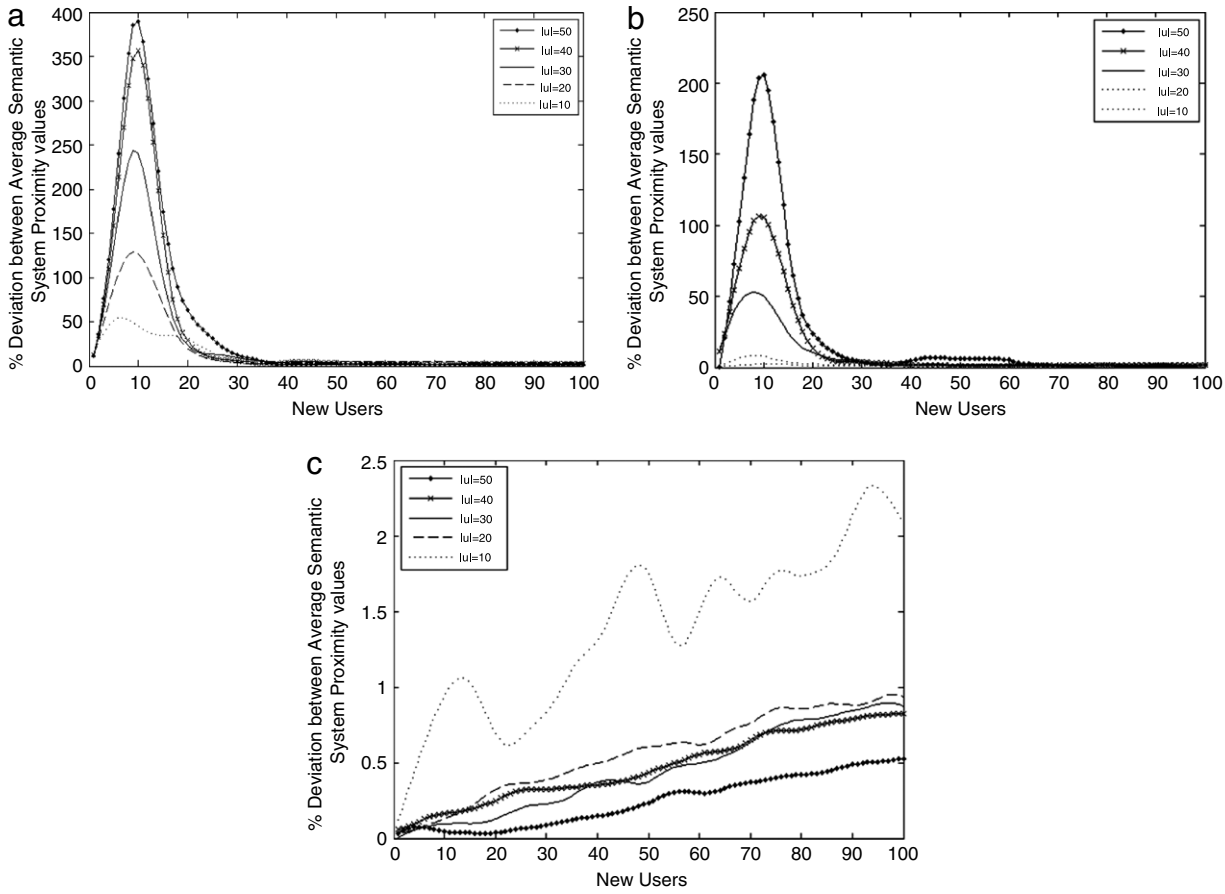

Fig. 10. Percentile deviation between Average System Semantic Proximity of our proposed heuristic and the Average System Semantic Proximity that would be yielded by a repetitive partitioning of using spectral clustering on each new user arrival; the figures assume Zipfian distribution of users' preferences and initial pools of 50 (a), 100 (b) and 500 (c) users.

A metric that expresses the suitability of each algorithm in the context of our framework is *standard deviation*: the lower the standard deviation at each distribution, the closer the cardinality of each group to the average cluster cardinality (which is a prerequisite for our framework). Fig. 1 provides the resulting standard deviation of the distributions resulting from the application of k-means, hierarchical and spectral clustering, respectively, versus profile cardinality; the keywords frequency distribution is assumed Zipfian in this case.

Three experiments were conducted, each one partitioning a different number of users (50, 100 and 500) so as to evaluate the impact of the initial pool of users in the standard deviation with which the system is initialized. Fig. 2 repeats the same experiment, assuming a Uniform user preferences distribution this time. The results appearing on both figures indicate spectral clustering's outperformance on the two other clustering methods, since the respective partitioning is characterized by significantly lower standard deviation for all used profile cardinalities, irrespective of the assumed preferences distribution and initial users group size. Spectral clustering, due to the optimization it offers in terms of normalized cut (see (5)), makes the algorithm applicable in the context of MAGNET Beyond social services.

Figs. 1 and 2 also provide a first feedback regarding the role of the profile cardinality in respect to the system clustering quality, which seems to be closely related to the underlying assumption in respect to user interests' distribution. All three diagrams appearing in Fig. 1 indicate that the standard deviation remains more or less unaffected by the increase of the profile cardinality. In contrast to these indications, Fig. 2 provides more clear indications regarding the correlation between these two values, since k-means and hierarchical clustering provide lower standard deviation values as the profile increases in terms of encompassed keywords. In fact, for a number of keywords equal to 40 and more, k-means gives distributions very close (in terms of standard deviation values) to the ones resulting in the application of spectral clustering. However, spectral clustering still gives the best performance, and succeeds in creating more equal sized groups than the other algorithms, even in this case, and for all values of $|u|$.

