Soc. Netw. Anal. Min. (2016) 6:44
DOI 10.1007/s13278-016-0352-y

ORIGINAL ARTICLE

A synthetic data generator for online social network graphs

David F. Nettleton[1]

Received: 9 October 2015 / Revised: 12 June 2016 / Accepted: 17 June 2016 / Published online: 1 July 2016

Abstract Two of the difficulties for data analysts of online social networks are (1) the public availability of data and (2) respecting the privacy of the users. One possible solution to both of these problems is to use synthetically generated data. However, this presents a series of challenges related to generating a realistic dataset in terms of topologies, attribute values, communities, data distributions, correlations and so on. In the following work, we present and validate an approach for populating a graph topology with synthetic data which approximates an online social network. The empirical tests confirm that our approach generates a dataset which is both diverse and with a good fit to the target requirements, with a realistic modeling of noise and fitting to communities. A good match is obtained between the generated data and the target profiles and distributions, which is competitive with other state of the art methods. The data generator is also highly configurable, with a sophisticated control parameter set for different "similarity/diversity" levels.

Keywords Graphs and networks · Online social networks · Synthetic data generation · Topology · Attributes · Attribute-values · Seeds · Communities

1 Introduction

Online Social Networks have in recent years become of ubiquitous use by people all over the world for social interaction (Facebook) or in business/employment (LinkedIn). In July 2014, Facebook was valued at USD$192bn, its number of users having grown from 1 million in 2004 to 1.32 billion in 2014 (Weil 2015). In 2012, it was estimated that social networks were producing an estimated 2.5 Exabytes of data a day (Mc Afee and Brynjolfsson 2012).

In social networks, users can create sophisticated profiles, defining a rich online set of data about themselves. Also, their activity in the OSNs provides another descriptive dimension of themselves, including friendship links, communication with others, page likes, and so on. However, it is obvious that these data are personal and controls must be applied to protect the privacy of the users. The European Union's recent Data Protection Directive (EU 2015) details legal proposals for the future of how Big Data must be treated. Also, OSNs are susceptible to fraudulent use by the infiltration of fake users, which has been identified as a large scale problem (Kelly 2012). Personalization and user profiling enhances the user experience, but it is well known that user behavior analysis has implications for privacy (Jones et al. 2007; Ramakrishnan 2001).

In the context of research, data analysts who work for the OSN provider companies have a significant advantage with respect to researchers outside of these companies with regard to the access to and analysis of this data; however, we assume they must also follow the data privacy legislation in force. Hence, data mining researchers in this area have a serious limitation with respect to data access. One solution would be to conduct specific user studies which would inevitably imply reduced groups of volunteer users who allow a rich set of their OSN data, links and activity to become available for analysis. Another solution is to ask users massively to participate in a study, and they volunteer what data they are prepared to make available, in each case. Another solution is to use sampled datasets which

✉ David F. Nettleton
david.nettleton@upf.edu

[1] Universitat Pompeu Fabra, Barcelona, Spain

guarantee the anonymity of the users and which comply with legal requirements. These are solutions all related to real data. However, in the case of OSNs, simulated data would solve two key problems associated we have mentioned: data availability and data privacy. The option of generating realistic simulated data is the theme for the current work described in this paper.

One issue is how do we know that the simulated data are good or realistic? How can we measure this? Real data also has noise and random aspects, which have to be incorporated. However, we do have tools and definitions within our reach to help us. For example, we can know data distributions for many of the key demographic attribute values in typical OSNs: gender, age, marital status, and so on. Also, we know rules which apply to how people create links with others, based on affinities such as age, gender, residence, education, and so on. This may not give us a perfect match to a real OSN, but it may give us a good approximation which is valid for analysis purposes.

A significant body of research exists in the specialized literature with respect to generator and evolutionary models for topologies (graphs) which represent social networks (Chakrabarti et al. 2004; Lescovec et al. 2005; Robins et al. 2005; Viswanath et al. 2009; Kossinets and Watts 2006; Tang et al. 2008). However, works on populating these topologies with realistic data are more scarce (Pérez-Rosés et al. 2016; Ali et al. 2014; Barrett et al. 2009; Boncz et al. 2014) and these are often specific to a given domain or data type.

Hence, in this work our objective is to design and implement a general stochastic modeling system which allows us to populate a graph topology with data, following distribution profiles, attribute value definitions, using a parameterizable set of data propagation rules and affinities. We benchmark our method with different synthetic and real (ground truth) topologies, and the resulting data are evaluated structurally and statistically.

The paper is organized as follows: In Sect. 2, we describe related work for synthetic topology and data generators; in Sect. 3, we define some preliminary concepts related to graph topology; in Sect. 4, we describe our approach for data population of OSN graph topologies; in Sect. 5, we describe the control parameters for the generator; in Sect. 6, we present the empirical results; and in Sect. 7, we present the conclusions.

2 Related work

For convenience, the related work will be divided into three main areas: (1) synthetic topology generation without data; (2) generating a topology and then generating synthetic data which is then associated with the topology; and (3) homophily.

2.1 Synthetic topology generation

It can be said that the main body of existing work lies in topology generation without data, and a diversity of evolutionary models and generation algorithms exist to produce graph topologies which approximate the characteristics of a real social network (Nettleton 2013). Such characteristics are typically cited as being a small graph diameter, small average path length, skew degree distribution, and community structures. Sala et al. (2010) conveniently divided graph models into three classes based on their approach: feature driven, such as Forest Fire (Leskovec et al. 2005); intent driven, such as random walk and nearest neighbor; and structure driven, such as Kronecker graphs and dK-graphs. A benchmarking was conducted of these different models with respect to their ability to fit to a Facebook graph.

RMat (Chakrabarti et al. 2004) is a commonly used method which employs a statistical approach and a recursive process to replicate the power law distributions, skew distributions and community structure (which can be hierarchical), while maintaining a small diameter for the graph. The algorithm is optimized in terms of computation cost. Cross-links between communities are also represented. A recursive partitioning is carried out, which can be considered as a binomial cascade in two dimensions. The expected number of nodes c_k with out-degree k is given by:

$$c_k = \binom{E}{k} \sum_{i=0}^{n} \binom{n}{i} \left[p^{n-i}(1-p)^i\right]^k \left[1 - p^{n-i}(1-p)^i\right]^{E-k}$$

where ρ is the probability of an edge falling into partition "a" plus the probability of an edge falling into partition "b," and E is the number of edges in the real graph. Also, the number of nodes in the RMat graph is 2^n, where typically $n = \log_2 N$ and N is the number of nodes in the real graph. Figure 1 shows a graphical representation of the way RMat hierarchically processes the dataset. Descriptive

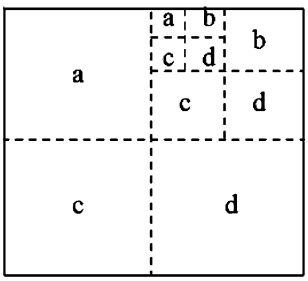

Fig. 1 Graphical representation of RMat hierarchical processing

parameters are used such as degree distributions, number of reachable pairs, number of hops, effective diameter, and stress distribution. One possible deficiency of RMat is the community structure. Boncz et al. (2014) and Pham et al. (2012) have reported that the generated topologies have communities with a similar size, instead of the long-tail distribution found in real OSNs. However, Chakrabarti et al. (2004) stated that RMat creates a hierarchical community structure, so a community extraction algorithm would have to take this into account. Also, real OSNs tend not to have neat community boundaries and the real situation is much more fuzzy and overlapping.

Robins et al. (2005) conducted a simulation for graph sizes ranging from 30 to 500 nodes. For a 100-node graph, up to 500,000 iterations were necessary to reach a stabilization of the statistical values. The model statistics used were: (1) number of edges; (2) number of 2-stars; (3) number of 3-stars; and (4) number of triangles. Aggregate measures (the graph statistics) are then calculated for: (a) degree distributions; (b) geodesic distributions; (c) clustering coefficient. A difficulty was found in the case of the "degree distributions," given that each sample had its own distribution. An "energy" value was defined and calculated for the graph at each iteration, the objective being to find the situation in which the energy reached a minimum. The authors cite four key conditions in order for a small world network to develop: (1) The individuals seek more than one network partner; (2) the costs of maintaining many partners is high; therefore, there is a tendency against a multitude of partners. Dunbar's limit (Dunbar 1993) gives a natural cognitive, sociological and anthropological maximum of 150; (3) there exists some tendency for network partners to agree about other possible partners, which leads to structural balance and clustering; (4) if point (3) is applied in excess, this produces cliques with insufficient links between nodes in order to give smaller path lengths. On the other hand, if it is not applied enough, there will be insufficient clustering in the network.

A model called "Forest Fire" (with reference to the way link creation propagates), is presented by Leskovec et al. (2005). In order to define the model, Leskovec first studied four "social network" datasets over time, in order to see how they change with respect to static models. The datasets studied are "arXiv citation HEP-TH," "patents citations," "autonomous systems (internet routers)," and "affiliation graph (ArXiv)." The main conclusions are that the graphs tend to get denser over time and the diameter tends to shrink; this last conclusion going against "conventional wisdom." They define a new graph generator, called the "Forest Fire" model, which is defined by the following: a densification exponent; a difficulty constant; a difficulty function; the number of nodes and edges at time "t"; a community branching factor; the expected average node out-degree; the height of the tree; H(v, w), which is the least common ancestor height of v, w; the forest fire "forward burning probability"; the forest fire "backward burning probability"; and the ratio of backward and forward burning probability. In terms of structure, the "rich-get-richer" (or preferential attachment) phenomenon is cited as the explanation of the heavy tailed in-degree power law distribution. Recursive community structures were found for computer networks based on geographic regions. For the patents dataset, the same situation was found in which conceptual groups ("chemistry," "communications," …) exist. In true OSNs on the other hand, users tend to group together based on "self-similarity." It is noted that in a citation database, a paper only generates outward bound links when it is created. On the other hand, inward bound links will be progressively generated and incremented over time. As a consequence of their observations, the authors require that their model creates a graph with the following characteristics: (1) "rich get richer"; (2) "copying" which leads to communities; (3) community guided attachment (densification); and (4) shrinking diameters.

2.2 Synthetic data and topology generation

In contrast to synthetic topology generation, less work exists in building a topology and associated data attributes and values to it. We will now review a selection of this work.

Firstly, the modeling approach of Pérez-Rosés and Sebé (2015) and Pérez-Rosés et al. (2016) is to simulate a LinkedIn network by defining a set of skills and for each skill define a directed graph where the nodes correspond to users' profile and the arcs represent endorsement relations. Five main skills (attribute values) were considered: "Programming." "C++," "Java," "Mathematical Modeling" and "Statistics." The base network and the endorsement digraphs are available at: http://www.cig.udl.cat/sitemedia/files/MiniLinkedIn.zip. An authority score is calculated (using the PageRank algorithm) for each node based on the quality and number of endorsements received. Leskovec's model (2008) was first used to generate an undirected network of contacts with 1493 nodes (contacts) and 2489 edges. The authors then use as starting point for data population a small sample taken from their own LinkedIn contacts (278 users in total), in which "skill" is the only attribute which can have one of five initial values. The skills are initially randomly assigned to nodes. A co-occurrence matrix is generated for the original skills, and this is used for the objective function to be matched in the generated data. The configuration is iteratively refined until a closest match is found with respect to the co-occurrence matrix. The original skills are clustered into two main

groups (programming and mathematical skills). Also, in Pérez-Rosés et al. (2016) additional skills are aggregated (up to 50) using an ontology and a deduction method based on the co-occurrences and a calculation using the Page Rank algorithm.

The approach of Ali (2014) is to grow a topology and assign the data at the same time, using preferential attachment to assign the attribute values which represent player skills. They also model interactions between players and game events in which the players participate. There are two phases: growth phase and attribute assignment phase. As the network grows, the homophily increases according to the formula $dh = i \times 0.05$, where i is the maximum node index and dh is the degree of homophily. This continues until the label homophily reaches a maximum predefined value. In the second phase, attribute assignment, node feature attributes are initially randomly assigned and then updated to fit the statistics of the original dataset. Links are created between nodes based on feature similarity, and additional links are created between nodes based on their degree of similarity. The attributes used represent crafting, movement, and combat skills possessed by the player of the corresponding online game. The fitness measure used by Ali for the attribute assignments is in terms of the average correlation between the target attributes and the generated attributes:

$$x = \frac{n(\sum xy) - (\sum x \sum y)}{\sqrt{\left[n\sum x^2 - (\sum x)^2\right]\left[n\sum y^2 - (\sum y)^2\right]}}$$

where x represents the target attribute values, y the assigned values and n the number of elements. Note that the attributes are represented numerically in a feature vector which acts as an index to the attribute values of the three attributes: craft, movement and combat skills.

Three datasets are used, one previously used by another author, DBLP-A (Wang and Sukthanker 2013) and two game player datasets which the authors have "scraped" from two online game APIs (Game X) (Travian). The DBLP-A dataset represents a collaboration network that includes information about 10,708 authors in 6 different computer science areas. In this network, the nodes represent the authors, and two authors are linked to each other if they co-authored at least one paper. The Game X dataset consists of data covering a 730-day period, and the Travian data are composed of one game cycle covering a period of 144 days. Game X is a browser-based exploration game in which multiple players act and interact as adventurers traveling a fictional game world in a vehicle. The dataset was extracted by observing gameplay between 3453 players. Two link types, message and attack, are defined. Nodes have attributes representing crafting, movement, and combat skills possessed by the player avatar. The Travian dataset (www.travian.com) was extracted from players participating in a massively multiplayer online game from the real-time strategy genre. This network has 7601 nodes, two link types (attack and message), and multiple time slices. Players have different roles, skills, allegiances, etc. The Travian dataset is available at: http://ial.eecs.ucf.edu/travian.php. Some references of other works which have analyzed Travian and Game X are Korsgaard et al. (2010), Wigand et al. (2012), Lakkaraju and Whetzel (2013) and Lee and Lakkaraju (2014). Specifically, Hajibagheri et al. (2015) study the nature of conflict and communication across two game worlds (Game X and Travian) that have different game objectives. They compare the structure of attack networks and patterns with trade and communication networks, with link types defined for attack, communication, and trading. Game analysis is a specific research area which is out of the scope of our current work.

Barrett et al. (2009) consider the generation and analysis of large synthetic social contact network. The work of Barrett et al. (2009) is somewhat different from the previous ones we have cited so far, because their objective is to model a countries population based on census type demographic and household information. This model is then used to predict spread of epidemics based on physical co-location (geographic proximity), and several large cities are benchmarked such as New York, Los Angeles and Seattle. A key aspect of the model includes trip/journey behavior of individuals based on their employment, population and household densities. They use labeled bipartite interaction graph to captures visits by people to different locations. Age group is also a key factor. They use: preschool (<5), school-age (5–18), adults (19–64) and seniors (>65).

An approach whose origins lie in the graph database field is that of Boncz et al. (2014) and Pham et al. (2012). It is oriented to the performance evaluation of "choke points" queries with a high computational cost, and of returning realistic results from SQL-type queries by creating local neighborhood based primarily on demographic data affinities.

In previous work (Nettleton 2015, 2016), an initial version of a synthetic OSN data generator was described for non-overlapping communities using an RMat (Chakrabarti et al. 2004) generated topology and simple control parameters. In Nettleton (2016), the synthetic data were used in a data privacy application.

2.3 Homophily, similarity and diversity

In this section, we discuss selected results from the state of the art, with respect to how "similar" individuals tend to congregate together in social networks. Works have been

chosen which identify specific key attributes and which quantify their effect on homophily. The counter-side of homophily is diversity, which also is a characteristic of human networks and which must also be taken into account. The works and empirical results described in this section have been taken into account when designing the seed-neighbor propagation mechanism detailed later in Sect. 5 of this paper.

The study by McPherson et al. (2001) defined homophily as the property of human networks to demonstrate an increased propensity for "like-minded" individuals to be connected, using the phrase "birds of a feather flock together." The authors concluded that homophily in race and ethnicity created the strongest divides in personal environments, with age, religion, education, occupation, and gender following in that order. Geographic closeness, families, organizations, and isomorphic positions in social systems also facilitated homophilous relations formation.

Wang et al. (2011) defined a metric which has been used by other authors in the state of the art, such as Ali et al. (2014). The link density of the network is controlled using a parameter, *ld*, and value homophily between agents using a parameter, *dh*. A high homophily value indicates that links are more likely to be formed between nodes with the same label; these labels can be viewed as being equivalent to community membership. Wang's generator supports the creation of binary, rather than continuous, node features that are designed to model personal attributes. The following formula was defined for "dynamic label homophily": As the network grows, the homophily increases according to the formula $dh = i \times 0.05$, where i is the maximum node index, until the label homophily reaches a maximum predetermined value. Wang's generator is limited to binary attributes. In Ali et al. (2014), Wang's generator was extended to include continuous node features and multiple link types, using stochastic optimization procedures for tuning the node features and link formation to match distributions from an existing social media dataset. It was commented that preferential linkages are often created based on attributes such as age, gender, or ethnicity. An agent network to model homophily was generated comprised of 500 nodes with 10 attributes, 4 attribute values, and using a *dh* value of 0.8 and 20 % noise.

