HySoN: A Distributed Agent Gossip Protocol for Group Formation in Social Networks

Fabrizio Messina and Giuseppe Pappalardo and Domenico Rosaci and Corrado Santoro and Giuseppe M.L. Sarné

Abstract On-line social networks allows people to easily interact with each other by means of social computer services. This scenario makes possible to search in a social network for affinities or new opportunities that satisfy specific requirements. However, for many users such activities often imply undesirable accesses to personal sensitive data. In this scenario we propose a novel approach, called HySoN, based on an overlay network of software agents, which exploits a gossip protocol. HySoN allows users to locally maintain sensitive user’s data, satisfying the privacy requirements preserving sensitive data. Indeed, the properties involved in the HySoN user aggregation are inferred by local data not published in the social network. Some experimental results obtained on simulated on-line social networks data show the searching of suitable nodes is very efficient due to the topology of the overlay network, which exhibits the small-world properties.

1 Introduction

The rapid diffusion of general purpose social networks like Facebook\(^1\), MySpace\(^2\), Twitter\(^3\), and thematic networks as, for instance, Flickr\(^4\) or Linkedin\(^5\), is represent-
ing an important opportunity for people to socialize, share multimedia contents, discover work opportunities and so on. Moreover, these online social environments are also exploited by companies to attract new clients and partners. However, in such a context, important privacy concerns arise for users, since the opportunity to find new contacts based on common interests strongly depends on data which users are willing to share in the network. This is a very typical concern, since users would like to find interlocutors with good affinity, but often this affinity can be found only by analysing sensitive data that they don’t want to share.

In this work we propose an approach for supporting the users of an online social network to form groups of users having similar interests. The goal of the approach is that of providing each user with homogeneous environments (the groups), suitable for performing social activities as sharing multimedia contents, making discussions, exchanging job information and so on. This approach is based on an overlay network of software agents [17], which allows the users to locally maintain sensitive user’s data, while contacting interlocutors having a certain level of affinity is still made possible. The role of software agents consists in assisting users by analysing sensitive (local) data in order to extract relevant properties which can help in the social network to find other users having similar interests. To this aim, each agent can be delegated by its own user to start the construction of a custom overlay network called HySoN (Hyperspace Social Network) whose topology reflects the distribution of a set of users’ interests (properties). In order to build such a network the software agents, totally trusted, exchange messages by a gossip protocol over the network of the friends. This way, as shown in the experimental section, approximately all the agents/nodes are reached by the overlay construction request with a limited number of steps and messages. Once the properties are mapped into the overlay network, the agents can send their requests over the network in order to find a set of agents reflecting some required properties. Moreover, as we will discuss in the experimental section, the searching of suitable nodes is very efficient due to the topology of the network, which exhibits the small-world properties. Finally, since properties involved in user aggregation are inferred by local data not published in the social network, the user privacy on sensitive data is basically preserved.

The remaining of the paper is organized as follows. In Section 2 we discuss related works. Section 3 describes the details of the basic scenario, while 4 introduces the technique for building the HySoN network and finding suitable nodes. Finally, an experimental evaluation of the proposed approach is provided in Section 5 and in Section 6 we draw some conclusions and future works.

2 Related works

In the scientific literature the concept of social network has been widely investigated [11, 16], and different definitions have been provided [49, 47]. Generally, a social network can be described as the sum of the relationships connecting some individu-
als with each other [33], where a relationship embeds both personal knowledge and social resources [10].

This background is shared from on-line social networks [12, 24], which mainly differ from other social structures for: (i) the presence of a communication network; (ii) the absence of physical contacts; (iii) the potential presence of fake identities. In this way people interact with each other by using social computer services, giving rise new behavioural dynamics, widely studied in these latter years. Therefore, in this section, the examined approaches are those that, to the best of our knowledge, are the closest to the material presented in this paper, that deals with the formation of groups in online social networks.

Nowadays, on-line social networks are a worldwide phenomenon with a massive economic impact. Such networks allow relationships and virtual groups to be created in a very simple and interactive way [27]. Virtual groups are formed, often in unplanned manner, on specific focuses, motivations or requirements and this process is often described in the form of a Constraint Satisfaction Problem (CSP) [9]. The group membership is generally barrier free even tough norms and requirements can exist [40], as that of playing well-defined roles statically or dynamically assigned (based on members’ skills).

Different approaches are available for exploring the mechanisms underlying the group formation [39], although it lacks a well defined standard methodology. For instance, both some analytic and synthetic measures (representing properties as cohesion, density, distances, etc. [45, 47]) and the graph theory (to obtain a visual representation of the social network by modelling actors and relationships as nodes and arcs of a graph) can be used. Besides, the graph theory can be directly adopted to search in a social network potential candidates to form groups, usually realizing an its partition by means of a merit function as in [7, 44].

Community members can be equipped with some kind of personal profile, potentially available to friends or similar entities. Such profiles store users’ orientations and can be exploited for discovering potentially interesting groups. These processes can be driven also by means of machine-to-machine approach, for example by using the multi-agent technology [8, 25, 30] that offer several opportunities of performing them in a personalized and dynamic way.

In particular, a multi-agent system is naively implementable in a decentralized manner, similarly to HySoN. In this context, [21] exploits agents able to be adaptive with respect to changes occurring in the graph topology by locally rewiring all their relationships and adapting themselves to the different social context. Also in [41], using a distributed approach, software agents associated with users dynamically and autonomously manage the evolution of the groups. For each user, agents detect the most suitable groups to join with based on a dissimilarity measure, while the agent associated with the group-administrator accepts only those join requests that come from users profile-compatible with the profile of the group.