The next couple of figures (Figs. 3 and 4) focus on the assessment of Average System Semantic Proximity, in respect to the profile cardinality. Fig. 3 presents the results for Zipfian user preferences distribution while Fig. 4 repeats the same experiment under the assumption that these preferences are uniformly distributed (in terms of frequency of appearances).

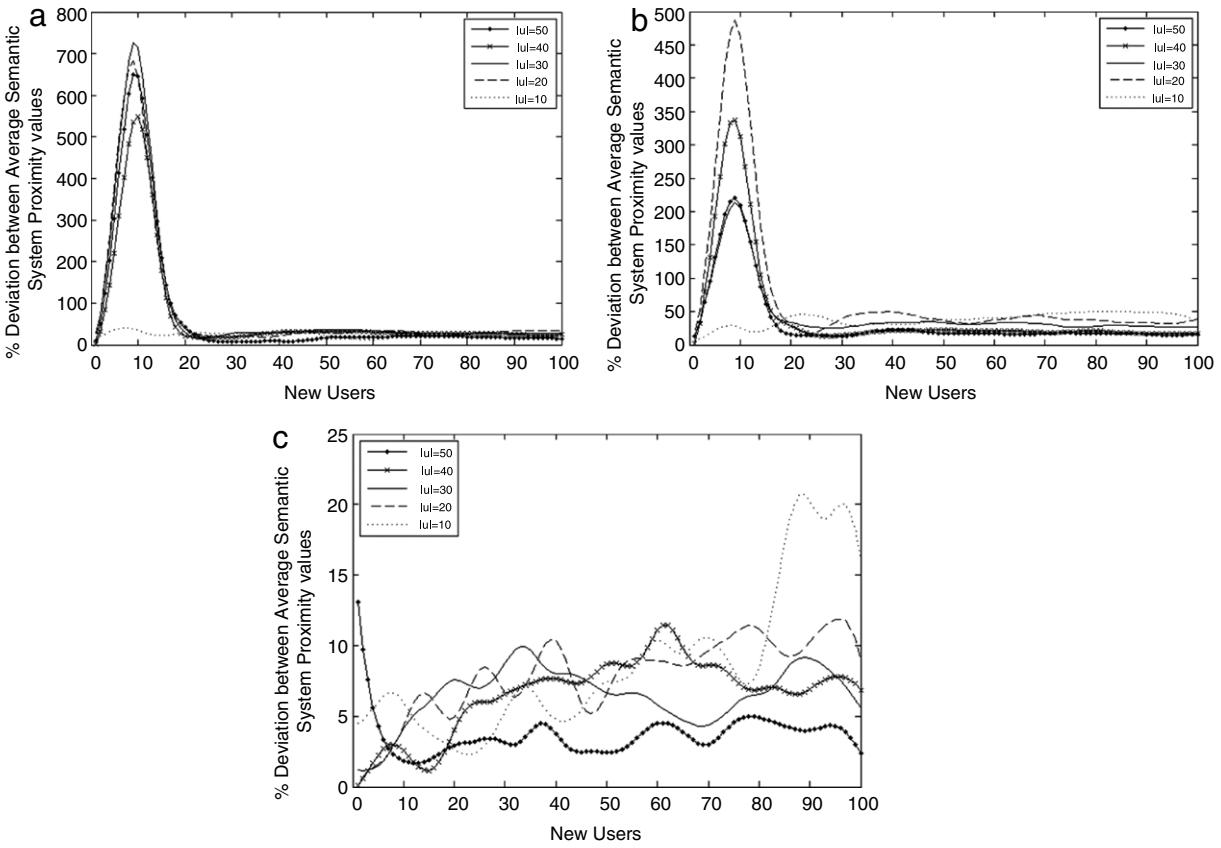

Fig. 11. Percentile deviation between Average System Semantic Proximity of our proposed heuristic and the Average System Semantic Proximity that would be yielded by a repetitive partitioning of using spectral clustering on each new user arrival; the figures assume Uniform distribution of users' preferences and initial pools of 50 (a), 100 (b) and 500 (c) users.

Fig. 3 gives a clear indication of spectral clustering's better quality of partitioning, since it is evident that, for all used profile cardinalities, the resulting groups present higher commonality of interests (i.e. on average, the users within the groups yielded after the application of spectral clustering share more interests than when applying the other two algorithms). It is also worth noting that the fewer users the system is initialized with, the more significant the performance difference between spectral clustering and the other two methods. Another inference that can be made by examining Fig. 3 pertains to the relation between profile cardinality and the system performance (which in this experiment is examined under the scope of $\overline{P_S}$): it becomes apparent that when the keywords' frequency of appearance follows Zipf's law, the increase of profile cardinality will not necessarily induce a performance improvement (i.e. the average keywords shared within a social group will not necessarily increase).

This relates to the specific characteristics of this distribution, since, as was also stated in Section 3.1, when assuming a Zipfian keywords frequency distribution, this implies few words that are very popular (i.e. will appear with very high probability in a user profile) while a large number of keywords is rarely used. In such a context, the percentage of keywords that will coincide within a group will be more or less steady and an increase in the profile cardinality will simply cause an increase of the probability of appearance of rare keywords, lowering or leaving unaffected the respective percentage of common keywords (depending on the skewness of the distribution).

Fig. 4 depicts the results of the same experiment as the previous figure, assuming now a Uniform distribution of user preferences. The first thing that can be noticed herein is that the performance variations between the three examined algorithms are practically negligible; all three methods attain to provide social groups with almost similar commonality of interests values, with slightly better values given by Hierarchical clustering. However, for profile cardinalities 30 and larger, the differences between the algorithms become practically indistinguishable. Another remark that can be made by examining Fig. 4, is that, in contrast to the previous case where user preferences were distributed according to Zipf's law, the Uniform distribution of this case makes the increase of profile cardinality yield an improvement in the clustering quality (always in terms of $\overline{P_S}$). Although such an increase seems to be small, it becomes evident, when also taking into account the results of Fig. 3, that the underlying distribution of users' preferences plays an important role in the overall system behavior, a conclusion that will also be substantiated in subsequent experiments.