Currarini and Vega 2013 focuses on the incentives and the forces that lead agents to distort their meeting rates away from the type composition at the population level. The equilibrium behavior implied by their model is of a threshold type, with agents "inbreeding" (that is, mostly meeting their own type) if their group is above a certain size. Thus, their conclusion is that larger groups, whose members are met with higher probabilities, would tend to make a smaller fraction of ties with dissimilar agents (inbreeding). On the other hand, "outbreeders" meet agents in a more restricted pool of outbreeders. Thus, if meeting is uniform, outbreeding groups should display an excess representation of other outbreeders. Outbreeding has a higher cost so the tendency is to inbreed. The threshold equilibrium of their model indicated that groups outbreed only when their size is below a given level.

The approach of Tarbush and Teytelboym 2012 is that homophily emerges from the correlations in an individuals' likelihood of interaction in similar social groups, as opposed to being governed by the allocation of time and by the relative size of the social groups in which agents interact. A high school dataset was processed which had the following attributes for each individual: gender, year of graduation, major, minor, dorm, and high school. homophily (in gender and year). The individual homophily index H_i in social category r of agent i at time t is defined as:

$$H_i = \frac{\text{number of friends of } i \text{ at } t \text{ that share } K_i^r}{\text{number of friends of } i \text{ at } t} = \frac{\sum_{\pi \in \prod_i^r} d_i^{\pi}(t)}{d_i(t)}$$

where $\pi_i^r = \{\pi_i(S) \in \pi_i | r \in S\}$ is a set of partition elements containing agents that share the characteristic K_i^r in category r with i. Another formulation was defined which expressed individual homophily as a function of degree instead of time.

Verbrugge (1983) defined a dyadic model in terms of occupation, age, gender, marital status, and geographic proximity, among others. They found that age, marital status, and occupation had the strongest effects on contact frequency. Young and elderly adults, never-married people, and students, production workers, and sales workers had the most contact with their friends. The R squared values in the model were 0.039–0.088 for age, 0.049–0.079 for marital status, and 0.013–0.074 for occupation. Age had a curvilinear relationship with friendship contact and coefficients were mostly positive for young adults (under 30) and elderly adults (70+), and mostly negative for ages 40–49. The author stated that this was consistent with the notion that job and family commitments are heaviest for middle-aged adults, leaving less time to see friends.

Pham et al. (2012) also analyzed a university student dataset and reported that city/location was an important factor for interaction, together with gender and date of birth (age). Correlation rules were used to define relations between attribute values and also for graph structure properties. They observed that in social networks, the number of friends of persons typically follows a skewed distribution (e.g., a power law): the majority of individuals have a few friends, and a minority have a lot of friends. A dictionary was used for attribute values which included a frequency distribution function based on the correlation

between the property values and the correlation dimensions. In our current work, we have followed a similar approach in which each attribute value has a pre-assigned target proportion.

(Kossinets and Watts 2009) identified several individual attributes that may represent different dimensions of homophily, specifically gender, age, status, field, year, and state (US). They found that adjacent pairs exhibit approx. 40 % higher similarity than the population average and that similarity is not only lower for non-friends than for friends, but decreases monotonically with distance from $d_{ij} = 1$ to $d_{ij} = 4$, where it approaches the population average. According to their model, similar individuals are far more likely to become acquainted than dissimilar individuals; specifically, the average tie-formation rate for a highly similar pair ($S_{ij} = 6$) is 50 times that for a highly dissimilar pair ($S_{ij} = 0$) and approx. 13 times that for a pair with average similarity ($S_{ij} = 2$). However, they questioned why successive rounds of induced homophily do not lead to a "balkanization" of the network, possibly even into disconnected, homogeneous components. Their explanation was that any process that reduces distances preferentially between already proximate pairs is inherently self-limiting, given that it is more difficult for already closely separated pairs to become closer still than it is for distant pairs. Also, a small fraction of "long-range" ties are always being formed and it is stated that even a small fraction of these is sufficient to ensure global connectivity thus acting as a natural brake on homophily.

Block and Grund (2014) concluded that having the same gender made a friendship tie among individuals who share the same ethnicity 1.22 times more likely (which is significantly less than the 1.92 that apply when ethnicity is ignored). Conversely, the additional effect for sharing the same ethnicity when two individuals already have the same sex is 0.06 and only marginally increases the chance for a tie to emerge (1.06 times more likely).

Kim and Leskovec (2011) calculated a homophily affinity matrix Θ for binary attributes in the LinkedIn network. The high values (0.8 and 0.9) on the diagonal entries of Θ indicated that link probability is high when nodes share the same attribute value. Their MAG model captured homophily and heterophily of different node attributes. Another dataset processed was the *AddHealth network,* a high school friendship network with 457 nodes and 2259 edges. Their goal was to study how real attributes explain the underlying network structure and which attributes affect the friendship formation and how. It was found that people were more likely to make friends of the same school year: students who are freshmen or sophomore being more likely (0.99) to form links among themselves than juniors and seniors (0.57). They calculated affinity matrix scores for each attribute and each attribute value. The score ranges for the attributes were as follows: school year (0.1–1.0), highest level math (0.3–0.8), cumulative GPA (0.4–0.8), AP/IB English (0.2–1.0), foreign language (0.4–0.7). Hence, it was found that affinity was not just attribute dependent, but attribute-value dependent (i.e., some attribute values have lesser affinity scores). This is a finding that we have taken into consideration in our synthetic data generator.

Finally, to conclude this section we will mention some recent user studies of content based online apps. Dehghani et al. (2016) performed an analysis of a corpus of approx. 700 K tweets and found that the distance between two people in a social network can be predicted based on differences in the "moral purity content" (love of same)—but not other moral content—of their messages. The focus was particularly on political affiliation in the USA, which they proposed manifested the well-defined ideological differences between political groups (especially liberals vs conservatives) in that country. However, Wattenhofer et al. (2012) performed a study of YouTube and reportedly found that only 12.49 % of users had more neighbors in the same main upload category than neighbors in another category. Similarly, for relation based on comments, only 10 % of the users had more than 50 % of their neighbors in the same main upload category. Therefore, at the user level, a lack of homophily was observed between linked users when comparing the main upload category. Cha et al. (2010) conducted other statistical testing for the presence of homophily (in terms of reciprocity) for sampled Twitter users and concluded, with high probability, that users linked with "following" relationships were interested in similar topics. The authors claim they found that reciprocity was content driven and attribute-based similarity (of the followers and the followees) was not so important. However, as the content chosen and shared between users is often correlated with such attributes as age, gender and sociocultural level, it can be said that the traditional OSN findings (McPherson et al. 2001) are still relevant for content-driven online apps.

3 Preliminaries

In this section, we formally define the following basic concepts which will be then used throughout the paper: graph, community, seed vertex, neighborhood of a seed vertex, target profiles for data assignment, fitness and homophily.

Graph Firstly, we define a graph G as a set of vertices V interconnected by a set of edges E, denoted by $G = (V, E)$. In this work, for modeling social interactions we assume that the graph is a weighted graph, that is for each edge *e* it has associated a numerical weight value *w(e)* [0,1] which is an indicator of the strength of relation (e.g., interaction

intensity). We consider the weighted graph G together with a table T, in which each tuple corresponds to a vertex v and has $\{a_1, a_2, a_3, \ldots, a_n\}$ as attributes and $\{va_1, va_2, va_3, \ldots, va_n\}$ as corresponding values.

Community The complete graph G is subdivided into communities $c \in C$, labeled by the Louvain method or by a real ground-truth community label. Communities are defined as being subsets of the whole graph in which nodes are densely connected and where nodes belonging to different communities are sparsely connected (Blondel et al. 2008). The quality of the resulting partitions is typically measured by a metric, such as modularity (Girvan and Newman 2002), that measures the density of links inside communities as compared to links between communities.

Seed vertex In order to avoid overlap/overwriting in the assignment of the data, we use a set of seed vertices that are going to be chosen with the following properties: Each seed has to have distance at least 2 to all the other seeds; each vertex of the original graph G is at distance at most 2 to some seed vertex; the seeds are chosen from the list of nodes in a community c, ordered by their distance to the medoid node of $c \in M_c$ as calculated by the centrality metric. The medoid M_c and centrality metric facilitate a homogeneous and optimum distribution of seed throughout the community topology. It is a natural assumption that the OSN graphs have to be similar between close acquaintances, hence, the condition of having a seed vertex at distance at most 2 guarantees that the vertices that are out from the set of seeds are at distance at most one from some seed's neighbor and therefore will intuitively be well represented. We denote the set of seed vertices for a given community as $S_c = \{sc_1, sc_2, \ldots, sc_n\}$.

Neighborhood of a seed vertex We denote the closed neighborhood of a seed vertex $s \in V(G)$ by $N(s)$, and it consists of all the neighbors of s in G together with s and all the edges of G that connect them. The neighborhood is a key aspect of the data propagation, given that a seed is assigned a profile directly, whereas its neighbors that are in the same community as the seed will be assigned a profile with a "similarity" to that of the seed, as determined by the control parameters described later in Sect. 5.

Target profiles for data assignment A set of target profiles P is defined in which each profile P is defined in terms of a set of attributes $a \in A$. An attribute will have a value $a_v \in A_V$ assigned from a subset which is defined for the given attribute. For a seed vertex sc_i, our data assignment method chooses the seed vertices $sc_2, \ldots, sc_n$ such that $M_c - sc_i$ is a minimum. These seeds are the ones to be assigned the predefined data profiles $P_1, \ldots P_c$. Each profile has an associated percentage value which represents the target proportion of the whole dataset that the profile should occupy. The success of the data assignment to the network can be measured as the ratio of the target proportion to the assigned proportion. Also, each attribute value has its own individual target and assigned proportion which are also fitted during data propagation.

Fitness The overall fitness of the generated dataset is evaluated by two criteria: (1) the difference between the target profile distributions and the assigned profile distributions in the whole dataset; (2) the difference between the target attribute-value distributions and the assigned attribute-value distributions in the whole dataset.

Fitness of profile distributions The assigned distribution for a given profile P_i is calculated thus:

$$\text{Assigned}(P_i) = \frac{\rho}{N} \tag{1}$$

where ρ represents the number of nodes which have the same attribute values as target profile P_i' and N is the number of nodes in the whole dataset.

If the target distribution of profile P_i is designated as Target (P_i), then the fitness for assigned profile P_i will be given as:

$$\text{Fitness}(P_i) = \frac{\text{Assigned}(P_i)}{\text{Target}(P_i)} \tag{2}$$

Fitness of attribute-value distributions A similar procedure is followed to the profile percentages, except that the profile is not considered, and the fitnesses of all values of a given attribute are averaged for that attribute:

$$\text{Assigned}(A_iV_j) = \frac{\sigma}{N} \tag{3}$$

where σ represents the number of nodes which have the same attribute value as target attribute value $A_i'V_j'$ and N is the number of nodes in the whole dataset.

If the target distribution of attribute value A_iV_j is designated as Target(A_iV_j), then the fitness for assigned attribute value A_iV_j will be given as:

$$\text{Fitness}\left(A_iV_j\right) = \frac{\text{Assigned}\left(A_iV_j\right)}{\text{Target}\left(A_iV_j\right)} \tag{4}$$

Then the fitness for attribute A_i will be the average fitness of all its possible values:

$$\text{Fitness}(A_i) = \frac{\sum_{j=1}^{|V|} \text{Fitness}\left(A_iV_j\right)}{|A_iV|} \tag{5}$$

Homophily of seed neighbors With reference to Sect. 2.3, homophily is defined as the property in which human networks that individuals with similar characteristics are more likely to become connected. With reference to Table 1, the measure of the homophily of the neighbors N of a seed vertex s is the averaged sum of the similarities of each respective attribute value av. We note that each attribute value av and attribute a can be weighted by $\urcorner^{av}$ in order to potentiate or dampen the overall contribution to

Table 1 Concepts and corresponding descriptions

Concept	Definition
Graph	We define a graph G as a set of vertices V interconnected by a set of edges E, denoted by G = (V, E). Each edge e has associated a numerical weight value $w(e)$ [0,1] which is an indicator of the strength of relation (e.g., interaction intensity)
Communities	The whole graph G is subdivided into communities $c \in C$, labeled by a community detection algorithm or by a real ground-truth community label. Communities generally contain users with common characteristics
Seed vertices	The set of seed vertices for a given community is denoted as $S_c = \{sc_1, sc_2, sc_n\}$ Seeds are assigned profiles and the profiles are then propagated to seed's immediate neighbors probabilistically. Seeds must be at least distance 2 from each other
Neighborhood of a seed vertex	Closed neighborhood of a seed vertex $s \in V(G)$ by $N(s)$ and it consists of all the neighbors of s in G together with s and all the edges of G that connect them
Target profiles for data assignment	Each profile P is defined in terms of a set of attributes a ϵ A. An attribute will have a value $a_v \in A_V$. Each profile and each attribute value will have a corresponding target proportion to be fitted during data propagation
Fitness of a profile	$\text{Fitness}(P_i) = \frac{\text{Assigned}(P_i)}{\text{Target}(P_i)}$
Fitness of an attribute value	$\text{Fitness}(A_i V_j) = \frac{\text{Assigned}(A_i V_j)}{\text{Target}(A_i V_j)}$
Homophily of seed neighbors	$H_s = \left(\sum_{n=1}^{n=\lvert N_s \rvert} \left(\sum_{a=1}^{a=\lvert A_s \rvert} \frac{\sum_{v=1}^{v=\lvert V_{sa} \rvert} sim(s(s_a,s_v),n(n_a,n_v)) \times \neg^{av}}{\lvert V_{sa} \rvert} \right) \Big/ \lvert A_s \rvert \right) \Big/ \lvert N_s \rvert$

the homophily measure. The degree of homophily is controlled by the assignment proportions of identical, closely similar and non-similar values for each attribute value. This will be described in Sect. 5.

4 Description of the method

The method has three overall steps which will be described in the following: topology generation/definition, data definition and data population. It could be debated that the data definition step should come first, followed by the definition of the topology, or that the topology should be evolved together with the data generation, such as in Boncz et al. (2014). However, in the present work, our focus and contribution is the population of an already existing topology, such as the ground-truth graphs we benchmark in Sect. 6. The topology generation/definition has two options: (1) synthetic topologies generated by RMat and then community identification using the Louvain method; (2) use of real topologies in which the "ground-truth" communities are already identified.

Once we have the topology and the communities assigned, we define the data we wish to use. We define the attributes and their values, together with the general percentage frequency in the complete population for each attribute value. Next we define a set of distinctive profiles in terms of the attribute values described previously. For each profile, we assign a target frequency which indicates the desired percentage of the records which will have this profile. The last step is to populate the empty topology with data, using the attribute values and the profiles defined previously. This is done by assigning "seed" nodes in each community and propagating data to their immediate neighbors and beyond until all the nodes in the graph have data assigned.

In Fig. 2, we see a schematic overview of the data processing steps which will be explained in the remainder of this Section. In Step 3, we see an iterative process in which the generated data are matched against the target profiles and distributions. The process iterates until the match between the target and generated values is within the required tolerance. Two in-line readjustments are available: seed reassignment (automatic) and control parameter readjustment (chosen from combination sets predefined from design of experiment).

4.1 Step 1: Topology preprocessing

Firstly, we have to obtain a topological structure. In the current, work we have applied two contrasting approaches. On the one hand, we have used RMat to generate synthetic topologies, and on the other hand, we have obtained topologies of real OSN community ground truths from the SNAP online repository (Amazon, YouTube and LiveJournal).

Community Labeling For the RMat-generated graph, we identify the communities in the graph structure by processing with the Louvain method (Blondel et al. 2008), which assigns a community label to each vertex in the graph. We note that we consider that the communities of the RMat model are non-overlapping. The effect of the topology and communities on the data assignment process is considered in more detail below in Sect. 4.1.1.

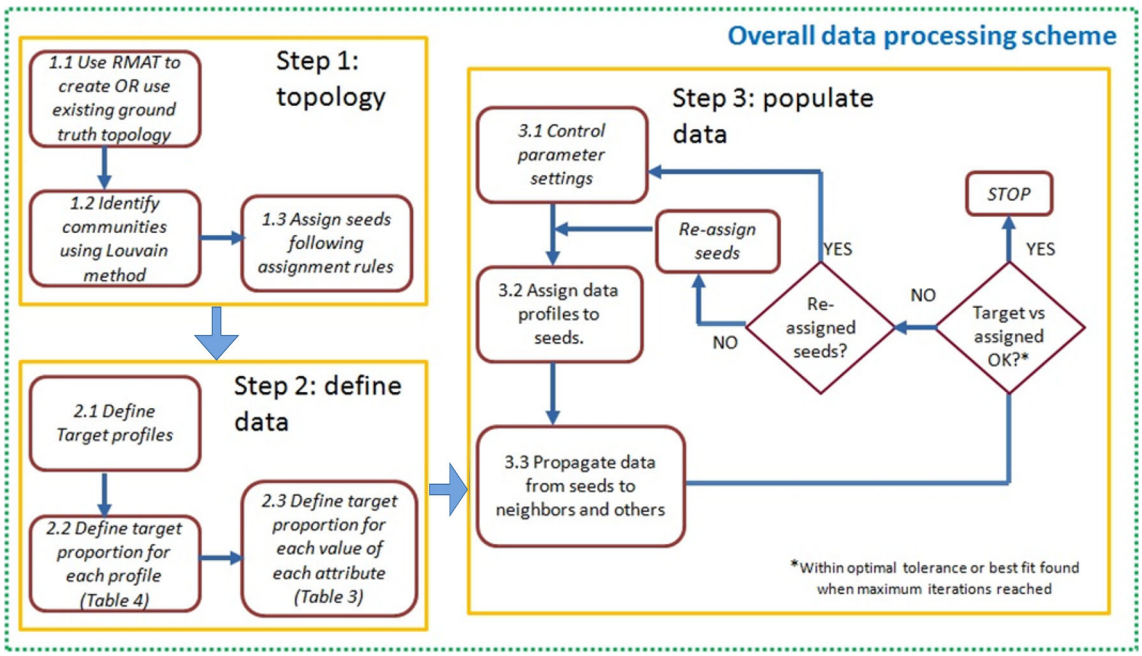

Fig. 2 Schematic overview of data processing

Table 2 RMat graph: communities and their % size with respect to the whole graph

Community Id	0	1	2	3	4	5	6	7	8	9
% of whole graph	0.216	0.172	0.211	0.097	0.081	0.157	0.024	0.005	0.028	0.009

For the ground-truth graph datasets, the (real) community is already identified, although we need some reformatting in order to obtain the required input files. In the case of the real graphs, the communities were overlapping, with a node potentially being a member of many communities. For each dataset, we chose the reduced (5000 top communities) option.