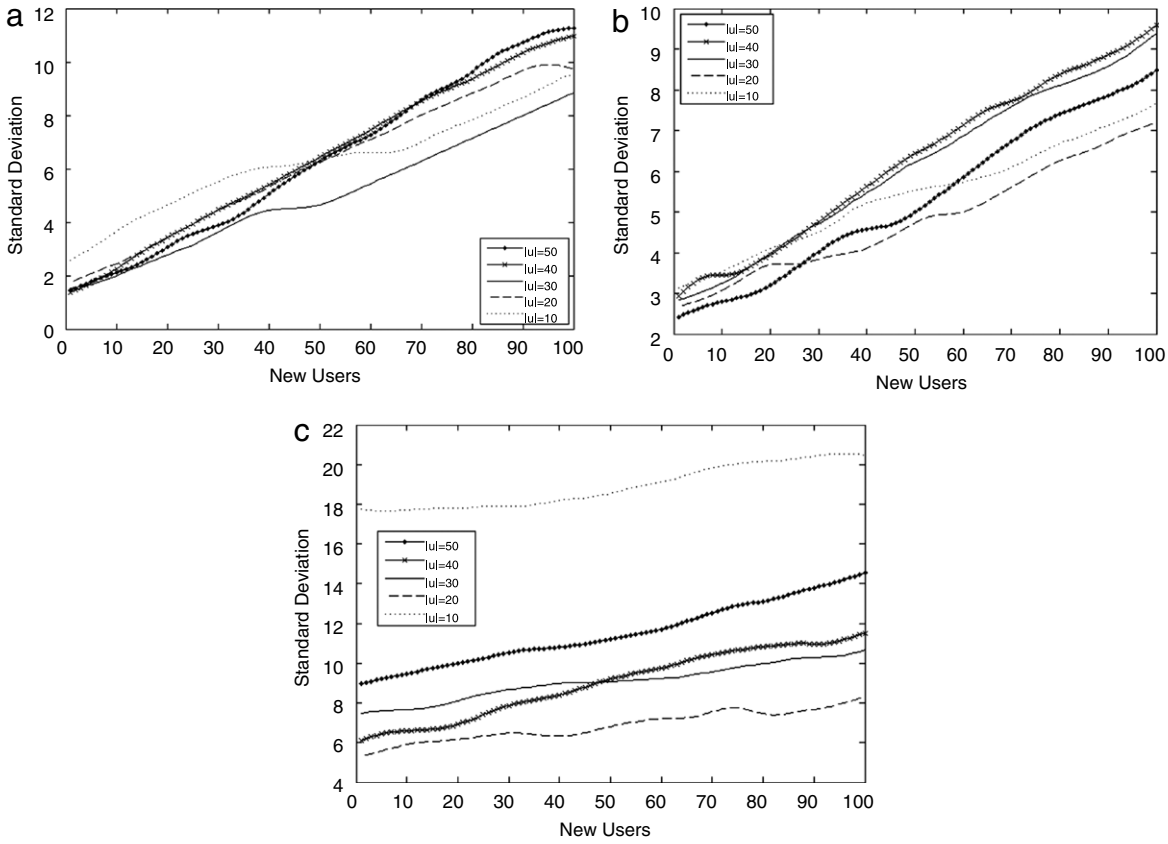

Fig. 12. Standard Deviation of users' partitioning distribution when applying our dynamic user placement heuristic, assuming initial pool of 50 (a), 100 (b) and 500 (c) users and Zipfian distribution of users' preferences.

Having substantiated the applicability of spectral clustering within the scope of *Icebreaker's* socialware, our final experiment in the view of static system performance assessment is to evaluate the impact of profile cardinality in the scope of clustering quality, as this is expressed by the (i) the Average Semantic System Proximity and (ii) the normalized cut value (see (5)).

In terms of $\overline{P_S}$, Fig. 5 presents the values of Average System Semantic Proximity for Zipfian (a) and Uniform (b) users' distributions versus profile cardinality, after the system's initialization through the application of spectral clustering. Although in the case of Zipfian (a) distribution of the keywords' frequency the increase of the profile cardinality does not seem to have a significant impact on the $\overline{P_S}$ values, in the case of Uniform distribution, using more keywords seems to provide slightly more coherent user communities.

6.3. Dynamic system view

The next set of experimental results is dedicated to the assessment of the system's performance in the view of dynamic user joins/departures into/from *Icebreaker* service. We assume the system is initialized using spectral clustering, since the results of the previous sections clearly indicated the algorithm's conformance to Icebreaker service requirements. Once the system is initialized, the heuristic algorithm introduced in Section 5 places new users that join the service into the appropriate social groups.

Once again, we have evaluated the behavior of the service assuming both Zipfian and Uniform distribution of user preferences; we have tested scenarios that combine initial pools of 50, 100 and 500 users, and have evaluated the impact of profile cardinality by using profiles composed of 10, 20, 30, 40 and 50 keywords. As in the case of the static system view, we have used, as indicative metrics of our service behavior, the standard deviation of the yielded partitioning and the Average System Semantic Proximity value. Since we now examine the system under a dynamic perspective, we have also made comparisons of the system evolution under the adoption of our proposed heuristic in relation to the resulting performance metrics that would characterize the system if, on each user arrival, we would re-decompose the users by an iterative spectral clustering application.