RMat-generated graph In Table 2, we see the size of each community as a percentage of the total nodes in the whole graph. We note that when we applied the Louvain method to the RMat-generated topology, it tended to obtain communities of a similar size. For the 1 K nodes graph, six communities were extracted by Louvain with the optimum modularity, corresponding to Ids 0–5 in Table 2, and with a size of between 17 and 22 % of the complete graph. Thus, in order to obtain some resemblance to a "long-tail" distribution of the community sizes, which is more typical of a real online social network, we applied the Louvain method recursively to communities 3 and 4 (which had the highest modularity values). From the resulting sub-communities, we chose the biggest and the smallest. For community 3 this gave us communities 6 and 7, and for community 4, this gave us communities 8 and 9. We note that the resulting communities, 3, 6, and 7 are non-overlapping. The same applies to communities 4, 8, and 9 (Fig. 3).

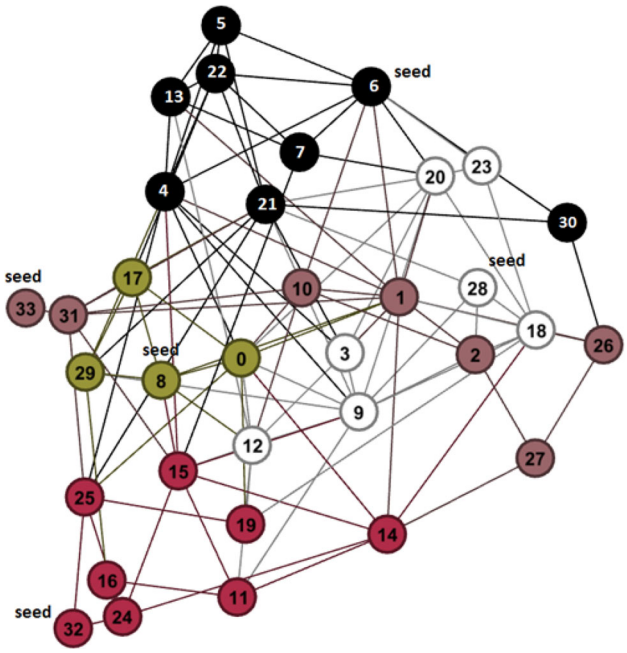

Fig. 3 Example of Rmat-generated topology. *Different colors* indicate communities. Seed nodes are indicated by "seed" (color figure online)

Seed Assignment Finally, we identify a set of seed vertices which will be used as the starting points for propagating the data. For each community, we find the medoid node in

terms of the statistical and topological characteristics (especially centrality and degree). Then we progressively assign seed vertices whose characteristics are closest to the medoid in each community, which gives a close to optimal coverage of the complete graph. We note that a rule was applied in which the neighbors of a seed node must be disjunct from the neighbors of any other seed node in the same community. This prevents overlapping of their immediate neighborhoods which avoids overwriting data propagated from different seeds. We were able to assign 110 seed nodes in this manner for the 1 K RMat-generated graph.

In the case of the ground-truth datasets, a similar procedure was followed, except that multiple community membership was taken into account. That is, if a seed node is a member of twenty communities, it is assigned a data profile just once and the assignment is registered for all the communities in which it is a member.

In Fig. 4, the pseudo-code is shown for the seed assignment procedure which, for a given graph with labeled communities, and a given number of seeds, attempts to assign the seeds to nodes in the graph such that each seed is at least distance 2 from any other seed.

4.1.1 *The effect of topology and communities on the data assignment process*

With respect to the influence of the community detection method to the overall data assignment process, it is clear that the community assignment has to be validated. For this reason, the empirical experiments cover both "artificial" communities (Sect. 6.1) generated with the RMat/Louvain method and "real" ground-truth communities (Sect. 6.2) from the Amazon, YouTube and LiveJournal datasets. The effect of the communities on the synthetic data generator also has to be widened to consider the real ground-truth community datasets, LiveJournal, Amazon and YouTube. These datasets are presented in Sect. 6.2, and their statistics are presented in Table 11 and Figs. 11 and 12. The two key statistics which affect the data propagation are, as shown in Table 11, "number of users per community" and "number of communities per user," that latter being a measure of degree of overlap. The real communities in these datasets represent user-defined groups (LiveJournal and YouTube) and groups based on product co-purchases (Amazon). We note that Pérez-Rosés and Sebé (2015) have a relatively simple concept of communities, based on the groupings of a small number of skills.

The Louvain community assignment method was chosen because it is widely used in the social network analysis community and is incorporated in major software packages such as Gephi (Bastian et al. 2009) and Python's NetworkX API (Schult and Swart 2008; Hagberg et al. 2004). The community assignment is validated first by the "modularity value," a widely used metric giving an "entropy" type value for the resulting network, and by the success of the target profile fit for the data assignment. As stated in Blondel et al. (2008), the objective of a community detection process is to partition a network into communities in which nodes are densely connected and where nodes belonging to different communities are sparsely connected. The quality of the resulting partitions is measured by a "modularity value" (Newman 2004) in which the modularity of a partition measures the density of links inside communities as compared to links between communities.

It is clear that the synthetic topology generation and the community assignment will both influence the performance and the results of the data propagation. A greater number of communities implies more processing for seed assignment and data profile propagation (especially boundary checking). However, even in the case of fragmentation into a large number of communities, the sum of the proportions of the assigned profiles can be easily verified by maintaining a subtotal vector which is checked against a target profile vector. One of the main aspects to validate is the long-tail distribution of sizes of the communities which is a key characteristic of online social networks. Also, communities should have well-defined boundaries, high intra-community connectivity and lower inter-community connectivity. For example, a hub node in a given community would not be expected to have 60 % of its immediate neighbors in

Fig. 4 Pseudo-code of seed assignment procedure

Procedure Seed_Assigner
Input: graph G, number of seeds desired nSeeds
Output: seed set S
1. **While** number of seeds assigned less than **nSeeds or** max iterations exceeded **do**
2. **For each** community c ∈ C in G **do**
3. **While** more seeds assignable **do**
4. **Choose** a vertex w from the set of nodes ordered by their centrality metric such that:
Each s ∈ S is at least at distance 2 from w.
End do
5. **End do**
6. **Save best configuration S' so far**
7. **End do**
8. **End Procedure**

Table 3 Example attributes, attribute values, and their overall target proportions for the complete dataset

Attribute	Values
Age	"18–25" (38.1 %), "26–35" (21.1 %), "36–45" (21.6 %), "56–65" (11.1 %), "66–75" (8.1 %)
Gender	Male (49.9 %), female (50.1 %)
Residence	"Palo Alto" (17.2 %), "Santa Barbara" (11.1 %), "Winthrop" (20.9 %), "Boston" (21.6 %), "Cambridge" (21.1 %), "San Jose" (8.1 %)
Religion	"Christian" (20 %), "Hindu" (11.1 %), "Jewish" (10.5 %), "Muslim" (15.7 %), "Buddhist" (21.1 %), "No religious affiliation" (21.6 %)
Marital status	"Single" (38.1 %), "Married" (29.7 %), "Divorced" (21.1 %), "Widowed" (11.1 %)
Profession (ISCO-08 structure)	"Manager" (21.1 %), "Professional" (20.9 %), "Sales and office" (21.6 %), "Student" (17.2 %), "Natural resources construction and maintenance" (11.1 %), "Production transportation and material moving" (8.1 %)
Political orientation	"Far Left" (8.6 %), "Left" (22 %), "Center Left" (17.2 %), "Center" (21.6 %), "Center Right" (20.9 %), "Right" (9.7 %)
Sexual Orientation	"Heterosexual" (78.9), "Bisexual" (21.1)
{like1, like2, like3}	Patterns: {"entertainment," "entertainment," "music artist"} (20.9 %), {"music artist," "music artist," "entertainment"} (29.2 %), {"drink brand," "drink brand," "entertainment"} (17.2 %), {"tv show," "drink brand," "soccer club"} (32.7 %)

another community. These aspects have been verified for the synthetic communities of Sect. 6.1. For overlapping communities, the validation is more complex, and this scenario is considered in Sect. 6.2.

Other paradigms exist for community assignment, such as label propagation-based algorithms such as LabelRank (Xie and Szymanski 2013), LabelRankT (Xie and Szymanski 2013) and GPSODM (Hajibagheri et al. 2013); however, a comparison of these methods is outside the scope of the present work. LabelRank is based on the idea of simulating the propagation of labels in the network. Here, we use node id's as labels. LabelRank stores, propagates and ranks labels in each node. During LabelRank execution, each node keeps multiple labels received from its neighbors. Nodes with the same highest probability label form a community. Briefly, a set of labels is defined to be propagated, and the system is initialized with a given label distribution over the nodes in the network. For each node, an entire distribution of labels is maintained and spread to neighbors. Each element $P_i(c)$ holds the current estimation of probability of node i observing label c element of C taken from a finite set of alphabet C. Newman's modularity measure (Newman 2004) is used as the fitness function.

4.2 Step 2: Data definition

The choice of data will be application specific. However, the distributions of the values of the different attributes should be similar to that of a real social network (ground truth). In order to achieve this, we can use sources of official statistics, such as government census data (www.indexmundi.com, www.census.gov, www.bls.gov), and statistical summaries made public by the social network providers, such as Facebook (www.adweek.com, fanpage list.com, http://royal.pingdom.com/2009/11/27/study-males-vs-females-in-social-networks). Example attributes and a priori proportions are shown in Table 3. We may also need some lookup tables for highly interrelated attributes values. For example, for the age group "18–25," there will be a much higher proportion of "profession = student" and "marital status = single," and for "gender = male" there will be a higher proportion of "like3 = soccer club." Another option for "ground truth" is publicly available OSN datasets, such as those found at the SNAP Web site: http://snap.stanford.edu/data/#communities.

Two specific sub-steps for data definition are as follows:

1. Define each attribute; define possible values for each attribute; define percentage of total population which has each attribute value (see Table 3).
2. Define data profiles (see Table 4).

RMat In the case of the RMat dataset, we define one profile for each community extracted by Louvain (we note that the number of communities can be controlled/limited by an input parameter). We also define what percentage of total dataset is desired (Target%) for each profile. The profiles are detailed in Table 4. Each profile will then be matched with the community whose percentage of the complete graph (in terms of number of nodes) is closest to the desired percentage for the profile, and assigned to its seeds (Tables 5, 6).

Ground Truth datasets These datasets have a much larger number of communities (5000) which display a long-tail size distribution. Hence, we have a fixed number of profiles (for example, 10), and we define an assignment probability to each (equivalent to the Target % used for the RMat dataset). Then, the seeds in the communities will

Table 4 Example profiles and their overall target proportions for the complete dataset

Profile Id	Profile	Target %
0	36–45, male, Boston, no religious affiliation, married, sale and office, center, heterosexual, TV show, drink brand, soccer club	0.216
1	26–35, female, Cambridge, Buddhist, divorced, manager, left, bisexual, music artist, music artist, entertainment	0.211
2	18–25, male, Palo Alto, Christian, single, student, center left, heterosexual, drink brand, drink brand, entertainment	0.172
3	18–25, female, Winthrop, Muslim, single, professional, center right, heterosexual, entertainment, entertainment, music artist	0.157
4	56–65, male, Santa Barbara, Hindu, widowed, natural resources construction and maintenance, right, heterosexual, TV show, drink brand, soccer club	0.097
5	66–75, female, San Jose, Jewish, married, production transportation and material moving, far left, heterosexual, music artist, music artist, entertainment	0.081
6	18–25, female, Winthrop, Christian, single, professional, center right, heterosexual, entertainment, entertainment, music artist	0.028
7	18–25, female, Winthrop, Jewish, single, professional, center right, heterosexual, entertainment, entertainment, music artist	0.024
8	56–65, male, Santa Barbara, Hindu, widowed, Natural resources construction and maintenance, left, heterosexual, TV show, drink brand, soccer club	0.009
9	56–65, male, Santa Barbara, Hindu, widowed, natural resources construction and maintenance, far left, heterosexual, TV show, drink brand, soccer club	0.005

Table 5 RMat topology

Community Id	0	1	2	3	4	5	6	7	8	9
Assigned profile	0	2	1	4	5	3	7	9	6	8

Assignment of profiles to communities based on their desired (profiles) and calculated (communities) % size with respect to the whole graph

be pseudo-randomly assigned the profiles depending on the assignment probability. For example, Profile 2 has a Target% of 21.1 and thus Profile will be chosen, on average, 21.2 % of the time to be assigned to the seeds of a given community.

4.3 Step 3: Data population

The four specific sub-steps for data population are as follows:

1. Assign each profile prototype to seeds of corresponding community. For RMat-generated topology, match profile percentages (Table 4) defined by user to community percentages present in topology. For ground-truth topologies, assign profiles with a probability proportional to the target percentage (see Table 4).
2. Assign neighbors of seeds based on profiles. Each neighbor attribute has a maximum allowed distance from corresponding seed attribute of z %. Neighbor attributes are assigned randomly k % of the time.

   In Fig. 5, the pseudo-code is shown for the procedure which first assigns the attribute values to the seeds based on the community profile definitions, and then assigns the attribute values to the neighbors of the seeds based on the assignment probability (dispersion level) and distance metrics for each attribute value. These criteria are included in the control parameter set $\mathbb{C}P$, as described in Sect. 5.
3. Assign attributes of nodes still unassigned (which are neither seeds nor neighbors of seeds). For each node, p% of the time a random assignment (by default $p = 10$) and q% of the time (90 % by default) each attribute is assigned the modal value of the neighbors of the node.

   In Fig. 6, the pseudo-code is shown for the procedure which assigns the attribute values to the vertices which have remained unassigned by the seed and seed neighbor assignment (procedure shown in Fig. 5). The attribute values are assigned based on the assignment probability and distance metrics for each attribute value. These criteria are included in the control parameter set $\mathbb{C}P$, as described in Sect. 5.
4. Check fitness of profile distributions for whole graph. If not within desired limits, return to previous steps and modify control parameters (see Sect. 5).

To initiate the population of the network with data, we use the set of seed nodes mentioned previously. The rest of the nodes will be assigned data by propagating from the seed nodes. The immediate neighbors of a seed will have a

Table 6 Allowed distance ranges for attribute-value assignment (seed to neighbors) in a community

Closest Distance thresholds								
Age	Gender	Residence	Politics	Sexuality	Religion	Marital	Profession	Likes
1/6	1	1/4	1/6	1/2	1/2	1/2	1/2	0.15

Fig. 5 Pseudo-code of data assignment to seeds and their neighbors

Procedure Assign_Data_to_Seeds_and_Neighbor_Vertices_in_Community
Input: S_c , the set of seeds in c; c, the current community id; control parameter set $\mathbb{C}P$
Output: NS_c , set of seeds and neighbor vertices with data assigned in community c

```
1.  For each vertex s ∈ S_c do
2.     Assign corresponding profile p_c to attributes of s
3.     Let Nv_c be the set of neighbors of s_c
4.     For each n ∈ Nv_c do
5.        For each attribute a of n do
6.           For each value v of attribute n do
7.              Assign {a, v} of ad_c to neighbor n according to ℂP
8.           End do
9.        End do
10.    End do
11.    End do
12. End Procedure
```

Fig. 6 Pseudo-code of data assignment to unassigned vertices

Procedure Assign_Data_to_Unassigned_Vertices_in_Community
Input: NS_c , the set of vertices in c with data assigned; c, the current community id; control parameter set $\mathbb{C}P$
Output: assigned set of vertices Vc in community c

```
1.  For each n in c ∉ NA_c do
2.     For each attribute a of n do
3.        For each value v of attribute n do
4.           Calculate average or modal value of
               corresponding attribute-value of neighbors
               of n as {n', a', v'}
5.           Assign {a', v'} or random value {a'', v''} to n according to ℂP
6.        End do
7.     End do
8. End Procedure
```

higher probability of being assigned similar attribute values. We also use ontologies/taxonomies and a distance measure to assign similar, rather than identical values (with an appropriate threshold) when propagating attribute values (Fig. 7).

The influence on assignment by the seed node has to be traded off by the desired overall proportions of the attribute values (diversity). In order to optimize the assignment, we can use a fitness function and find the optimum configuration for the control parameters using a stochastic process.

For example, Fitness = f(Ω, Φ, Γ), where Ω = set of seed assignments, Φ = set of data propagation and homophily ranges and distance thresholds, Γ = set of required data distributions (profiles).

5 Control parameters

In this section, we will describe how we can control the generator behavior by define a set of control parameters which can act generally and also on specific attributes and their possible values. In order to change the resulting distributions and assignments, we can vary these parameters. There are five major control parameters, which are: NS, number of seeds; {SP}, set of seed profiles; SD = similarity/difference; {DT} set of distance thresholds for each attribute; RU, random assignment threshold. Each of these parameters will now be described in detail.

(1) *NS*, number of seeds (110 for RMat 1000 node graph, 5000 for Amazon, 12,000 for YouTube and

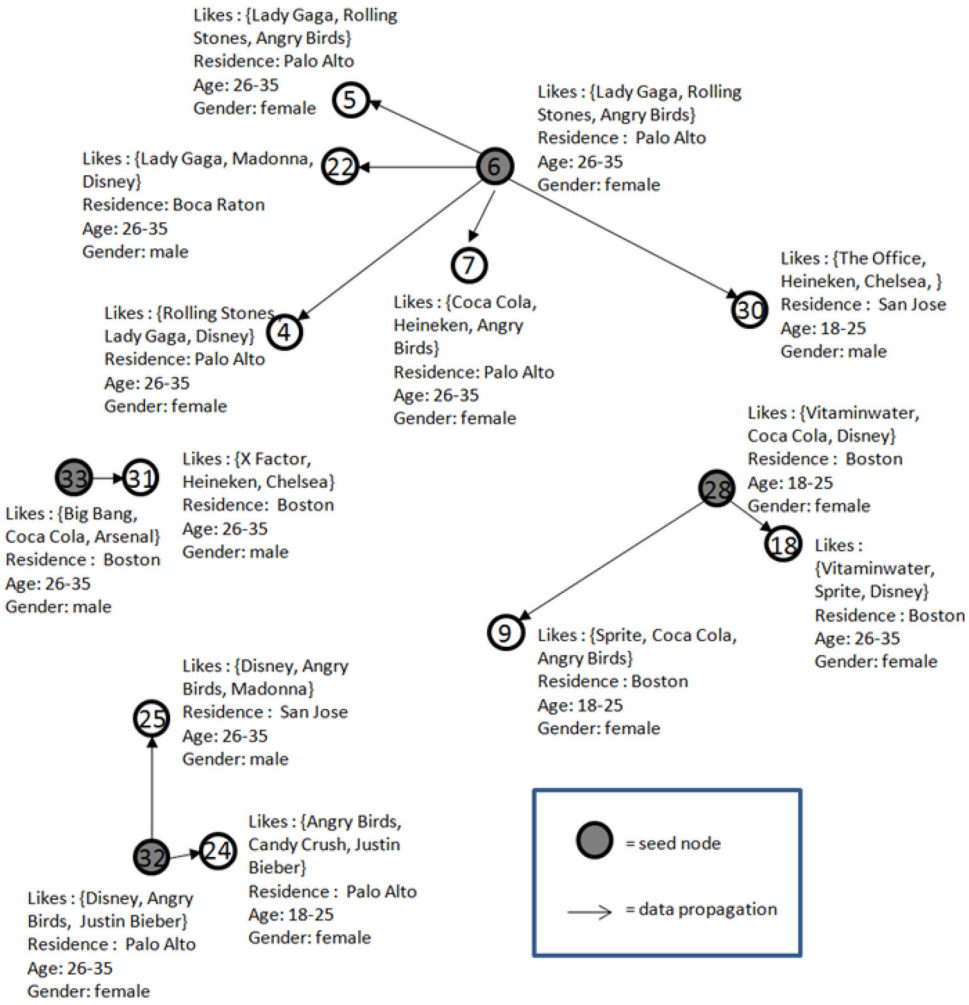

Fig. 7 Data propagation from seed nodes to immediate neighbors, in the topology of Fig. 3

LiveJournal). This value is dataset dependent, and by several trials on each graph dataset we found an optimum value in terms of processing time and coverage (evaluated as the number of nodes during processing which were neither seeds nor neighbors of seeds).

(2) *SP*, seed profiles. RMat: same number as communities, % desired for each. Ground truth datasets: probabilistic assignment of profiles to communities. Examples are shown in Table 4.

(3) *DT*, distance thresholds. These define the similarity between a seed attribute value and a neighbor attribute value and are set for each attribute. The distance range is attribute dependent, because the nature of the values affects how we calculate the distance. See Table 3 for a complete list of attributes and their possible values. In general, the first distance threshold is zero, which means the neighbor will be assigned the same attribute value as its seed; the second distance threshold is equal to the distance to the most similar distinct attribute value. For politics and age, it is 1/6, for gender it is 1, for religion, sexual orientation and marital status it is ½. For likes, two thresholds are used, 0.15 and 0.24. Residence has two thresholds, ¼ and ½.

Let us take the politics attribute and its values as an example. Consider the attribute-value politics = center. The two closest values to "center" for attribute politics are "center left" and "center right," which are at distance 1/6 from "center." Next we have "left" and "right" which are at distance 2/6 from "center." Finally, we have "far left" and "far right" which are at distance 3/6. So, we can define threshold 1 to be 0 and threshold 2 to be 1/6.

In practice, we keep the distance thresholds constant and calibrate the homophily proportions for each attribute value to assign the seed neighbors, using distance zero and closest distance to assign into the partitions.

Attribute "gender" is a nominal and has two possible values and the distance between two instance will be 0 (the same) or 1 (different).

Attribute "age" is an ordinal and has 7 possible values (categories). The distance goes from 0, in steps of 1/6 to 1. Attribute "political orientation" is also considered as ordinal and has 6 possible values (categories, see Table 3). The distance goes from 0, in steps of 1/6 to 1.

Residence is represented as a hierarchical category with four geographic levels (USA): county, state, division and region. If the residence of two instances is equal the distance is zero; if the residence is not equal but it is in the same county, the distance is 0.25; if the residence is not equal but it is in the same state, the distance is 0.50; if the residence is not equal but it is in the same division, the distance is 0.75; if the residence is not equal but it is in the same region, the distance is 0.90; otherwise, the distance is 1.0.

The distance between likes is calculated using an "affinity" table. When two likes are the same, the distance is zero; otherwise, the pair is looked up in the table to find the corresponding affinity. The distance between "entertainment" and "music artist" is 0.25; between "music artist" and "entertainment" is 0.25; between "tv show" and "drink brand" is 0.50; between "soccer club" and "drink brand" is 0.50; and so on.

When we calculate the distance, we sum the distance of each seed value to each neighbor value. For example, consider the case when the seed "like" values are: sv_1 = "entertainment," sv_2 = 'entertainment' and sv_3 = 'music artist' and the potential neighbor 'like' values are: nv_1 = "music artist," nv_2 = "music artist" and nv_3 = "entertainment." Firstly, we compare and calculate the distance of sv_1 to nv_1, nv_2 and nv_3 which gives corresponding distances of 0.25, 0.25 and 0.00 and a subtotal of 0.50. Now we do the same for sv_2 calculating its distance to nv_1, nv_2 and nv_3, which again gives 0.25, 0.25 and 0.00 and a subtotal of 0.50. Next we do the same for sv_3 calculating its distance to nv_1, nv_2 and nv_3, which gives 0.00, 0.00 and 0.25 and a subtotal of 0.25. Lastly we sum the three subtotal to give the distance between the tuples $\{sv_1, sv_2, sv_3\}$ and $\{nv_1, nv_2, nv_3\}$ as $0.50 + 0.50 + 0.25 = 1.25$. Finally, we divide by 9 to give a normalized value (between 0 and 1) of 0.139.

For the attribute religion, we have considered that "Buddhist," "Hindu," and "Sikh" have a relative affinity so if the religion for each of two instances is not equal but is one of these three, then their mutual distance will be 0.5. In a similar manner, if the religion for two instances is not equal but is one of "Christian" or "Jewish," then their mutual distance will be 0.5. Otherwise, the distance will be zero (different) or 1 (equal).

For the attribute marital status, we have considered that "Married," "Divorced" and "Widow" have a relative affinity so if the marital status for each of two instances is not equal but is one of these three, then their mutual distance will be 0.5. Otherwise, the distance will be zero (different) or 1 (equal).

For the attribute profession, we have considered that "Manager" and "Professional" have a relative affinity so if the profession for each of two instances is not equal but is one of these two, then their mutual distance will be 0.5. In a similar manner, if the profession for two instances is not equal but is one of "Service" or "Sales and office," then their mutual distance will be 0.5. The same applies for the professions "Natural resources construction and maintenance" and "Production transportation and material moving." Otherwise, the distance will be zero (different) or 1 (equal).

Finally, we define a weight w which is assigned to each edge e (link between two user nodes in the graph) where $w(e) \in [0,1]$ which is an indicator of the strength of relation (e.g., interaction intensity). This is calculated as the last step of the processing, when all the data are assigned. The value of the weight is calculated as the "grade of similarity" or "distance" between the respective attribute-value sets of two user nodes. The distance between each attribute value is calculated in the same manner as we have described in this Section for the distance thresholds (DT). The overall distance is given by the weighted sum of the attribute values, where an equal attribute weighting is used by default.

We note that it is clear that the rules we have defined are modifiable depending on the data, context and application.

(4) *SD*, similarity/diversity. This controls the similarity (sometimes referred to as homophily in the literature) and the diversity of the attribute characteristics, which are propagated from a given seed node to its immediate neighbors. This follows some general rules, but then is calibrated for each attribute and each topology, generating datasets to find those which give the closest fit to the desired profile and attribute percentages. We recall that seed nodes are assigned a given profile, which corresponds to a set of attribute values. In order to establish orientative values for the attribute value and homophily ranges, we have evaluated the empirical results from the state of the art, which has been presented previously in Sect. 2.3.

Three scenarios are considered for each attribute $value_{ij}$: **(a)** Immediate neighbors of a seed are assigned same attribute $value_{ij}$ as seed; **(b)** immediate neighbors of a seed are assigned an attribute $value_{ik}$ which is equal to the "closest" non-identical attribute $value_{ij}$ of the seed; **(c)** immediate neighbors of a seed have an attribute $value_{ij}$ whose normalized distance is greater than the "closest" non-identical attribute $value_{ij}$ of the seed.

Hence, we can define three distances corresponding to these scenarios: d_0, d_c and $d_{>c}$ and three corresponding

proportions of the neighbors who are assigned attribute values with that distance from the seed attribute value. The proportion values are defined within a range and must sum to 1.0. The ranges are defined manually, depending on the characteristics of each attribute and value. Then the exact proportion is calibrated by an iterative process which generates random numbers with a Gaussian distribution with a given mean and standard deviation. The homophily range is calibrated in the same manner. The iterative process stops when the fitness (see Sect. 3) reaches a given maximum value or the maximum number of iterations is reached.

For example, consider the case of a seed with attribute "political orientation"="center." In order to propagate and assign the attribute "political orientation" to this seeds' neighbors, the following is done: assign x % of the neighbors with a value which has distance zero (neighbor attribute value same as seed attribute value), which will be "center." Then for y % of the neighbors, we assign "political orientation" a value with the "closest" distance of 1/6 (neighbor attribute value is at distance 1/6 from seed attribute value), which will be "center left" or "center right" (one of these two options is selected based on overall proportions table for attribute values). Finally, for z % of the neighbors we assign "political orientation" a value with distance $> 1/6$ (neighbor attribute value is at a distance greater than 1/6 from seed attribute value), which will be one of "left," "right," "far left" or "far right" (one of these two options is selected based on overall proportions table for attribute values). I*n the case of attribute "age," which has 7 ordinal categories, x was calibrated (see previous explanation to have a proportion of* $\approx$ *48 %, y to* $\approx$ *31 % and z to* $\approx$ *19 %.*

As commented previously, the percentages x %, y % and z % are calibrated using an iterative Gaussian generator and the target proportions as the fitness objective. The homophility value is also calibrated within this loop.

We recall from Sect. 3 that the homophily for a given seed and neighbor data assignment is established by calculating the similarity of each of its neighbors as the average of the distances based on the corresponding attribute-value distances. Hence, we can run tests and calibrate a given homophily value which also has a close fit to the required profile percentages. The range for the homophily value is assigned from typical values of the state of the art (see Sect. 2.3), and taking into account the profile targets, we have defined for the seeds.

(5) *RU*, random assignment threshold for unassigned nodes. Nodes which are neither seeds nor neighbors of seeds can have their attribute-values assigned randomly or they can be assigned as the mean/modal values of their neighbors which have already been assigned attribute values. An example threshold would be 30 % (1-M^m), that is, 30 % of the time the assignment is random and 70 % (M^m) of the time the assignment is based on the modal values. Making the threshold bigger will make the community less homogeneous and less similar to the seed profiles. This may be useful in the case that we wish to control the overall distributions of minority attribute values. For example, in the current overall distribution we have a relatively high proportion (approx 17 % of "religion = Buddhist." If we wish to make the overall distributions representative of, for example, the USA, we would have to reduce this overall proportion. This could be done by increasing the random assignment for the corresponding profile/community.

(6) Control Parameter Set $\mathbb{C}P$. A first control parameter set is defined as [NS, {SP}, SD = H^h, {DT}, RU = M^h]. This corresponds to a lower intra-community diversity and a high homophily between nodes and their neighbors. We designate this as "level 1."

A configuration for a somewhat higher dispersion and medium homophily than level 1 would be [NS, {SP}, SD = H^m, {DT}, RU = M^m]. We designate this as "level 2."

A configuration for a somewhat higher dispersion and lower homophily than level 2 would be [NS, {SP}, SD = H^l, {DT}, RU = M^l]. We designate this as "level 3."

In Table 7, we see a summary of the range assignment scheme for the three levels of dispersion we have defined and tested. The homophily is measured using the formula as defined in Sect. 3, and as an example, the corresponding range limits could be assigned as $H_1^h = 0.8$, $H_u^h =$

Table 7 Control parameters $\mathbb{C}P$ for seed to neighbor and non-neighbor assignment

Dispersion Level	Neighbor homophily with seed ranges[a]	RU ranges[b]	
		Medoid neighbors	Random
Level 1—Low	$H^h: H_l^h \rightarrow H_u^h$	$M^h: M_l^h \rightarrow M_u^h$	$1- M^h$
Level 2—Medium	$H^m: H_l^m \rightarrow H_u^m$	$M^m: M_l^m \rightarrow M_u^m$	$1-M^m$
Level 3—High	$H^1: H_l^1 \rightarrow H_u^1$	$M^1: M_l^1 \rightarrow M_u^1$	$1-M^l$

[a] Ranges for distance assignment to neighbors

[b] Ranges for unassigned nodes

1.0, $H_l^m = 0.6$, $H_u^m = 0.8$, $H_l^l = 0.4$ and $H_u^l = 0.6$. These ranges have been orientatively assigned by studying the empirical results in the state of the art (see Sect. 2.3 of the paper) and applying the homophily formula (see Sect. 3) to the attribute-value data for the target proportions (see Sect. 4). *As described previously, the exact value is then calibrated in an iterative process by generating Gaussian random numbers within the range and giving the mean (range midpoint) and standard deviation (taken from the range).*possible. Our synthetic data

In order to calibrate the control parameter sets for different dispersion levels, a "Minitab" approach (MiniTab 2010), specifically the 2-level factorial (2–15 factors) option, was used to design an experiment set and execute systematic testing. This allows the investigation of the effects of input variables (control parameter set) on an output variable (data assignment distributions). The experiments then consist of a series of runs in which purposeful changes are made to the input variables, and the results are collated. An analysis of the results allows the identification of the process conditions that affect the quality (fitness, match of desired distributions to generated distributions), from which the factor settings are determined that optimize results.

Thus, a test set of trials with different control parameter sets, graphs datasets, number of nodes and communities were performed. Firstly, the quality of the result was measured by the profile fitness measure we described in Sect. 3 and whose results are later shown in Sect. 6. Secondly, a good result was evaluated as being that the communities had a clearly identifiable profile, but a realistic diversity and noise were also present in the attribute values of each record. It was found that obtaining a good dispersion was also dependent on the community size, where smaller communities tended to have homogenous assignments (because a seed would be connected to all members of the community). Later, in Sect. 6, Tables 12, 13, 14 and 15 present the results of the attribute-value distributions for different control parameter sets (dispersion levels) and datasets.

Sensitivity analysis [NS, {SP}, {SD}, {DT}, RU]. A test bed was defined of different combinations (see below) in order to identify the sensitivity of the result to the different parameters of the control set. The result is sensitive to the number of seeds NS, for which the maximum number of seeds possible must be assigned while respecting the assignment rules (see Sect. 4.1). The seed profiles vector (% of each) decides the proportion of each profile to assign. The number of profiles and the degree of fragmentation are sensitive to the result. SD controls the assignment to the neighbors of a seed and thus the propagation so the result is highly sensitive to this. The first value is the most determinant and has a strong effect on the assigned data, that is the homophily of the neighbors attribute values to their corresponding seed's attribute values. DT is not quite so determinant on the data assignment as SD. RU controls the assignment to nodes unassigned after the seed and seed-neighbor assignment. As the seeds and their neighbors should give a good coverage of the whole graph, the number of unassigned seeds should be relatively small and thus the resulting assignment is not so strongly affected by this assignment.

Table 8 Sensitivity of control parameters (1)

Control parameter	Abbrev.	Sensitivity
No. of seeds	NS	1.0
Profile percentages	SP	1.0
Similarity/diversity	SD	0.8
Distances	DT	0.7
Assignment to remaining nodes	RU	0.4

In Table 8, we see an evaluation of the "sensitivity" of each of the control parameters: "sensitivity" means the effect varying this parameter has on the overall outcome of the data assignment. It is measured on a scale of 1.0 (highest) to 0.0 (lowest). For example, the data assignment is very sensitive to the number of seeds assigned in the whole graph (1.0), but the number of seeds is limited by the graph size and topology to give an optimal coverage between a narrow range thus its variability is low. On the other hand, the parameter RU which controls the probability of random assignment to remaining nodes has a low sensitivity and variability, because the number of random nodes is already decided by the number of seeds and their neighbors, and the number of remaining nodes tends to be relatively small.