The first performance assessment results presented in Figs. 6 and 7 depict the evolution of Average Semantic System Proximity values yielded by the application of our heuristic algorithm, for various profile cardinalities. We notice that, in

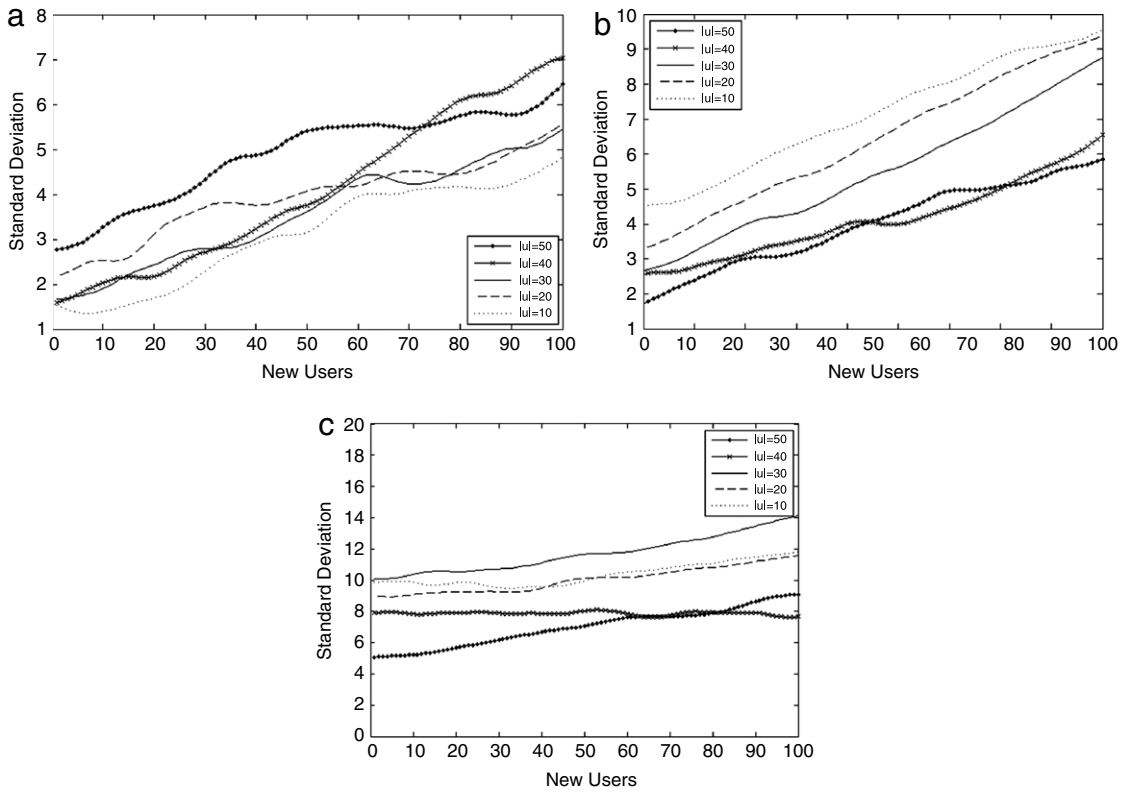

Fig. 13. Standard Deviation of users' partitioning distribution when applying our dynamic user placement heuristic, assuming initial pool of 50 (a), 100 (b) and 500 (c) users and Uniform distribution of users' preferences.

the case of Zipfian distribution, groups are more coherent in terms of interests' commonality, as in the case for Uniform distribution. In the first case, $\overline{P_S}$ takes values between 0.64 and 0.75 (maximum value of $\overline{P_S}$ is 1, which indicates full coincidence between users' interests within the social groups); in the second case, the respective values are significantly lower, around 0.05 (on average). This of course relates to the underlying distribution where we see that, in the usual cases, (where the keywords frequency of appearances follows Zipf's law) the system attains a high degree of social groups' coherence and maintains it at its progression. On the other hand, if these keywords present highly dispersed popularities, it becomes more difficult to attain the same values of $\overline{P_S}$ although, as before, an increase of the profile size has a positive effect in the system's performance.

The next two figures (Figs. 8 and 9) present the same results for iterative spectral clustering application on each new user arrival. The results indicate the inefficiency of such an approach in terms of system progression tracking, when the initialization is done with few users. In the cases of 50 and 100 users' initialization, it becomes evident that there is a significant drop of $\overline{P_S}$ value during the arrival of the first 20 users (50-users initialization case) and 30 users (100-users initialization case) for both Zipf and Uniform user preferences distribution. Such an effect becomes less prominent when adopting a small profile cardinality ($|u| = 10$); this, however, has the cost (in the case of Uniform keywords distribution) of lower $\overline{P_S}$ value when the system is stabilized.

Figs. 10 and 11 depict the percentile deviation between the $\overline{P_S}$ values attained by our proposed algorithm and the respective values of repetitive application of spectral clustering. In cases of system initialization by 50 and 100 users, we notice a significant deviation between these two values during the arrival of the first 20 users, which practically vanishes as new users join the system. This deviation, however, corresponds to the fluctuations of $\overline{P_S}$ that results from the application of spectral clustering; this becomes evident through a closer inspection of Figs. 9–12 which demonstrates that the values of $\overline{P_S}$ remain almost constant when applying our algorithm which, however, is not the case for the iterative spectral clustering application that presents a significant decrease of $\overline{P_S}$ values during first users' arrival. Nonetheless, in all cases, the system seems to settle down after the arrival of the first 20 users and, in the case of 500 users' initialization, the tracking is close to optimal for all incoming users, exhibiting higher deviation values in the case of Uniform preferences distribution.

The next set of experiments is devoted to the monitoring of Standard Deviation values regarding the distribution of social groups' population, as new users join the Icebreaker service. Figs. 12 and 13 depict the deviation of users' partitioning distribution when applying our dynamic user placement heuristic, assuming initial pool of 50 (a), 100 (b) and 500 (c) users and Zipfian (Fig. 12) and Uniform (Fig. 13) distribution of users' preferences.

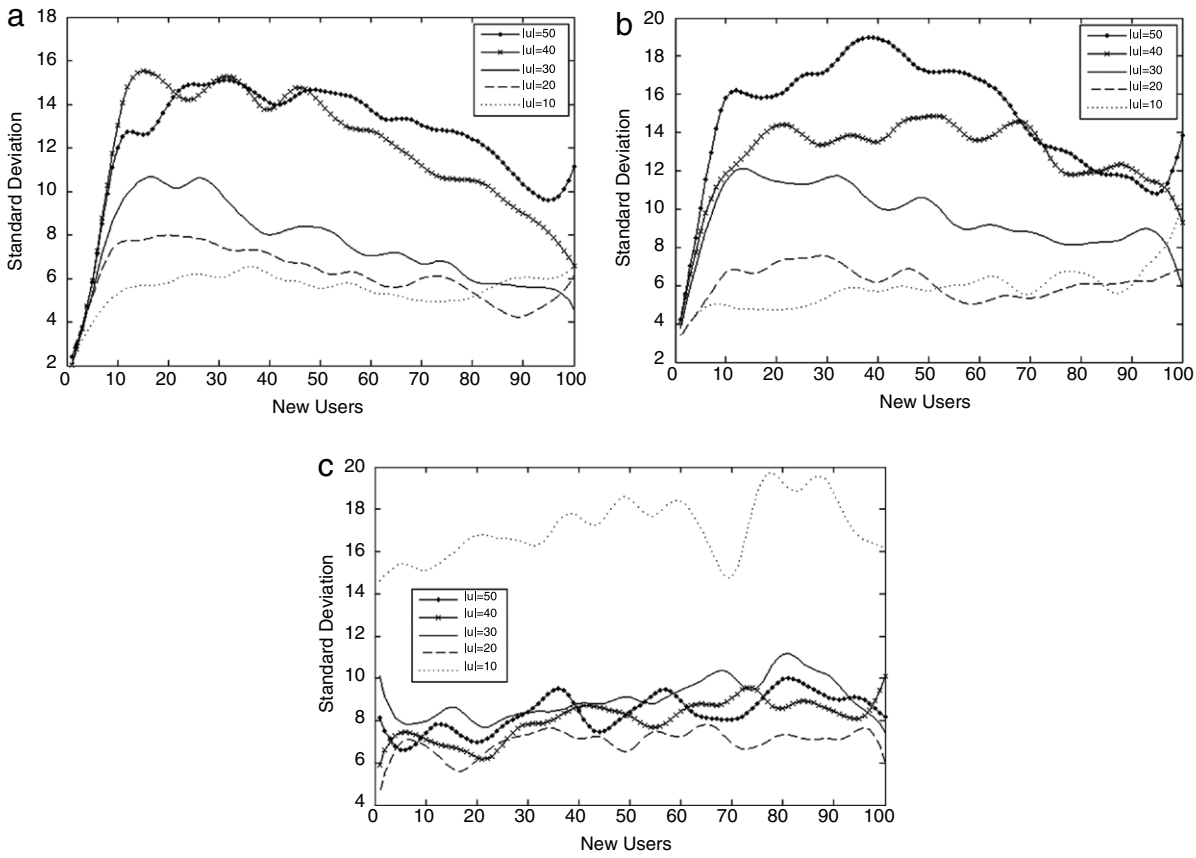

Fig. 14. Standard Deviation of users' partitioning distribution when applying spectral clustering, assuming initial pool of 50 (a), 100 (b) and 500 (c) users and Zipfian distribution of users' preferences.

The same results are depicted in the next couple of figures (Figs. 14 and 15), this time in regard to iterative spectral clustering application. The overall inspection of these graphs indicates that our proposed heuristic manages to maintain Standard Deviation of the yielding distribution in levels close to the initial partitioning, for almost all new user arrivals. In contrast, the system that is iteratively re-initialized by the spectral clustering application on each new arrival, presents significantly poorer performance, since there is a noticeable increase in the Standard Deviation values on the first arrivals.

A brief examination of system's behavior on user departures is offered by the next diagrams. Fig. 16 presents the percentile deviation between the Average Semantic System Proximity values yielded by the application of our heuristic algorithm and the ones that would be yielded by the iterative application of the optimal approach (i.e. of the spectral clustering). The results of a similar simulation are depicted in Fig. 17, this time in respect to the Standard Deviation values. In both cases, Zipfian distribution has been assumed in regard to frequency of keywords' appearance, as also an initial pool of 500 users. The two figures showcase the efficiency of our proposed heuristic in the case of users' departures too. More specifically, the Average Semantic System Proximity values yielded by (a) the optimal spectral decomposition and (b) our proposed heuristic, deviate only by a 0.8%. The respective deviation in regard to the Standard Devation metric is 12% (and this only occurs when a 10-keyword profile is used). In all other cases, the distance between optimal and proposed Standard Deviation values is no greater than 6%.

6.4. Cluster number selection

The issue of cluster number selection is more or less a matter of the requirements pertaining to the specific instantiation of Icebreaker's service. The selection of the number does not have a significant impact on the performance of the overall system. To substantiate this, we plot in Fig. 18 the value of Average Semantic System Proximity versus the number of clusters used to decompose the users. As is evident, the Average Semantic System Proximity value remains more or less the same for all cluster numbers (having a slight increase only when the 10-keywords profile is used).