In Table 9, we see the sensitivity for the three data assignment variations of the control parameter SD. The determinant assignment is the probability that the attribute values of the neighbors of a seed are *identical* to those of the seed. This has quite a wide range of variability depending on the strength of homophily desired between the seed and its neighbors. The other two assignments are dependent on the assignment of *identical*, that is, they share the remaining proportion after the proportion of *identical* is assigned. Hence, their sensibility and variability is lower.

6 Empirical analysis

In this section, we present the statistical evaluation of the generated OSN graphs and associated data. First, in Sect. 6.1 we analyze the data for which RMat has been used to generate the topology. This topology is designed to have the same number of communities as profiles. Each

Table 9 Sensitivity of control parameters (2)

Control parameter	Data assignment	Sensitivity
Similarity/diversity (SD)	Identical	1.0
	Close	0.8
	Not (identical or close)	0.5

user is a member of only one community. Then, in Sect. 6.2 we analyze the data for which three real ground-truth topologies are used. These topologies have a large number of communities (5000) and users can be members of many communities. We note that the "likes" and the "edge weight" attributes have not been included in the analysis of the results due to space restrictions and to maintain clarity. We note that the proportional distributions of the "like" attribute values followed a similar tendency to the other categorical values, as expected. We also confirmed that the "edge weight" values were correlated to the similarity between attribute-value sets of corresponding connected node (user) pairs, as defined in Sect. 5 (iv).

Experimental setup The hardware used was a desktop PC with a 64 bit Intel i5 processor @2.3 Ghz, quad core and 8 Gb RAM; the operating system was Windows 10; for software development and execution, Eclipse Mars release and Java 7 was used.

6.1 RMat-generated topology

In the following, we present the results for the RMat-generated 1 K node graph in which the communities were identified/labeled by the Louvain method.

6.1.1 *Data distributions: matching of profiles to communities*

Figure 8 shows the attribute-value distributions for the whole graph. We see that age is predominantly categories 18–25 and 26–35, gender is equitably distributed, and sexual orientation has a high imbalance. The last attribute (community id) has been defined as the "class value," and thus all other attribute values show the composition with respect to this attribute.

For example, community 2 (light blue) has a high relative proportion of attribute values "gender = female,"

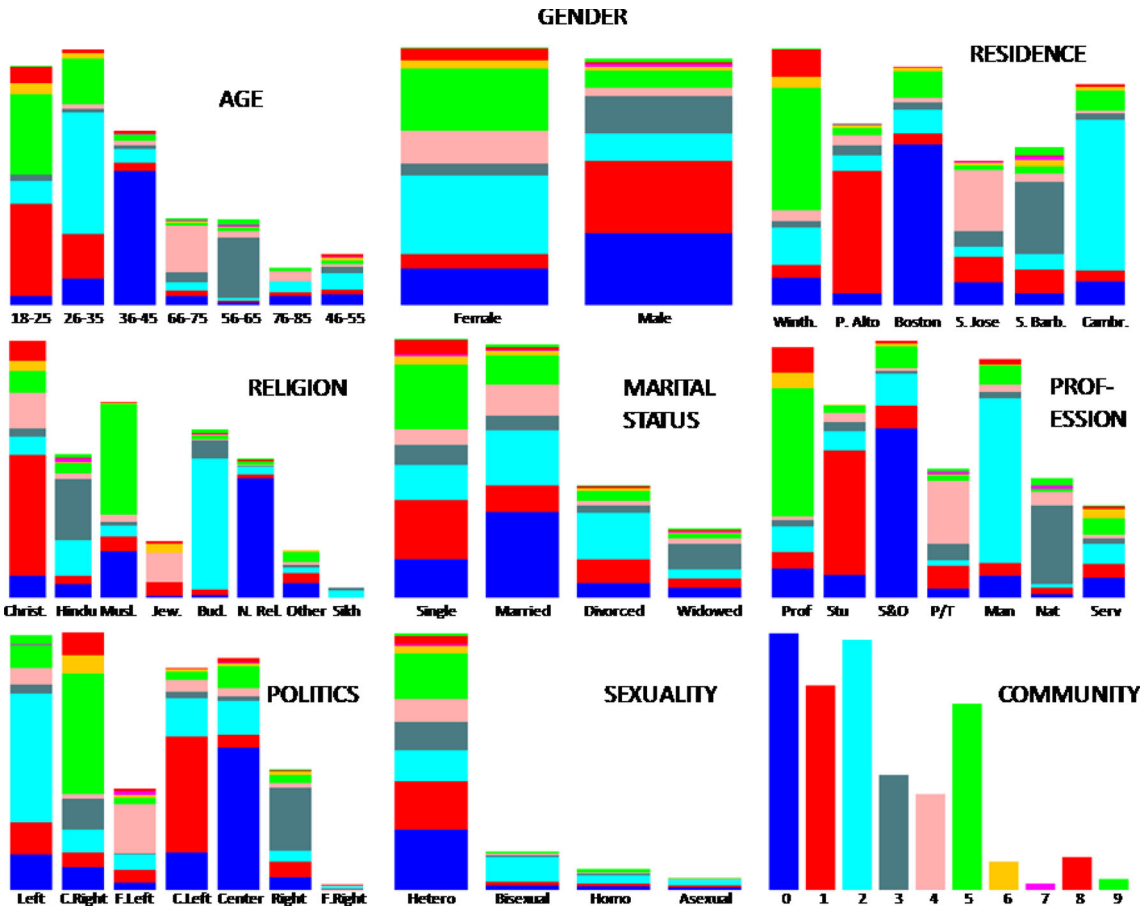

Fig. 8 Distributions of attribute values for the complete graph

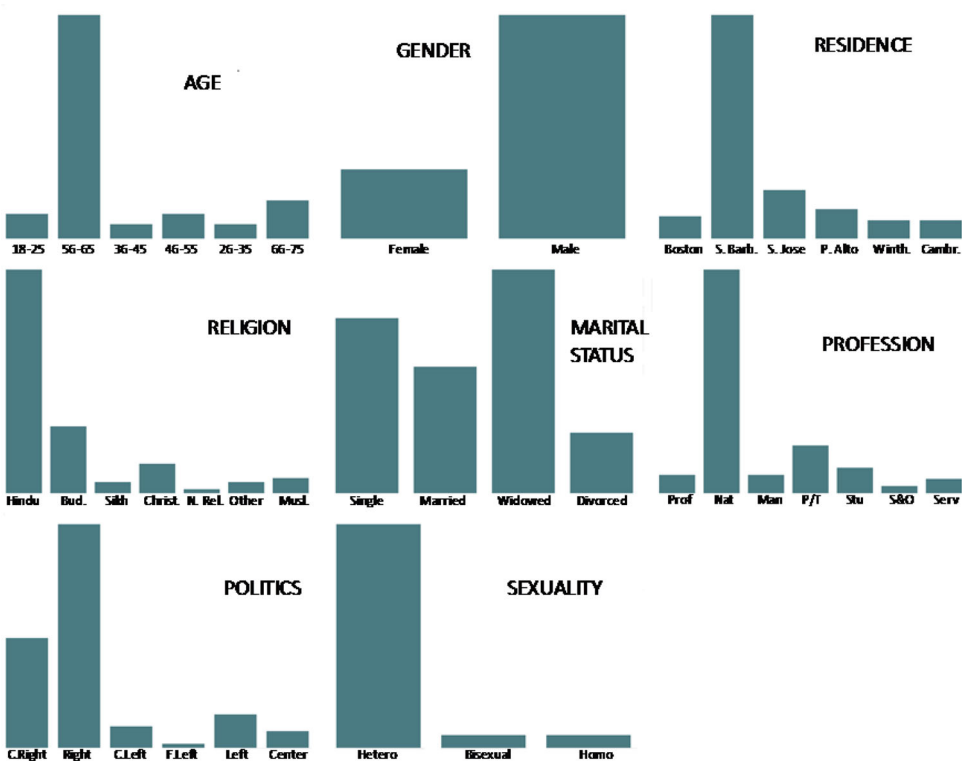

Fig. 9 Distributions of attribute values for Community 3 (which was assigned Profile 4)

"religion = Buddhist," "residence = Cambridge," "profession = Manager," "sexual orientation = bisexual" and "marital status = divorced." If we check the community→profile correspondence, we find that community 2 has been assigned profile 1, and profile 1 includes these attribute value assignments. Likewise, community 3 (gray/blue) has a high proportion of attribute values "gender = male," "religion = Hindu," "residence = Santa Barbara," "profession = Natural…" and "marital status = widow." If we check the community→profile correspondence, we find that community 3 has been assigned profile 4, and profile 4 includes these attribute value assignments. Figure 9 shows the attribute-value distributions only for community 3. The bias of the attribute-value distributions to profile 4 is clearly evident. Hence, we have successfully obtained the desired characteristics for this community. Table 10 shows the distributions for the values of attributes "age," "religion," and "profession" for each community. The bias of the attribute-value distributions to the corresponding profiles (see Tables 3, 4) is clearly evident. Hence, we have successfully obtained the desired characteristics for this community (Table 11).

6.1.2 *Supervised/non-supervised evaluation of matching of profiles to communities*

In Fig. 10, we see the decision tree induced by C4.5 using the community id as the class label. We have compacted the decision tree output by specifying the minimum number of objects as 10, due to space limitations. The overall precision of the model was 65 %, and the precision for individual communities was over 61 % for all communities except C6 to C9 for which the model was unable to build predictive rules. We note that C6 to C9 have a small number of instances relative to the other communities and this causes a class imbalance problem of C4.5. The recall was over 60 % for all communities except C6 to C9.

However, we must emphasize that we are not performing a data mining exercise to build the most precise supervised model possible. Our synthetic data generator purposely "hedges" the primary attribute value (e.g., age="26–35") in a Profile to include a measured amount of similar/close attribute values (e.g., age = "18–25", "36–45"). Also, it introduces a measured amount of noise, that is, attribute values which are neither the "primary" one nor the "close" ones. Hence, the overall precision of 65 % with the medium level dispersion is what we expect. We would expect that with the low dispersion the precision of the C4.5 model would go up and with the high dispersion it would go down. However, this is a trivial consequence of the resulting correlation of the community to the frequency count of the attribute values. We perform a more detailed evaluation of the distributions for different dispersion levels in Sect. 6.2.

With respect to the class imbalance problem, this is minimized for larger datasets. If we do wish to process

Table 10 Distributions for top 3 most frequent attribute values for "age," "religion," and "profession" (by community/profile)

Community	Assigned profile	Age	Religion	Profession	No. of instances in community
0	0	36–45, 26–35, 46–55[a]	No Rel, Muslim, Christian	Sales & Off, Prof, Student	216
		{0.66, 0.19, 0.15}[b]	{0.54, 0.31, 0.15}	{0.61, 0.19, 0.20}	
1	2	18–25, 26–35, 36–45	Christian, Muslim, Jewish	Student, Sales & Off, Prod/Trans	172
		{0.57, 0.33, 0.10}	{0.69, 0.16, 0.16}	{0.56, 0.21, 0.23}	
2	1	26–35, 18–25, 46–55	Buddhist, Hindu, Christian	Manager, Sales & Off, Prof	211
		{0.62, 0.20, 0.18}	{0.61, 0.25, 0.14}	{0.61, 0.21, 0.18}	
3	4	56–65, 66–75, 46–55	Hindu, Buddhist, Christian	Nat Rec, Prod/Trans, Student	97
		{0.66, 0.19, 0.15}	{0.62, 0.27, 0.11}	{0.63, 0.21, 0.16}	
4	5	66–75, 76–85, 56–65	Christian, Jewish, Muslim	Prod/Trans, Nat Rec, Student	81
		{0.62, 0.22, 0.16}	{0.43, 0.44, 0.12}	{0.60, 0.22, 0.17}	
5	3	18–25, 26–35, 36–45	Muslim, Christian, Hindu	Prof, Sales & Off, Manager	157
		{0.55, 0.36, 0.10}	{0.69, 0.20, 0.11}	{0.64, 0.20, 0.16}	
6	7	18–25, 26–35, 46–55	Jewish, Christian, Other Rel	Prof, Service, Sales & Off	24
		{0.50, 0.38, 0.13}	{0.46, 0.42, 0.13}	{0.50, 0.38, 0.13}	
7	9	56–65, 36–45, 46–55	Hindu, Buddhist	Nat Rec, Prod/Trans	5
		{0.40, 0.40, 0.20}	{0.80, 0.20, 0.00}	{0.60, 0.40, 0.00}	
8	6	18–25, 26–35, 36–45	Christian, Jewish, No Rel	Prof, Manager, Sales & Off	28
		{0.64, 0.25, 0.11}	{0.71, 0.18, 0.11}	{0.71, 0.21, 0.07}	
9	8	56–65, 66–75, 18–25	Hindu, Buddhist, No Rel	Nat Rec, Prod/Trans, Service	9
		{0.67, 0.33, 0.00}	{0.44, 0.56, 0.00}	{0.67, 0.33, 0.00}	

[a] Top 3 categories

[b] % top category, % 2nd and 3rd categories, % all other categories

Table 11 Ground Truth dataset statistics (5000 top communities)

Dataset name	Nodes	Edges	No. of users per community: range (Avg.)	No. of communities per user: range (Avg.)
Amazon	14,771	87,322	2–327 (177.51)	1–1614 (56.84)
YouTube	39,841	448,470	2–2217 (14.59)	1–54 (1.83)
LiveJournal	84,438	3,043,040	3–1441 (27.8)	1–20 (1.64)

small communities, different data mining solutions exist for class imbalance, one of which is "boosting."

An example of interpretation of the tree in Fig. 10 would be as follows: If religion = "Christian" and residence = "Palo Alto," then community = 1 with a confidence level of $89/(89 + 10) = 90$ %. If we reference Table 5, we see that community 1 was assigned Profile 2; then, if we reference Table 4, we see that Profile 2 was defined as being Christians living in Palo Alto. Another example would be: If religion="Christian" and residence = "Winthrop" and age="18–25," then community = 8 with a confidence level of $22/(22 + 12) = 65$ %. If we reference Table 5, we see that community 8 was assigned Profile 6; again, if we reference Table 4, we see that Profile 6 was defined as being Christians living in Winthrop whose age is in the range 18–25.

We also ran Kmeans on the dataset, using the community id for a class to cluster evaluation, with the number of clusters set to 10, which gave 62 % correctly clustered instances. Finally, we applied the Weka attribute selection method "InfoGain" with the Ranker option to the dataset, which ranked the attributes in the following order, with respect to the community id: religion, profession, age, residence, political orientation, gender, marital status, sexual orientation.

6.2 Real topologies: ground-truth community datasets

Instead of using RMat to generate the topology, we will now use several real topologies which represent "ground-truth" communities obtained from the SNAP online

```
religion = Christian
|   residence = Winthrop
|   |   age = 18-25: 8 (22.0/12.0)
|   |   age = 26-35: 5 (12.0/8.0)
|   |   age = 36-45: 0 (6.0/3.0)
|   |   age = 66-75: 0 (1.0)
|   |   age = 56-65: 4 (1.0)
|   |   age = 76-85: 4 (1.0)
|   |   age = 46-55: 8 (4.0/1.0)
|   residence = Palo Alto: 1 (89.0/10.0)
|   residence = Boston
|   |   maritalstatus = Single: 5 (11.0/7.0)
|   |   maritalstatus = Married: 0 (10.0/6.0)
|   |   maritalstatus = Divorced: 1 (4.0/2.0)
|   |   maritalstatus = Widowed: 4 (1.0)
|   residence = San Jose
|   |   profession = Professional: 1 (4.0/2.0)
|   |   profession = Student: 1 (11.0/3.0)
|   |   profession = Sales and office: 4 (0.0)
|   |   profession = Production transportation and material moving: 4 (22.0/2.0)
|   |   profession = Manager: 2 (3.0/2.0)
|   |   profession = Natural resources construction and maintenance: 4 (4.0/1.0)
|   |   profession = Service: 1 (2.0)
|   residence = Santa Barbara: 1 (18.0/8.0)
|   residence = Cambridge: 2 (25.0/15.0)
religion = Hindu
|   profession = Professional: 2 (16.0/11.0)
|   profession = Student: 1 (11.0/8.0)
|   profession = Sales and office: 0 (16.0/10.0)
|   profession = Production transportation and material moving: 3 (11.0/5.0)
|   profession = Manager: 2 (29.0/11.0)
|   profession = Natural resources construction and maintenance: 3 (51.0/8.0)
|   profession = Service: 2 (7.0/4.0)
religion = Muslim
|   profession = Professional: 5 (86.0/10.0)
|   profession = Student: 1 (15.0/9.0)
|   profession = Sales and office: 0 (44.0/15.0)
|   profession = Production transportation and material moving: 0 (9.0/6.0)
|   profession = Manager: 5 (18.0/10.0)
|   profession = Natural resources construction and maintenance: 3 (6.0/4.0)
|   profession = Service: 5 (14.0/7.0)
religion = No religious affiliation: 0 (137.0/20.0)
```

Fig. 10 C4.5 Pruned Tree. Complete dataset with community id as the classifier label (some level 1 nodes have been removed for brevity)

repository (https://snap.stanford.edu/data/). In the previous section, we recall that we generated the topology using RMat and assigned the Communities using the Louvain algorithm. Now we do not need to generate the topology because it is a real one from real OSN apps (Amazon, YouTube and LiveJournal). Also, we do not need to assign the communities because they are also real, calculated by online group membership in the corresponding apps. In this section, we also benchmark the three different control parameter sets and evaluate the results. The control parameter sets correspond to three "dispersion" levels for the data: level 1 (low), level 2 (medium) and level 3 (high). We recall that in Sect. 6.1, we generated the data for the RMat topology using the level 2 dispersion level.