7. Conclusions

In this work, we have presented results of ongoing work in MAGNET Beyond project, regarding theoretical evaluation of

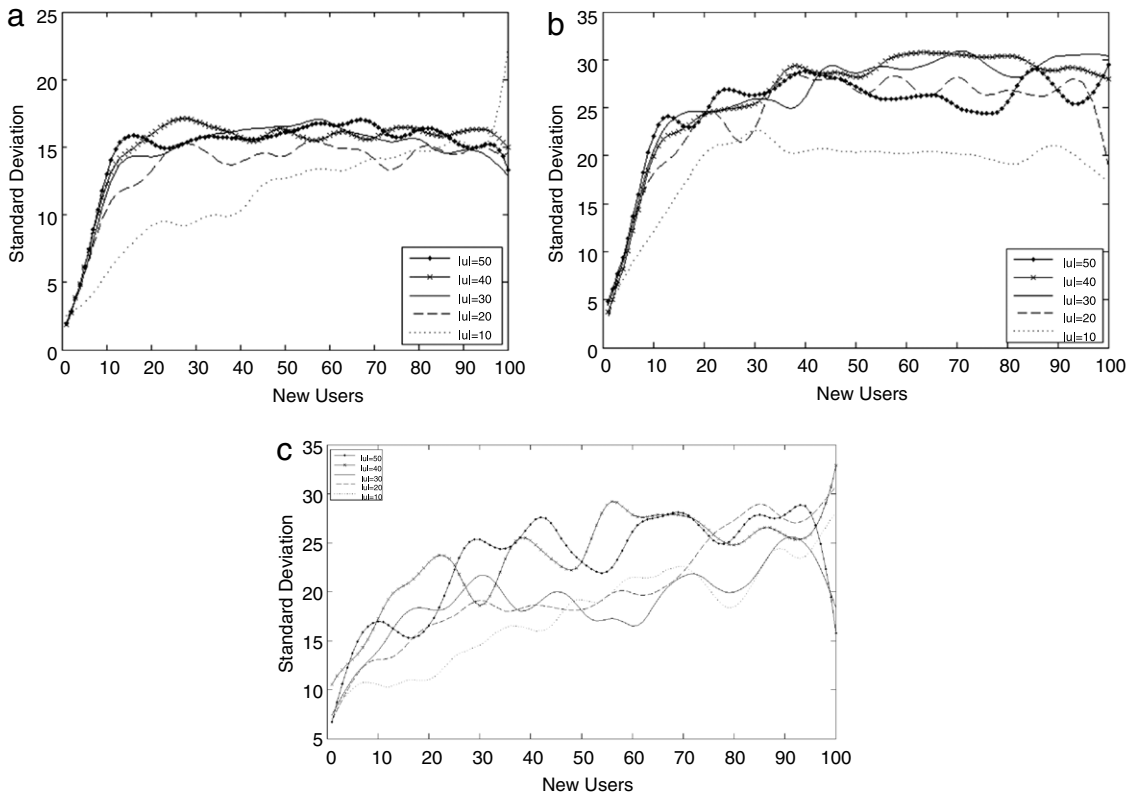

Fig. 15. Standard Deviation of users' partitioning distribution when applying spectral clustering, assuming initial pool of 50 (a), 100 (b) and 500 (c) users and Uniform distribution of users' preferences.

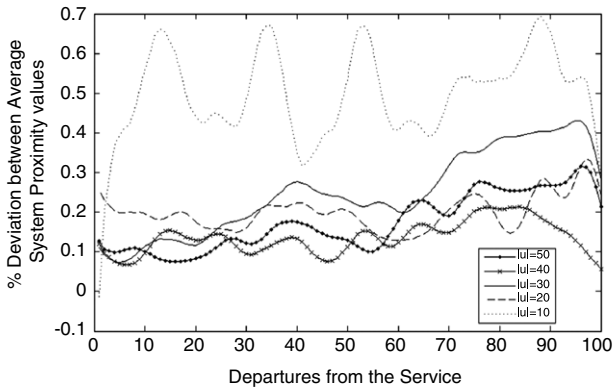

Fig. 16. Percentile deviation between Average Semantic System Proximity values yielded by users' partitioning distribution when applying spectral clustering and our heuristic method, assuming initial pool of 500 users and Zipfian distribution of users' preferences (user departures only).

an algorithmic framework for social networking services. We have introduced all the necessary metrics in order to provision for an integrated evaluation framework in regard to three widespread clustering algorithms (i.e. k-means, hierarchical clustering and spectral clustering). We have defined the notions of semantic distance, semantic proximity and semantic centroid and the Average System Semantic Proximity measure that allowed us to evaluate these clustering methods in the scope of Icebreaker's service requirements and in regard to the cardinality of the profile used to assess users' preferences. We examined our social network, both under static (initial user partitioning) and dynamic view (new users joining the service). Our static system examination clearly showed that spectral clustering provides better user partitioning in regard to both prerequisites (equal-sized groups and high commonality of interests between users of the same group). Regarding the impact of the profile cardinality in the overall system performance, our results showed that this is clearly dependent on the underlying frequency distribution used to characterize users' preferences (keywords). We introduced a heuristic algorithm that places new users into appropriately selected clusters, without the need of an iterative spectral clustering application,

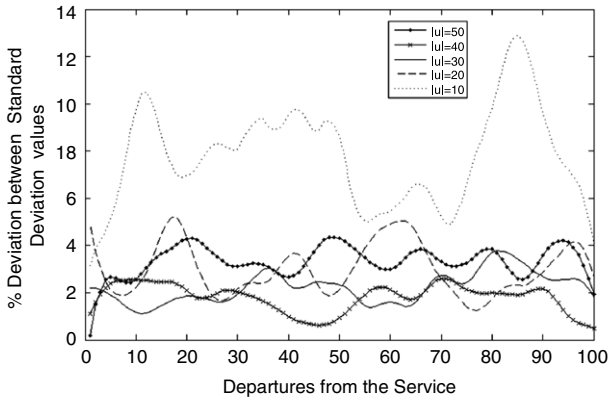

Fig. 17. Percentile deviation between Standard Deviation values yielded by users' partitioning distribution when applying spectral clustering and our heuristic method, assuming initial pool of 500 users and Zipfian distribution of users' preferences (user departures only).

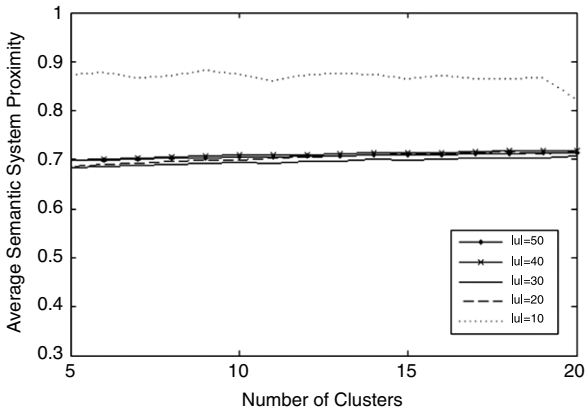

Fig. 18. Average Semantic System Proximity values in respect to number of clusters used.

which is a highly demanding computational process. Our results verified that, once the system has been initialized through spectral clustering on a number of users, our algorithm traces the evolution of the system efficiently (in terms of Standard Deviation and Average System Semantic Proximity values) and in much lower complexity.

The main difference from the majority of applications related to social networking lies in the very nature of Icebreaker. The service is destined to be deployed over conferences, large-scale meetings/events etc., usually offering short-time socialization. Therefore, due to the brevity that will characterize such service instantiations, long-term community formation and evolution will not be possible. Thus, service clients will not have the necessary ease of time to monitor the exhibited communities and select which of them to join, on the basis of the respective topics of interest. Such an approach is not new and has been incorporated in other relevant social networking services [42,43].

Our comparative study indicates that the two other clustering algorithms (hierarchical and k-means), tend to create highly dispersed groups sizes, which practically translates to the fact that, for example, in a scenario of 100 users, there may be several groups comprising 2–10 persons and a large cluster comprising about 70 users (these are real numbers, taken from specific experiments we have conducted). In a short-term socialization framework, suitable to be instantly deployed in events etc where there may not be any pre-assessment of user preferences, it becomes necessary to avoid the creation of isolated users, or groups incorporating just a couple of persons.

A future version of the proposed system is studied at present, which will deploy the proposed concepts into larger scale communities, found on the Internet. Other significant aspects that will leverage Icebreaker's potentials are the incorporation of stemming techniques (so as to deal with the presence of, for example, synonyms within profiles) and dynamic relevance feedback techniques [44,45] that will integrate efficient users' interaction with the system. The final step of Icebreaker's completion is to conduct mock-up demos with real users, so as to retrieve a final feedback before the final release of the service.

References

[1] S. Schiaffino, A. Amandi, Polite personal agent, IEEE Intell. Syst. 21 (1) (2006) 12–19.
[2] D.S. Phatak, R. Mulvaney, Clustering for personalized mobile web usage, in: Proceedings of the 2002 IEEE International Conference on Fuzzy Systems, vol. 1, USA, 2002, pp. 705–710.