The Amazon product co-purchasing network and ground-truth communities dataset was collected by crawling the Amazon website (Yang and Leskovec 2012). It is based on Customers Who Bought This Item Also Bought feature of the Amazon website. If a product i is frequently co-purchased with product j, the graph contains an undirected edge from i to j. Each product category provided by Amazon defines each ground-truth community.

YouTube social network and ground-truth communities (Yang and Leskovec 2012). YouTube is a video-sharing Web site that includes a social network. In the YouTube social network, users form friendship each other and users can create groups which other users can join. We consider such user-defined groups as ground-truth communities. These data are provided by the reference given in Mislove et al. (2007) and is available at: http://socialnetworks.mpi-sws.mpg.de. In Figs. 11 and 12, we see the characteristic "long-tail" distribution of the number of communities per user and the number of users per community, respectively.

LiveJournal social network and ground-truth communities (Yang and Leskovec 2012). LiveJournal is a free online blogging community where users declare friendship each

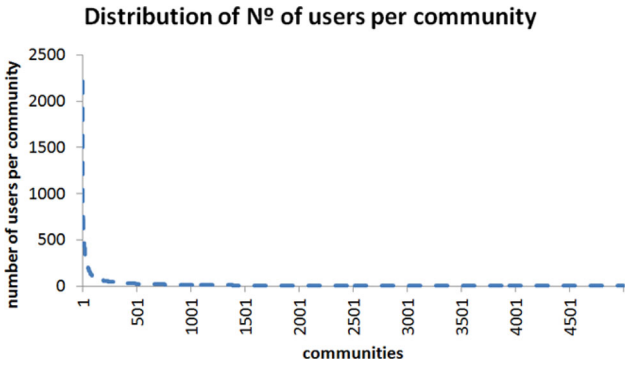

Fig. 11 Distribution of the number of users per community (YouTube dataset)

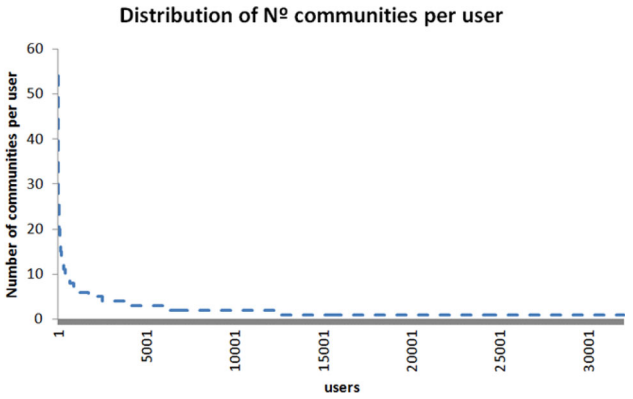

Fig. 12 Distribution of the number of communities per user (YouTube dataset)

other. LiveJournal also allows users form a group which other members can then join. These user-defined groups are considered as ground-truth communities. The authors have made the data (LiveJournal friendship social network and ground-truth communities) available in the SNAP online repository.

For each dataset, as described in (Yang and Leskovec 2012), each connected component in a group is considered as a separate ground-truth community. The ground-truth communities which have less than 3 nodes were removed. The datasets corresponding to the top 5000 communities with highest quality, according to the metrics described in (Yang and Leskovec 2012), were used in the present work.

6.2.1 *Data processing approach for ground-truth communities*

These datasets present a different scenario to the RMat synthetic dataset and communities presented in Sect. 6.1. Firstly, the ground-truth datasets have a much higher number of communities (over 5000) whose size presents a long-tail distribution and whose average size is much smaller. Secondly, a user can be a member of many communities. This scenario requires a rethink of the seed assignment, profile assignment and data propagation which we described in Sect. 4.

As before, we first try to assign a maximum number of seeds in each community. Then we assign the profiles to the seeds. We recall that a profile is a set of attribute values such as those defined in Table 4.

In contrast to previously, when we had the same number of profiles to assign as communities, now we will have a fixed number of profiles (those shown in Table 4) and will assign a profile to each community based on a probability distribution. That is, each profile has a probability of being assigned between 0 and 1. Each community will have one profile assigned. However, we must take into account that a node which is a seed may also be present in many other communities. In some communities, it may also be a seed and in others, not. Hence, once a node (seed, in this case) is assigned a profile, it will have the same profile in all communities it is present.

Once the seeds are assigned, for each community we propagate, as before, from the seed to its immediate neighbors. Finally, as before, we assign the unassigned nodes. However, for all non-seed nodes we also have to consider they may also be present in many communities. Thus, once a node is assigned with a profile, that profile is the same for all communities in which the node is present. To facilitate this, when we assign a node for the first time, we check its community list and flag the node as assigned in all those communities. Thus the community assignment is initially lineal but then propagates out into common communities.

6.2.2 *Results for ground-truth communities*

The results are presented for each dataset. For each dataset, we show the overall statistics for each attribute in terms of top modal values and dispersion. This enables us to compare the generated data with the profile definitions.

In Table 12, we see the distribution statistics of the complete Amazon synthetic dataset for the three levels of dispersion as we defined previously in Sect. 5. We see that the relative percentages of the most frequent attribute values remains very constant for greater dispersion levels. This is the desired effect. We wish to maintain the overall proportions as defined in the profiles (Table 4) for the whole dataset. We will see later in Table 15 that the dispersion acts effectively and clearly at a community level. Returning to Table 12, we see, for example, that the percentage of individuals in the dataset who have an age of 18–25 years varies between 26 % (level 3) and 29 % (level 1). If we look at Table 4, we will see that profiles 2, 3, 6, and 7 have this age group value. The sum of the target proportions of these profiles is $17.2 + 15.7 +$

Table 12 Amazon—complete dataset

Level of dispersion	Age	Gender	Residence	Religion	Marital	Profession	Politics	Sexuality
Level 1	18–25, 26–35, 36–45[a]	Male, Female	Boston, Palo Alto, San Jose	Christian, No Rel, Buddhist	Married, Single, Divorced	Sales & Off, Student, Prod/ Trans	Center Left, Left, Center	Hetero, Bisex, Homo
	{0.29, 0.40, 0.31}[b]	{0.54, 0.46}	{0.20, 0.38, 0.42}	{0.29, 0.28, 0.43}	{0.39, 0.51, 0.10}	{0.20, 0.35, 0.46}	{0.22, 0.41, 0.37}	{0.84, 0.14, 0.02}
Level 2	18–25, 26–35, 36–45	Male, Female	Boston, Palo Alto, San Jose	Christian, No Rel, Hindu	Married, Single, Divorced	Sales & Off, Student, Prof	Center, Left, Center Left	Hetero, Bisex, Homo
	{0.27, 0.40, 0.32}	{0.58, 0.42}	{0.22, 0.34, 0.44}	{0.28, 0.32, 0.41}	{0.39, 0.50, 0.11}	{0.22, 0.33, 0.45}	{0.22, 0.42, 0.36}	{0.81, 0.16, 0.03}
Level 3	18–25, 26–35, 36–45	Male, Female	Palo Alto, Boston, Cambridge	Christian, Buddhist, Hindu	Married, Single, Divorced	Sales & Off, Student, Manager	Left, Center Left, Center	Hetero, Bisex, Homo
	{0.26, 0.41, 0.33}	{0.55, 0.45}	{0.20, 0.36, 0.44}	{0.29, 0.31, 0.40}	{0.36, 0.51, 0.13}	{0.20, 0.36, 0.45}	{0.24, 0.40, 0.36}	{0.73, 0.23, 0.04}

[a] Top 3 categories

[b] % top category, % second and third categories, % all other categories

Table 13 YouTube—complete dataset

Level of dispersion	Age	Gender	Residence	Religion	Marital	Profession	Politics	Sexuality
Level 2	18–25, 26–35, 36–45[a]	Male, Female	Palo Alto, Boston, Cambridge	Christian, Muslim, No Rel	Married, Single, Divorced	Student, Sales & Off, Prof	Left, Center Left, Center,	Hetero, Bisex, Homo
	{0.28, 0.42, 0.30}[b]	{0.55, 0.45}	{0.19, 0.35, 0.46}	{0.28, 0.30, 0.42}	{0.37, 0.53, 0.10}	{0.20, 0.35, 0.45}	{0.24, 0.40, 0.36}	{0.79, 0.18, 0.03}

[a] Top 3 categories

[b] % top category, % second and third categories, % all other categories

$2.8 + 2.4 = 38.1$ %. Also, we see that the percentage of individuals in the dataset who are female is between 42 and 45 %. Again, if we look at Table 4, we see that profiles 1, 3, 5, 6, 7 have this gender value. The sum of the target proportions of these profiles is $21.1 + 15.7 + 8.1 + 2.8 + 2.4 = 50.1$. If we apply the same procedure to the religion attribute, we see in Table 12 that Christian is 28–29 %, and in Table 4 the sum of the proportions of the profiles (2 and 6) which have that this religion value is $17.2 + 2.8 = 20.0$ %.

In terms of the overall proportions attribute values in the complete dataset, it is much easier to maintain these proportions for non-overlapping communities; we recall that we are assigning profiles to communities. However, we recall that in the ground-truth graphs, the number of users per community and the number of communities per user has a "long-tail" distribution. Thus, in order to obtain the best fit of profile assignments to communities in the desired proportions, we assign the communities in decreasing order of size. It may occur that the first community by size has a high overlap with other (smaller communities) which may skew the overall proportions. However, this would represent the real structure of the graph so we could say the data assignment would be correct in reflecting this structure. Nevertheless, in spite of these difficulties, we can see that the attribute values and the profiles themselves are quite well distributed and reasonably dimensioned with respect to the initial data definitions.

In general, from Table 12 we see that the profile which appears most frequently in proportional terms is {18–25, Male, Boston, Christian, Married, Sales & Off, Center Left, Hetero}. If we then refer to Table 4, we see that these attribute values are the same or are close to those which are assigned to the profiles with the highest target proportions.

In Tables 13 and 14, which show the level 2 dispersion proportions for the YouTube and LiveJournal "ground-truth" graph datasets, we see similar trends emerging to those we have just commented for Table 12. In Tables 13 and 14, we only show level 2 dispersion for brevity and because there is not a great variation in distributions

Table 14 LiveJournal—complete dataset

Level of dispersion	Age	Gender	Residence	Religion	Marital	Profession	Politics	Sexuality
Level 2	18–25, 26–35, 36–45[a]	Male, Female	Boston, Palo Alto, Winthrop	Christian, Muslim, No Rel	Married, Single, Divorced	Sales & Off, Student, Prof	Left, Center Left, Center,	Hetero, Bisex, Homo
	{0.28, 0.41, 0.31}[b]	{0.54, 0.46}	{0.19, 0.35, 0.46}	{0.30, 0.29, 0.41}	{0.40, 0.50, 0.10}	{0.20, 0.34, 0.46}	{0.24, 0.39, 0.37}	{0.82, 0.15, 0.03}

[a] Top 3 categories

[b] % top category, % second and third categories, % all other categories

between Levels, for the reasons we have also explained previously, this being a desirable overall property. The results of Tables 13 and 14, for the YouTube and LiveJournal show that the data assignment process can successfully assign data to significantly different topological graph structures, graph sizes, number of communities and community overlap.

In Table 15, we show a different scenario to that of Tables 12, 13 and 14. In Table 15, we see the dispersion proportions for specific communities, profiles and dispersion levels. We recall that the dispersion is designed to act at a community level, because the data assignment process tries to assign profiles (those of Table 4) to individual communities. However, this process becomes more complex (and realistic) when we have a complex overlap of user assignment to communities. That is, a user can belong to many communities. We also recall that we assign the data profiles to individual users in a community, and once a user is assigned a profile, this profile is assigned for all the communities in which that user is a member.

The first three rows of Table 15 show the assignment proportions for the attribute values for the Amazon dataset, for Profile 0 and for the three levels of dispersion. If we refer to the definition of Profile 0 in Table 4, we see that the top attribute values (those which have the highest proportion) assigned in Table 15 are indeed the same ones as defined for Profile 0. That is, the data assignment process has chosen Profile 0 as the one to be assigned to this community. If we now compare the dispersion statistics for Levels 1, 2 and 3 (rows 1 to 3), we see that as the dispersion level increases, in general, the top attributes' proportion decreases, and the proportions of the second and third categories, and all other categories, increase. This is what we mean by dispersion. We see that the control parameters are, in general, influencing the dispersion level as expected, for the attribute values. However, we also see that there is not always a direct correlation of the change in proportion with dispersion levels 1 and 2. There are two exceptions: see Table 15, attribute "profession" for Amazon profile1 and YouTube profile2. As mentioned previously, for the overlapping communities another process is acting: individuals who are members of many communities may have a greater influence on the attribute-value assignment in a community, as this may bias the proportions in a given community. This is the reason there is not always perfect correlation with the dispersion level. And this reflects the realistic "ground-truth" overlapping graph structure within which we assign the data. There is also the difficulty in matching communities in graph datasets each one of which is generated by different levels (different control parameter sets). In practice, this was performed by ordering the dataset by community id and key attributes, and performing a manual inspection of the records.

However, taking into account these difficulties, the results of Table 15 do in general show a significant dispersion change between levels 1, 2 and 3. For example, for Amazon (profile 0) attribute-value "age = 35–45" has a proportion of 79 % for level 1 (the least disperse), a proportion of 74 % for level 2 and a proportion of 40 % for level 3 (the most disperse). Likewise, for YouTube (profile 2), attribute-value "residence = Palo Alto" has a proportion of 83 % for level 1, a proportion of 50 % for level 2, and a proportion of 35 % for level 3.

6.2.3 *Summary of benchmarking for overlapping ground-truth communities*

In this section, we summarize the results for execution time, fitness of attributes (level 1 dispersion) with respect to target proportions of the values for the whole graph, and fitness of profiles with respect to their target proportions for the whole graph.

From Table 16, we see that the processing time is practically linear with the graph size in terms of nodes and also with the number of seeds. The seed assignment represents between 82 and 87 % of the total processing time.

In Table 17, we have calculated the average fit (Eq. 4, Sect. 3) of each attribute to its target distributions (of each of its possible values) in the whole dataset, averaged for its corresponding attribute values. From Table 17, we see that the fit for the attributes is attribute dependent. This is designed in the propagation rules in which homophily is

Table 15 All datasets—dispersion levels 1 to 3—selected individual communities

Level of dispersion	Age	Gender	Residence	Religion	Marital	Profession	Politics	Sexuality
Amazon (profile 0) level 1	36–45, 26–35, 18–25[a] {0.79, 0.17, 0.05}[b]	Male, Female {0.91, 0.09}	Boston, Winthrop, Palo Alto, {0.76, 0.10, 0.14}	No Rel, Christian, Hindu {0.77, 0.16, 0.07}	Married, Single, Divorced {0.90, 0.10, 0.00}	Sales & Off, Prof, Service {0.72, 0.16, 0.12}	Center, Left, Center Left {0.72, 0.16, 0.12}	Hetero, Bisex, Homo {0.93, 0.07, 0.00}
Amazon (profile 0) level 2	36–45, 26–35, 46–55 {0.74, 0.19, 0.07	Male, Female {0.76, 0.24}	Boston, Santa B., Palo Alto, {0.72, 0.16, 0.12}	No Rel, Christian, Hindu {0.74, 0.19, 0.07}	Married, Single, Widowed {0.77, 0.21, 0.02}	Sales & Off, Student, Manager {0.69, 0.14, 0.17}	Center, Left, Center Left {0.67, 0.21, 0.12}	Hetero, Bisex, Homo {0.88, 0.12, 0.00}
Amazon (profile 0) level 3	36–45, 66–75, 56–65 {0.40, 0.48, 0.12}	Male, Female {0.56, 0.44}	Boston, San Jose, Winthrop {0.40, 0.40, 0.21}	No Rel, Jewish, Christian {0.35, 0.48, 0.17}	Married, Single, Widowed {0.63, 0.29, 0.08}	Sales & Off, Prod/Trans, Prof {0.38, 0.40, 0.23}	Center, Left, Far Left {0.48, 0.38, 0.15}	Hetero, Homo, Bisex {0.79, 0.15, 0.06}
Amazon (profile 1) level 1	26–35, 18–25, 36–45 {0.75, 0.19, 0.06}	Male, Female {0.09, 0.91}	Cambridge, Palo Alto, Boston {0.75, 0.19, 0.06}	Buddhist, Hindu Christian {0.63, 0.38, 0.00}	Divorced, Married, Single {0.50, 0.50, 0.00}	Manager, Prod/Trans, Sales & Off {0.55, 0.40, 0.05}	Left, Center Left, Far Left {0.75, 0.25, 0.00}	Bisex, Hetero {0.57, 0.43, 0.00}
Amazon (profile 1) level 2	26–35, 18–25, 56–65 {0.58, 0.22, 0.20}	Male, Female {0.12, 0.88}	Cambridge, Santa B., San Jose {0.60, 0.31, 0.10}	Buddhist, Muslim, Christian {0.60, 0.36, 0.05}	Divorced, Single, Married, {0.50, 0.44, 0.06}	Manager, Service, Sales & Off {0.63, 0.31, 0.06}	Left, Center Left, Center {0.62, 0.27, 0.11}	Hetero, Bisex, Asexual {0.53, 0.47, 0.00}
Amazon (profile 1) level 3	26–35, 66–75, 18–25 {0.55, 0.40, 0.05}	Male, Female {0.25, 0.75}	Cambridge, Palo Alto, Boston {0.49, 0.29, 0.22}	Buddhist, Christian, Jewish {0.58, 0.29, 0.13}	Divorced, Married, Single {0.49, 0.47, 0.04}	Manager, Prod/Trans, Nat Rec {0.53, 0.33, 0.13}	Left, Far Left, Center Left {0.55, 0.43, 0.02}	Bisex, Hetero, Homo {0.44, 0.50, 0.06}
YouTube (profile 2) level 1	18–25 {1.00, 0.00, 0.00}	Male, Female {0.93, 0.07}	Palo Alto, Cambridge {0.83, 0.17, 0.00}	Christian, Muslim {0.83, 0.17, 0.00}	Single {1.00, 0.00, 0.00}	Student, Manager {0.83, 0.17, 0.00}	Center Left, Far Left {0.83, 0.17, 0.00}	Hetero, Homo {0.83, 0.47, 0.00}
YouTube (profile 2) level 2	18–25, 26–35, 66–75 {0.64, 0.36, 0.00}	Male, Female {0.83, 0.17}	Palo Alto, Cambridge, Santa B {0.50, 0.43, 0.07}	Christian, Buddhist, Muslim {0.57, 0.29, 0.14}	Single, Married, Widowed {0.62, 0.32, 0.06}	Student, Manager, Prof {0.57, 0.29, 0.14}	Center Left, Right, Center {0.57, 0.36, 0.07}	Hetero, Homo {0.79, 0.21, 0.00}
YouTube (profile 2) level 3	18–25, 26–35, 36–45 {0.30, 0.49, 0.22}	Male, Female {0.41, 0.59}	Palo Alto, Boston, Santa B. {0.35, 0.41, 0.24}	Christian, Buddhist, No Rel {0.46, 0.46, 0.08}	Single, Married, Widowed {0.57, 0.43, 0.00}	Student, Sales & Off, Prof {0.59, 0.22, 0.19}	Left, Center Left, Center {0.43, 0.49, 0.08}	Hetero, Homo {0.62, 0.38, 0.00}