[3] M. Smith, Tools for navigating large social cyberspace, Commun. ACM 45 (4) (2002) 51–55.
[4] X. Li, Buddy finding in the mobile environment, Technovation 25 (9) (2005).
[5] http://www.ist-magnet.org.
[6] J.E. Ulmschneider, J. Doszkocs, A practical stemming algorithm for online search assistance, Online Rev. 7 (4) (1983) 301–318.
[7] J. Xu, W.B. Croft, Corpus-based stemming using co-occurrence of word variants, ACM Trans. Inf. Syst. 16 (1) (1998) 61–81.
[8] S. Handurukande, A.-M. Kermarrec, F. Le Fessant, Laurent Massoulié, Exploiting semantic clustering in the eDonkey P2P network, in: SIGOPS European Workshop, September 2004, pp. 109–114.
[9] C. Seitz, M. Berger, B. Bauer, Mobile profile based distributed grouping, in: Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications Workshops, 2004.
[10] G. Palla, A.-L. Barabasi, T. Vicsek, Quantifying social group evolution, Nature 446 (2007) 664–667.
[11] M. Newman, Modularity and community structure in networks, Proc. Natl. Acad. Sci. USA 103 (23) (2006) 8577–8582.
[12] M. Newman, Detecting community structure in networks, Eur. Phys. J. B 38 (2004) 321–330.
[13] L. Danon, J. Duch, A. Diaz-Guilera, A. Arenas, Comparing community structure identification, J. Stat. Mech. (2005) P09008.
[14] L. Lancieri, N. Durand, Evaluating the impact of the user profile dimension on its characterization effectiveness: Method based on the evaluation of user community organizations quality, in: Proceedings of IEEE International Symposium on Computational Intelligence for Measurement Systems and Applications, 2003. CIMSA '03, 2003, pp. 130–134.
[15] D.N Sotiropoulos, G.A. Tzihrintzis, A. Savvopoulos, M. Virvou, A comparison of customer data clustering techniques in an e-shopping Application, in: Proceedings of 2nd International Workshop on Web Personalization, Recommender Systems and Intelligent User Interfaces, Ireland, 2006.
[16] K. Ntalianis, S. Ioannou, K. Karpouzis, G. Moschovitis, S. Kollias, Visual information retrieval from annotated large audiovisual assets based on user profiling and collaborative recommendations, in: Proc. of IEEE International Conference on Multimedia and Expo, 2001. ICME 2001, 22–25 Aug. 2001, pp. 1001–1004.
[17] ETSI, Human Factors (HF), User Profile Management, ETSI Guide, EG 202 325 v.1.1.1, October 2005.
[18] H. Liu, H. Wang, A. Feng, Applying information agent in open bookmark service, Adv. Eng. Softw. 32 (7) (2001) 519–525.
[19] O. Nouali, P. Blache, A semantic vector space and features-based approach for automatic information filtering, Expert Syst. Appl. 26 (2) (2004) 171–179.
[20] G. Salton, A. Wong, C.S. Yang, A vector space model for automatic indexing, Commun. ACM 18 (11) (1975) 613–620.
[21] T. Joachims, Learning to Classify Text Using Support Vector Machines, Methods, Theory and Algorithms, Springer, 2002.
[22] J.L. Huang, M.S. Chen, H.P. Hung, A QoS aware transcoding proxy using on demand data broadcasting, in: Proceedings of IEEE INFOCOM 2004, vol. 2, Hong Kong, 2004, pp. 2050–2059.
[23] A. Metwally, D. Agrawal, A.E. Abbadi, Efficient computation of frequent and top-k elements in data streams, in: Database Theory, in: Lecture Notes in Computer Science, vol. 3363/2004, Springer, 2005, pp. 398–412.
[24] E. Tonkin, Searching the long tail, in: 17th SIG/CR Classification Research Workshop, Austin, USA, 2006.
[25] L. Breslau, P. Cao, L. Fan, G. Phillips, S. Shenker, On the implications of Zipf's law for web caching, in: Proceedings of IEEE INFOCOM 1999, New York, USA, 1999.
[26] B. Huberman, P. Pirolli, J. Pitkow, R. Lukose, Strong regularities in World Wide Web Surfing, Science 280 (5360) (1998) 95–97.
[27] A. Kobsa, J. Fink, An LDAP-based user modeling server and its evaluation, in: User Modeling and User-Adapted Interaction, vol. 16, No. 2, Kluwer Academic Publishers, 2006, pp. 129–169.
[28] D. Burago, Y. Burago, S. Ivanov, Iu.D. Burago, A Course in Metric Geometry, American Mathematical Society, 2001.
[29] J. Bollen, R. Luce, Evaluation of digital library impact and user communities by analysis of usage patterns, D-Lib Magazine 8 (6) (2002).
[30] C.G. Minetou, S.Y. Chen, X. Liu, Grouping users' communities in an interactive Web-based learning system: A data mining approach, in: Proceedings of 5th IEEE International Conference on Advanced Learning Technologies, 2005, ICALT 2005, 5–8 July 2005, pp: 474–475.
[31] A. Rencher, Methods of Multivariate Analysis, in: Wiley Series in Probability and Statistics, 2002.
[32] R.L. Breiger, S.A. Boorman, P. Arabie, An algorithm for clustering relational data with applications to social network analysis and comparison with multidimensional scaling, J. Math. Psychol. 12 (1975) 328–383.
[33] D. Pierrakos, G. Paliouras, C. Papatheodorou, V. Karkaletsis, M. Dikaiakos, Construction of Web Community Directories using Document Clustering and Web Usage Mining, in: Proceedings of the 1st European Web Mining Forum, Workshop at ECML/PKDD-2003, Cavtat-Dubrovnik, Croatia, September 22, 2003.
[34] Y. Zhang, G. Hu, Using Web clustering for Web communities mining and analysis, in: The 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, Australia, 2008.
[35] U. von Luxburg, A tutorial on spectral clustering, Max Planck Institute for Biological Cybernetics Technical Report, No. TR-149, 2006.
[36] B. Hendrickson, R. Leland, An improved spectral graph partitioning algorithm for mapping parallel computations, SIAM J. Sci. Comput. 16 (1995) 452–469.
[37] M. Meila, J. Shi, A random walks view of spectral segmentation, in: Proceedings of 8th International Workshop on Artificial Intelligence and Statistics (AISTATS), 2001.
[38] S. Shortreed, M. Meila, Unsupervised spectral clustering, in: Proc. of UAI 2005.
[39] F. Bach, M. Jordan, Learning spectral clustering, in: Advances in Neural Information Processing Systems 16 (NIPS), MIT Press, Cambridge, MA, 2004.
[40] G. Semeraro, P. Basile, M. de Gemmis, P. Lops, Discovering user profiles from semantically indexed scientific papers, in: Web to Social Web: Discovering and Deploying User and Content Profiles, in: Lecture Notes in Computer Science, vol. 4737, Springer, 2007, pp. 61–81.
[41] Y. Takayama, R. Flournoy, S. Kaufmann, S. Peters, Information Retrieval Based on Domain-Specific Word Associations, in: Proc. of PACLING '99, Waterloo, Ontario, Canada, June 1999.
[42] M. Papagelis, D. Plexousakis, Recommendation-based discovery of dynamic virtual communities, in: Proc of CAiSE Forum 2003, Klagenfurt/Velden, Austria, June 2003, pp. 197–200.
[43] S. Chen, I. Radovanovic, J. Lukkien, R. Verhoeven, M. Tjiong, R. Bosman, Virtual community based secure service discovery and access for 3D video steaming applications, in: Multimedia Content Analysis and Mining, in: Lecture Notes in Computer Science, vol. 4577/2007, Springer, 2007, pp. 391–397.
[44] A. Doulamis, N. Doulamis, Generalized non-linear relevance feedback for interactive content-based and organization, IEEE Trans. Circuit. Syst. Video Technol. 14 (5) (2004) 656–671. Audiovisual analysis for interactive multimedia services (special issue).
[45] N. Doulamis, A. Doulamis, Optimal adaptive relevance feedback algorithms for interactive multimedia content personalization, IEEE Multimedia Mag. 10 (4) (2003) 38–47.