[a] Top 3 categories

[b] % top category, % second and third categories, % all other categories

Table 16 Benchmarking: execution time

Graph	Nodes	Seeds	Seed assignment (s)	Data assignment (s)
RMat	1000	110	0.45	0.08
Amazon	15,000	5000	71	11
LiveJournal	84,000	12,000	295	64

Table 17 Fit of target attribute distributions to assigned attribute distributions (level 1 dispersion)

Graph	Attributes								Avg.
	Age	Gender	Residence	Religion	Marital	Profession	Politics	Sexuality	
YouTube	81.4[a]	88.1	77.6	72.3	82.9	71.4	78.1	76.3	78.5
	{0.444}**	{0.342}	{0.387}	{0.257}	{0.415}	{0.238}	{0.399}	{0.026}	
Amazon	75.2	87.3	67.5	65.9	68.4	61.5	64.2	71.5	70.2
	{0.270}	{0.303}	{0.161}	{0.134}	{0.077}	{0.074}	{0.108}	{0.004}	
LiveJournal	81.9	86.7	73.2	69.4	78.3	68.1	75.3	76.5	76.2
	{0.452}	{0.277}	{0.278}	{0.197}	{0.278}	{0.172}	{0.328}	{0.028}	
Average									74.9

** p value

[a] Note that level 1 dispersion introduces approx. 20 % noise into the dataset, thus reducing the fit to the target in approximately this amount

Table 18 Fit of target profile distributions to assigned profile distributions (level 1 dispersion)

Graph	Profiles										Avg.
	0	1	2	3	4	5	6	7	8	9	
YouTube	83.1[a]	66.8	86.6	73.4	78.9	78.2	83.7	79.2	82.1	87.4	79.9
	{0.466}**	{0.092}	{0.628}	{0.301}	{0.558}	{0.581}	{0.823}	{0.784}	{0.889}	{0.944}	
Amazon	68.5	61.1	77.4	65.8	69.4	68.1	71.7	71.5	77.3	81.2	71.2
	{0.112}	{0.034}	{0.372}	{0.147}	{0.349}	{0.371}	{0.664}	{0.686}	{0.853}	{0.912}	
LiveJournal	79.4	65.2	85.7	71.2	77.3	75.1	78.4	77.5	81.5	83.5	77.5
	{0.356}	{0.072}	{0.602}	{0.251}	{0.522}	{0.516}	{0.756}	{0.763}	{0.885}	{0.925}	
Average											76.2

** p value

[a] Note that level 1 dispersion introduces approx. 20 % dispersion into the dataset, thus reducing the fit to the target in approximately this amount

strongest for gender (86–88 %) and weakest for profession (61–71 %). We see that amazon has consistently lower fitness than YouTube and LiveJournal for all attributes. If we go back and take a look at Table 11, we see that the Amazon graph dataset has a much higher number of users per community and number of communities per user (multiple membership) than the other two datasets. We recall that multiple membership makes it more difficult to obtain a deterministic assignment of a given profile to a given community, and this is reflected in the relatively lower fitness values for the Amazon dataset as shown in Table 17. The average distribution fits are 78.5 % for YouTube, 70.2 % for Amazon and 76.2 % for LiveJournal. However, we note that approx. 20 % noise has been intentionally introduced during the data assignment process (level 1 dispersion). Also in Table 17, we see the p-values calculated for each attribute and dataset. The p value gives quantified value for the validity of the null hypothesis, which in this case is that two sets of points are the same. We note that each point represents an attribute-value proportion for a given attribute, and the two sets represent the target and the generated proportions. Thus in this case a p-value greater or equal to 0.05 means that the null hypothesis cannot be rejected at the 95.0 % confidence level. As we see from Table 17, all attributes/datasets have $p \geq \alpha$, where $\alpha = 0.05$, except for the "sexuality" attribute. To calculate the p-value, the "Statgraphics" software was used with the "multiple sample comparison" option.

In Table 18, we have calculated the average fit of a profile to its target distribution in the whole dataset. That is, if profile 0 corresponds to age = 36–45, gender = male and residence = Boston, then we have calculated the

proportion of these attribute values in the communities to which profile 0 is a target. From Table 18, we see that the fit for profiles 0 and 2 is between 68 and 86 %. Again we see that the Amazon fit statistics are consistently lower than those of the other two datasets due to the higher multiple community membership of this dataset. The average distribution fits are 79.9 % for YouTube, 71.2 % for Amazon and 77.5 % for LiveJournal. However, we note that approx. 20 % noise has been intentionally introduced during the data assignment process (level 1 dispersion). In Table 18, the p-values have also been calculated for each profile and dataset. In this case, to calculate the p value, the "Statgraphics" software was used with the "compare→two samples→hypothesis test" function and the "binomial proportions," "use z-score" options and alpha = 0.05, and null hypothesis "difference between proportions = 0.0." The null hypothesis is that the two proportions (the target and the generated) are the same. Thus in this case a p-value greater or equal to 0.05 means that the null hypothesis cannot be rejected at the 95.0 % confidence level. We see that all p-values are superior or equal to 0.05, with the exception of Profile 1.

6.3 Discussion

In this section, we first consider general issues, followed by scalability of our approach, its algorithmic complexity, advantages, and comparison with other approaches.

6.3.1 General issues

This work presents a series of challenges which we will now comment: (1) A high degree node chosen as a seed may have a disproportionate influence on the network. This is mitigated by the use of medoid values based on the centrality metric or degree; (2) The restrictions on the placement of the seed nodes and the topology of the communities may cause a significant percentage of nodes to have random assignments (coverage); (3) it is easy to make non-overlapping communities representative of key attribute-value profiles, but we also need diversity on different attribute values within communities and realistic overlap between communities; (4) obtaining "ground truth" not just for the topology but also for the data assignments.

With respect to the attribute set chosen in the current work, some demographic attributes and their categories are standard, such as age, gender, religion, marital status, sexuality. On the other hand, attributes such as profession and especially residence will probably require customization. Application-specific and activity-related information, such as "likes" and edge weights, can also be adapted by the user of the data simulator. However, changing the attribute values or introducing new attributes will require the adaption of existing control parameter rules or the creation of new rules. As we see in Sect. 5, each attribute and attribute-value set requires its own customized rule.

In order to make the results comparable in the benchmarking, we have used the same set of attribute values for all topologies. In future work, we could experiment with specific attribute values for each ground-truth topology, for example, a classification of videos in the case of YouTube, blog keywords for LiveJournal or purchased product categories in the case of Amazon. In fact, one of the future benefits of our approach will be the ability to populate any published ground-truth topology with customized data. With respect to data validation, one approach could be the sampling of real OSN data from these applications, in order to compare the attribute-value distributions and intra-community correlations with the synthetically generated data. Indeed, this approach could be used to "fine tune" the synthetic data for a given "ground-truth" topology.

We have seen that the high number of overlapping communities in the ground-truth datasets presents a big challenge for the data assignment. We recall that in Sect. 6.1 we had non-overlapping communities which were significantly simpler to process than the overlapping ones of Sect. 6.2, and the resulting profile to community assignments fitted better because there was no inter-community interference. That is, although we assign a profile N to the seed nodes in a community A, some nodes which are not seeds in community A may be seeds in another community B, assigned with profile N'. Hence, the predominance of profile N in community A may be indirectly challenged by profile N'. As future work, we will evaluate this situation in more detail. However, we propose this gives us realistic data because it is representing the overlap of the ground-truth communities, and the communities will vary from being more homogeneous to more heterogeneous in nature.

6.3.2 Scalability of the approach

Although the parallelization of the method is outside the scope of the current work, we propose that the design allows for a scalable implementation. Specifically, the MapReduce (Dean and Sanjay 2008) could be applied in which a Map process subdivides the dataset into parts which can be processed independently and the reduce process collates the intermediate results of the individual processes to produce a final output (or an intermediate output between iterations). For synthetic topology generation, we can apply a parallel version of RMat and Louvain, of which several implementations exist (see below).

Then for the seed assignment and data propagation a scheme similar to parallel kMeans can be applied (Zhao et al. 2009). Zhao et al. presented a parallelization of

k-Means using MapReduce, in which it is stated that for the k-Means algorithm, the main computational cost involves the distance calculations. For each iteration, nk distance computations are performed, where n is the number of objects and k is the number of clusters being formed. Within a given cluster, the distance calculation between an object and the center is independent of the same calculation in another cluster. Hence, distance computations for different clusters can be executed in parallel. However, in each iteration, the new centers to be used in the next iteration must be updated. This makes it necessary for each iteration to be executed serially. Information is passed to the next iteration in the form of a vector of the new cluster centers calculated in the previous iteration. Within each iteration, the clusters are processed in parallel, depending on the number of physical processors (e.g., CPU cores) available. For example, if there are C clusters (processes) and N processors, then C/N processes will be assigned to each processor.

This approach can be adapted to our method, in which the communities will take the place of the k-Means clusters. That is, a degree of parallelism of C/N can be achieved, where C is the number of communities and N is the number of assignable processes. In terms of the parallelization of the topology and data generation process, if an existing topology and community labeling is used (as in Sect. 6.2) the cost will be zero. For building a topology and community labeling from scratch, parallel versions (including apis) of RMat (Plimpton and Devine 2011) and the Louvain method (Dean and Sanjay 2008) (Ovelgonne 2013) (Que et al. 2013) have been published and made available.

Once we have a topology and community labeling, the following step is the seed assignment. The following steps are considered for which computational cost may be an issue: (1) seed assignment and (2) similarity calculation for data propagation. (1) Seed assignment involves obtaining a good coverage of seeds in the overall graph and to each community, while complying with restrictions of inter-seed distance (>2) and the proportion of a seed's neighborhood which falls within the seed's community. Both these values are user defined. Within a given community, the distance calculation (that is, the number of links) between a seed and all other assigned seeds in the same community is independent of the same calculation in another community. For each iteration of the whole graph, $O(S^2C)$ distance calculations are performed, where S is the average number of seeds per community and C is the number of communities. Communities can be processed in parallel, as has been previously explained, with information passing between successive iterations. (2) The next step, the similarity calculation for data propagation, involves a distance computation for each neighbor of each seed. That is, the profile assigned to a neighbor must be within a given distance to the profile of the seed. For each iteration of the whole graph, O(NS x AN) distance computations are performed, where NS is the number of seeds and AN is the average number of neighbors per seed. Again, distance computations for different communities can be executed in parallel, and information is passed between successive iterations. For each iteration, the new assignments based on the desired proportions of data values are updated. We note that an efficient data representation of the graph is used: a hashtable where each entry represents a node in the graph with a pointer to a linked list of its neighbors.

Overlapping communities represent a bigger challenge for parallelization, because in the work case scenario, all communities would mutually overlap and be inter-dependent, making independent processing impossible in the same iterative cycle. However, several characteristics of the data reduce the difficulty: (1) In the long-tail distribution of communities by size, many nodes in small communities are members of only one community; (2) we can assign seeds depending on their "impact factor" in the whole graph, as a function of how many communities they are a member of, and the average size (number of nodes) of each of those communities. Then we can choose seeds, for example, with an average "impact factor"; (3) above a given community size threshold, target profiles can be fitted to communities based on the closeness of the community size to the target profile percentage of the whole graph. Below that threshold (approx. where the long-tail commences), target profiles can be assigned probabilistically based on target profile percentage. This issue is further discussed in the section "algorithmic complexity." Also, In Sect. 6.2, Table 11 and Figs. 11 and 12 summarize key community statistics which affect the complexity of the data propagation process.

6.3.3 *Complexity of the approach*

We have already considered the computational cost of the seed assignment and data propagation steps in the discussion about parallelization. For the computational cost of RMat and Louvain, the reader should consult the references for those methods and their parallel version which we have mentioned previously. As we have also mentioned, the topology creation and community labeling cost is only incurred for the experiments of Sect. 6.1. We envisage our method is mainly to be used for populating existing topologies, such as those processed in Sect. 6.2. However, we see the main challenge will be to efficiently process overlapping communities. In this scenario, the simple approach of considering that each community can be processed individually within a MapReduce iteration no longer holds for the seed assignment and data propagation steps.

Cross-checks have to be done in order to control data propagation which a node can be a seed in one community and be a neighbor of a seed in another community. In order to implement this, for each node a list is maintained of the communities in which it is present. Data propagation: The assignment of the target profiles to the seeds is very fast and will be O(NS × P), where NS is the number of seeds and P is the number of data attribute values to be assigned. Next, the assignment of data attribute values from seeds to neighbors is also fast and O(NS × MS × P) where NS is the number of seeds, MS is the average number of neighbors and P is the number of data attribute values. The similarity function which approximates the seed profile to be assigned to its neighbors will have cost $O(P^2)$ where P is the number of date attribute values. Finally, the remaining non-assigned nodes will have an assignment cost of O((NG − (NS × MS)) × P) where NG is the number of nodes in the whole graph. Finally, the similarity function which approximates the profile assignment to the non-assigned nodes will have a cost of $O(P^2)$. We note also that the average degree of the nodes is in general not very high for OSNs.

6.3.4 Advantages of the approach

The key advantages our approach are as follows: (1) The data assignment method is independent of the topology, whereas other methods require a given topology or build the topology together with the data assignment (if this is done). (2) Our method creates a rich dataset, without limit of the number of attribute values or their types. The work presented processes eight different attributes with a total of 46 distinct attribute values and 10 distinct profiles. (3) Our method is highly parameterizable, using target profiles to generate the data definitions themselves as well as their distributions, proportions and profiles, which are then propagated in the network. The propagation rules themselves can also be defined by the user, although a default configuration is given. (4) We also process overlapping communities, and (5) allow for three different levels of "dispersion" to represent real "noisy" social networks. This represents a level of complexity significantly greater than the majority of works in the state of the art (See Sect. 2).

6.3.5 Comparison with other approaches

In the following, we will comment the comparison of our approach with two other authors (Pérez-Rosés and Sebé 2015; Pérez-Rosés et al. 2016) and (Ali et al. 2014) and then we make some general comments about comparison between different approaches for synthetic OSN data generation.

In terms of the fitness measures of the results, Ali et al. used a fitness function which measures the fit between target attribute statistics and the statistics of the generated attributes (see Sect. 2.2 for details). Using the PSO stochastic optimization approach, a best fitness of 95 % was obtained, according to the authors. Pérez-Rosés, on the other hand, used a fitness measure based on the fit between the original skills datasets and the enriched (deduced) skills dataset. According to the authors, the fitness ranged between 76 % (for the "programming" skill) and an average of 97 % for the other four skills. For our method, using level 1 dispersion (see Sect. 5 for a description of this) we obtained a fit between target and generated attribute values of nodes in the three test datasets of between 70 and 78 % with the average fitness being 74.9 %. However, it is noted that for level 1 dispersion, we purposely introduce ≈20 % of noise in the node attribute-value assignments for each profile (refer to Sect. 5, assignment of seed neighbors' attribute values and homophily level). Then, for the remaining nodes (which are neither seeds nor neighbors of seeds) for level 1 we assign randomly ≈10 % of the time and assign the modal value ≈90 % of the time. In practice, this gives an overall target fitness of approx. 80 %. Thus, the average observed fitness of 74.9 % (with respect to 80 %) gives a fit value of 93.6 %. This fitness is competitive both with Ali's and Pérez-Sebé's published fitness.

In the case of the fitness of the profiles, we have taken an average value for the target percentages for each profile with respect to the assigned percentages, which gave a fit of between 71 and 79 % for the profiles evaluated (see Table 18). Again, taking into account the 20 % noise, this is competitive with Ali's and Pérez-Sebé's fitnesses. However, in our case the overlapping communities represent a more complex data assignment challenge, as well as the greater diversity of attributes and their values.

Next, in terms of computation (elapsed) time, we observe from the empirical section that we have processed an 84,000 node graph (LiveJournal) in 359 s, which is very favorable when compared with other approaches such as Ali et al. (2014), 7500 s for 10,000 node graph and (Pérez-Rosés et al. 2016), 411 s for 1925 node graph (however, Ali et al. are also building the topology simultaneously). We note that our method uses a time cutoff for the seed assignment: The majority (approx. 90 %) of the seeds are located fairly quickly, and then the algorithm finds it increasingly difficult to locate the remaining 10 %. However, with a 90 % coverage the method can easily complete the data population using median values for the communities and pseudo-random assignment (for noise). Finally, we observe that the papers Pérez-Rosés et al. (2016) and Ali et al. (2014) did not include details of their hardware configuration or the language their software was implemented in, thus making it more difficult to evaluate their absolute running times (our experimental setup is detailed at the start of Sect. 6).

In general, for a synthetic data generator which is going to perform "one off" data generations, our opinion is that the computation cost is secondary, as long as the process is parallelizable and the rate of increase in cost is not prohibitive as the network increases in size and data richness (more attributes and attribute values). However, the computation cost is a useful statistic to include. Second, the fit of the target profiles with the observed profiles is a good measure of the success of the method. Thirdly, the difference between target proportions and observed proportions of key attributes is also a good measure if this is the objective. Comparing topological statistics (degree, average path length, diameter, …) are only relevant if the topology and data generation are inter-dependent (part of the same process).

7 Conclusions

In this work, we have tackled the problem of generating realistic synthetic graph data which approximates an online social network, by populating a graph topology. We have tried two main approaches, the first using non-overlapping synthetic communities and the second using real overlapping "ground-truth" communities. The three-step data propagation process has proved effective: seed assignment, seed-neighbor assignment and assignment of the remaining nodes. As expected, the non-overlapping communities result in a more clean assignment of the profiles to the communities, whereas the overlapping communities tend to increase the "chaos" in the data assignment process.

Using a comprehensive set of data generation parameters, we have been able to control the process which enables us to obtain a good approximation of the desired profiles, proportions, and community assignments. We can also augment the level of "noise" in the system in terms of "dispersion" levels defined by different control parameter sets.

Acknowledgments This work is partially funded by the Spanish MEC (project TIN2013-49814-EXP). The author is grateful for the suggestions of Prof. Vladimir Estivill-Castro of the Pompeu Fabra University, Barcelona, Spain, and of Dr. Julián Salas of the University Rovira i Virgili, Tarragona, Spain.

Appendix: Pseudo-code of synthetic data generator

Procedure Synthetic_Data_Generator_1 // non overlapping communities
Input: Number V of vertices and E of edges,
control parameter set $\mathbb{C}P$ = [NS, {SP}, SD, {DT}, RV] // see Section 5 for definitions
Output: graph G populated with data
1. *RMat*
2. **For** |V| vertices and |E| edges **generate** an OSN-like topology.
3. *Communities*
4. **Calculate** communities using Leuven method and
assign community tag to each vertex.
5. *Calculate* medoid values MC for each community using centrality metric
6. Calculate distance Cdn of each node n in each community to mediod MC
7. Order nodes in each community by distance Cdn
8. S = **Seed_Assigner**(G, nSeeds)
9. // *Assign data to seeds and their neighbors in each community*
10. **For each** community c ∈ C do
11. S_c is the set of seed nodes in community c
12. NS_c = **Assign_Data_to_Seeds_and_Neighbor_Vertices_in_Community (**S_c , c, $\mathbb{C}P$**)**
13. // NS_c is the set of seeds and neighbors in c with data assigned
14. // *Assign data to remaining nodes in each community*
15. **Vc = Assign_Data_to_Unassigned_Vertices_in_Community**(NS_c , c, $\mathbb{C}P$)
16. // Vc is the set of all nodes in c with data assigned
17. **End do** // for each community
18. **For each** edge e connected to n, n' in G **do**
19. Assign a weight between 0 and 1 based on calculated distance between respective attribute-values.
20. **End Procedure**

Procedure Synthetic_Data_Generator_2 // overlapping ground truth communities
Input: Graph topology G, control parameter set $\mathbb{C}P$ = [NS, {SP}, SD, {DT}, RV] // see Section 5 for definitions
Output: graph G populated with data
1. *Read ground truth community graph*
2. *Communities*
3. **Assign** communities from ground truth labels by
assigning community tag to each vertex.
4. *Calculate* medoid values MC for each community using centrality metric
5. Calculate distance Cdn of each node n in each community to mediod MC
6. Order nodes in each community by distance Cdn
7. S = **Seed_Assigner**(G, nSeeds)
8. *Assign data to seeds and their neighbors in each community*
9. **For each** community c ∈ C do
10. S_c is the set of seed nodes in community c
11. NS_c = **Assign_Data_to_Seeds_and_Neighbor_Vertices_in_Community (**S_c , c, $\mathbb{C}P$**)**
12. // NS_c is the set of seeds and neighbors in c with data assigned
13. // *Assign data to remaining nodes in each community*
14. **Vc = Assign_Data_to_Unassigned_Vertices_in_Community**(NS_c , c, $\mathbb{C}P$)
15. // Vc is the set of all nodes in c with data assigned
16. **End do** // for each community
17. **For each** edge e connected to n, n' in G **do**
18. Assign a weight between 0 and 1 based on calculated distance between respective attribute-values.
19. **End Procedure**

References

Ali AM (2014) Synthetic generators for simulating social networks, 2014. Masters thesis, Univ. Florida

Ali AM, Alvari H, Hajibagheri A, Lakkaraj K, Sukthankar G (2014) Synthetic generators for cloning social network data. In: Proceedings of SocInfo 2014

Barrett CL, Beckman RJ, Khan M, Kumar VSA, Marathe MV, Stretz PE, Dutta T, Lewis B (2009) Generation and Analysis of Large Synthetic Social Contact Networks. In: Proceedings of the 2009 Winter Simulation Conference, 13–16 Dec 2009, pp 1003–1014

Bastian M, Heymann S, Jacomy M (2009) Gephi: an open source software for exploring and manipulating networks. Int AAAI Conf Weblogs Soc Media ICWSM 8(2009):361–362

Block P, Grund T (2014) Multidimensional homophily in friendship networks. Netw Sci (Camb Univ Press) 2(2):189–212

Blondel VD, Guillaume JL, Lambiotte R, Lefebure E (2008) Fast unfolding of communities in large networks. J Stat Mech P10008

Boncz P, Perez M, Gavalda R., Angles R, Erling O, Gubichev A, Spasić M, Pham MD, Martínez N (2014) Benchmark Design for Navigational Pattern Matching Benchmarking. LDBC Cooperative Project FP7 – 317548. Coordinators: Arnau Prat, Alex Averbuch. Issue 3 28/09/2014

Cha M, Haddadi H, Benevenuto F, Gummadi KP (2010) Measuring User Influence in Twitter: The Million Follower Fallacy. In: Proceedings of 4th Int. AAAI Conf. on Weblogs and Social Media (ICWSM), vol 10, pp 10–17

Chakrabarti D, Zhan Y, Faloutsos C (2004) R-mat: A recursive model for graph mining. In: Proc. SIAM Data Mining Conference, 2004. SIAM, Philadelphia, PA

Currarini S, Redondoy FV. A Simple Model of Homophily in Social Networks (2013) University Ca' Foscari of Venice, Dept. of Economics Research Paper Series No. 24, 2013

Dean J, Sanjay G (2008) MapReduce: simplified data processing on large clusters. Commun ACM 51(1):107–113

Dehghani M, Johnson K, Hoover J, Sagi E, Garten J, Parmar NJ, Vaisey S, Iliev R, Graham J (2016) Purity homophily in social networks. J Exp Psychol Gen 145(3):366–375

Dunbar RIM (1993) Coevolution of neocortical size, group size and language in humans. Behav Brain Sci 16(4):681–735

EU's Data Protection Directive (2015) Justice, Protection of personal data. http://ec.europa.eu/justice/data-protection/

Girvan M, Newman MEJ (2002) Community structure in social and biological networks. Proc Natl Acad Sci USA 99(12):7821–7826

Hagberg A, Schult D, Swart, P, Conway D, Séguin-Charbonneau L, Ellison C, Edwards B, Torrents J (2004) Networkx. High productivity software for complex networks. Webová strá nka http://networkx.lanl.gov/wiki

Hajibagheri A, Hamzeh A, Sukthankar G (2013). Modeling information diffusion and community membership using stochastic optimization. In Advances in Social Networks Analysis and Mining (ASONAM), 2013 IEEE/ACM International Conference on (pp 175–182). IEEE. describes our community detection algorithm, GPSODM

Hajibagheri A, Lakkaraju K, Sukthankar G, Wigand RT, Agarwal N (2015) Conflict and Communication in Massively-Multiplayer Online Games, Social Computing, Behavioral-Cultural Modeling, and Prediction, Vol. 9021, Lecture Notes in Computer Science, pp 65–74, 17 March 2015

Jones R, Kumar R, Pang B, Tomkins A (2007) I know what you did last summer: Query logs and user privacy, Sixteenth ACM Conf. on Information and Knowledge Management, ser. CIKM. 2007, pp 909–914

Kelly, H. (2012) "83 million Facebook accounts are fakes and dupes". CNN, August 3, 2012. http://edition.cnn.com/2012/08/02/tech/social-media/facebook-fake-accounts/

Kim M, Leskovec J (2011) Modeling Social Networks with Node Attributes using the Multiplicative Attribute Graph Model. In: Proc. UAI 2011, 27th Conf. on Uncertainty in Artificial Intelligence, Barcelona, Spain, July 14–17, 2011

Korsgaard M, Picot A, Wigand R, Welpe I, Assmann J (2010) Cooperation, coordination, and trust in virtual teams: Insights from virtual games. In: Online Worlds: Convergence of the Real and the Virtual

Kossinets G, Watts D (2006) Empirical analysis of an evolving social network. Science 311(5757):88–90

Kossinets G, Watts D (2009) Origins of homophily in an evolving social network. Am J Sociol 115(2):405–450

Lakkaraju K, Whetzel J (2013) Group roles in massively multiplayer online games. In: Proceedings of the Workshop on Collaborative Online Organizations at the 14th International Conference on Autonomous Agents and Multiagent Systems

Lee J, Lakkaraju K (2014) Predicting guild membership in massively multiplayer online games. In: Proceedings of the International Conference on Social Computing, Behavioral-Cultural Modeling, and Prediction, Washington, D.C., April 2014

Leskovec J (2008) Dynamics of Large Networks. PhD Thesis, School of Computer Science, Carnegie-Mellon Univ

Leskovec J, Kleinberg J, Faloutsos C (2005) Graphs over time: densification laws, shrinking diameters and possible explanations. In: Proc. KDD '05, 11th ACM SIGKDD Int. Conf. of Knowledge Discovery and Data Mining, 2005, pp 177–187

McAfee, A., Brynjolfsson, E. (2012) Big Data: The Management Revolution, Harvard Business Review, October 2012 Issue

McPherson M, Smith-Lovin L, Cook J (2001) Birds of a feather: homophily in social networks. Annu Rev Sociol 27:415–444

Mislove A, Marcon M, Gummad, KP, Druschel P, Bhattacharjee B (2007) Measurement and Analysis of Online Social Networks. In: Proceedings of IMC '07, 7th ACM SIGCOMM Conference on Internet Measurement, pp 29–42

Minitab 17 Statistical Software (2010). [Computer software]. State College, PA: Minitab, Inc. (www.minitab.com)

Nettleton DF (2013) Data mining of social networks represented as graphs. Comput Sci Rev 7:1–34

Nettleton, DF (2015) Generating synthetic online social network graph data and topologies, 3rd Workshop on Graph-based Technologies and Applications (Graph-TA), UPC, Barcelona, Spain, March 18th 2015

Nettleton DF, Salas J (2016) A data driven anonymization system for information rich online social network graphs. Expert Syst Appl 55:87–105

Newman MEJ (2004) Fast algorithm for detecting community structure in networks. Phys Rev E 69:066133

Ovelgonne M (2013) Distributed community detection in web-scale networks. In Advances in Social Networks Analysis and Mining (ASONAM), 2013 IEEE/ACM International Conference on, pp 66–73

Pérez-Rosés H, Sebé F (2015) Synthetic generation of social network data with endorsements. J Simul 9(4):279–286

Pérez-Rosés H, Sebé F, Ribó JM (2016) Endorsement Deduction and Ranking in Social Networks, Computer Communications, Vol. 73, Part B, 1 January 2016, Pages 200–210, Elsevier

Pham MD, Boncz P, Erling O (2012) S3G2: a Scalable Structure-correlated Social Graph Generator. In: Proc. 4th TPC Technology Conference, TPCTC 2012, Istanbul, Turkey, August 27, 2012, Lecture Notes in Computer Science, vol. 7755, pp 156–172

Plimpton SJ, Devine KD (2011) MapReduce in MPI for large-scale graph algorithms. Parallel Comput 37(9):610–632

Que X, Checconi F, Petrini F, Wang T, Yu W (2013) Lightning-fast Community Detection in Social Media: A Scalable Implementation of the Louvain Algorithm. Technical Report AU-CSSE-PASL/13-TR01 (Auburn University, IBM TJ Watson)

Ramakrishnan N, Keller B, Mirza BJ. (2001). A. Grama, and G. Karypis, "Privacy risks in recommender systems," IEEE Internet Computing, vol. 5, no. 6, pp. 54–62, 2001

Robins G, Pattison P, Woolcock J (2005) Small and other worlds: global network structures from local processes. Am J Sociol (AJS) 110(4):894–936

Sala A, Cao L, Wilson C, Zablit R, Zheng H, Zhao BY (2010) Measurement-calibrated Graph Models for Social Network Experiments, WWW 2010, April 26–30, 2010, Raleigh, North Carolina, USA

Schult DA, Swart P (2008) Exploring network structure, dynamics, and function using NetworkX. In: Proceedings of the 7th Python in Science Conferences (SciPy 2008). Vol. 2008. 2008

Tang L, Liu H, Zhang J, Nazeri N (2008). Community evolution in dynamic multi-mode networks. In: Proc. of the 14th ACM SIGKDD, KDD'08, New York, NY, USA, 2008, pp 677–685

Tarbush B, Teytelboym A (2012) Homophily in Online Social Networks, Internet and Network Economics, Volume 7695 of the series Lecture Notes in Computer Science pp 512-518 (2012). In: Proc. Internet and Network Economics: 8th International Workshop, WINE 2012, Liverpool, UK, December 10–12, 2012. Springer Berlin Heidelberg

Verbrugge LM (1983) A research note on adult friendship contact: a dyadic perspective. Soc Forces 62(1):78–83

Viswanath, B, Mislove A, Cha M, Gummadi, KP. (2009). On the Evolution of User Interaction in Facebook. In: Proceedings of 2nd ACM workshop on Online Social Networks, WOSN'09, Barcelona, Spain, 2009, pp 37–42

Wang X, Sukthankar G (2013) Link prediction in multirelational collaboration networks. In: Proceedings of the IEEE/ACM Int. Conf. on Advances in Social Networks Analysis and Mining, pp 1445–1447, Canada, Aug 2013

Wang X, Maghami M, Sukthankar G (2011) Leveraging network properties for trust evaluation in multi-agent systems. In: Proc. IEEE/WIC/ACM Int. Conf. on Web Intelligence and Intelligent Agent Technology, pp 288–295

Wattenhofer M, Wattenhofer R, Zhu Z (2012) The YouTube Social Network. In: Proc. 6th Int. AAAI Conf. on Weblogs and Social Media, Dublin, Ireland, 4–7 June, 2012, pp 354–361

Weil, J. (2015) "Mark Zuckerberg: Creator of Facebook", Abdo Publishing, Minneapolis, USA. Ed. Arnold Ringstad, ISBN 978-1-62403-647-7 (2015)

Wigand R, Agrawal N, Osesina O, Hering W, Korsgaard M, Picot A, Drescher M (2012) Social network indices as performance predictors in a virtual organization. In: proceedings of the 4th international conference on Computational Aspects of Social Networks (CASoN) pp 144–149

Xie J, Szymanski BK (2013). Labelrank: A stabilized label propagation algorithm for community detection in networks. In: Network Science Workshop (NSW), 2013 IEEE 2nd (pp 138–143)

Xie J, Chen M, Szymanski BK (2013). LabelrankT: Incremental community detection in dynamic networks via label propagation. In: ACM Proceedings of the Workshop on Dynamic Networks Management and Mining (pp 25–32)

Yang J, Leskovec J (2012) Defining and Evaluating Network Communities based on Ground-truth. ICDM, 2012

Zhao W, Ma H, He Q (2009) Parallel K-Means Clustering Based on MapReduce. In: Proc. CloudCom 2009, LNCS 5931, pp 674–679, 2009