Clustering algorithms [6, 19, 29, 44] are a common way to discovery groups based on quantitative measures, usually extracted by data stored in personal profiles. They are classified in: (i) hierarchical [23] (recursively finding nested clusters either in agglomerative or divisive mode) and (ii) partitional (finding all the clusters
simultaneously as a data partition without to impose a hierarchical structure). As a result, users are representable as points in a multidimensional space (the number of dimensions corresponds with the number of considered parameters in their profiles), where the Euclidean distance summarizes the adopted clustering measure similarly to HySoN, although, as explained in the sections below, it works in a full distribute manner, hence it allows users to not maintain sensitive data in the remote server.

Two relevant issues might be taken into account in composing groups for specific tasks, i.e. (i) the workload balancing and (ii) the coordination costs. For instance, in [20, 31, 32] the first issue seems to be completely ignored. Differently, Anagnostopoulos et al. do not consider in [1] the coordination costs, although in [2] they provide to consider together the two mentioned issues, while in [32] the proposal of Li and Shan differ for a priori fixing the numerosity of each group by exploiting a modified the Enhanced-Steiner algorithm [31]. In HySoN each member performs a little slice of the overall task to constitute a group providing, in such a way, a solution to the first issue, while the second issue is trivially solved by the approach adopted in the construction of the HySoN overlay network (see Section 4).

The dynamics of group formation processes have been recently addressed in [4] and in the recent models proposed by Sarkar and Moore [43] and by Holme and Newman [22] to consider frequent changes occurring in the relationships and attributes, while an unsupervised discovery of groups in networks densely connected, similarly to many real-world networks, is proposed in [46]. Similarly, our proposal can automatically rearranging groups to take into account changes occurred in the social network even in presence of huge networks (similarly to HySoN).

Greedy strategies are typical operational research techniques that can be applied, likewise to HySoN, in a social network scenario. For example, it has been used on GitHub (github.com), a Facebooklike platform, which permits to each affiliated of self-organising teams of whatever size devoted to solve specific tasks or for minimizing the team cost formation. Another greedy algorithm to solve the problem of realizing a team of experts is described in [13], here firstly it is constructed a weighted bipartite graph with experts on one side and tasks on the other and then it continues by assigning an expert to each task until tasks or experts end.

Finally, concerning fully decentralized resource selection algorithm, an interesting approach is presented in [14], where a two-layered gossip based protocol is exploited to deal with nodes failures, joins and leaves, leading to the construction of an additional random overlay network with a high level of redundancy. As we discuss below, our solution performs overlay maintenance by preserving the "essentially critical neighbours" in only one step within the overlay construction and maintenance algorithm, as in [34, 35, 37], by exploiting the properties of small-world networks.
3 Scenario

We consider a Social Network $S$, having $n$ users, and represented by a graph $G = \langle N, A \rangle$, where $N$ is a set of nodes, such that each node $n \in N$ represents a user of $S$ that we denote by $n_u$, and $A$ is a set of arcs, such that each arc $a = \langle u, v \rangle$ represents the existence of a friendship relation between the users $n_u$ and $n_v$.

We assume that a software agent is allowed to reside on each user machine. This assumption partially overturns the architecture and the vision of the typical social network, on which user data reside on the Cloud [3, 18] and the interactions between users occur through communications with the remote server. Nevertheless, as stated in Section 1, local software agents, which are trusted by users [42] are allowed to retrieve and analyse their personal (local) data which are considered “sensitive” by the users. Hence software agents can exploit a detailed profile based not only on data shared in the social network, but also on private data. For instance, a software agent is able to analyse the private emails of its user to extrapolate information about its main interests. Hence the user profile can be used to create new groups of users with a decentralized technique (see Section 4).

From hereafter, we will use the terms user and node interchangeably. We also assume that a set $P$ of properties is associated with the network of agents, where a property can represent, for instance, the interest of a user for a set of topics, or the set of social network pages that the node likes, or the set of social network groups which the node joins with, or the number of black and white photos which reside on the local disk of the user, and so on. Moreover, we also assume that each property $p \in P$ has a value belonging to a given domain $D_p$. For instance, if $p$ represents the number of black and white photos held by the user and detected by the agent, then $D_p$ will be a set of positive integer values, while if $p$ represents the set of the topics which the agent consider relevant for the user, then $D_p$ will be the set of all the possible sets of topics available in $S$. A profile schema $PS$ is also associated with the social network, composed of a list of properties belonging to $P$, denoted by $p^1, p^2, ..., p^l$, where $l$ is a value depending on the given social network. In this scenario, a profile instance $p_n$ is associated with each node $n$ of $N$, where $p_n$ is a list $p^1_n, p^2_n, ..., p^l_n$, such that each element $p^i_n$ is an instance of the property $p^i \in PS$.

Each node, in order to evaluate if joining or not in a group with another node, can express some requirements on the properties of the other node. A requirement on a property $p$ is a boolean function that accepts as input a property $p \in P$ and returns a boolean value $b \in \{true, false\}$. For instance, a requirement on the number of friends can state that a node must satisfy a given condition regarding the number of friends (e.g. having a number of friends higher than 50), returning the value true if the statement is verified by the property instance of the node, false otherwise.

Furthermore, each node $n$ has associated a set of requirements $R_n$, and $n$ accepts to belong to a group if at least the $\alpha$ percent of the members of the group satisfy at least the $\beta$ percent of the requirements of $R_n$, where $\alpha$ and $\beta$ are parameters set by the node $n$. Moreover, each node $n$ can express his preferences regarding the group types he wants to join with, using the parameters availability, accessibility and interests, where availability is a boolean value, accessibility is a value belong-
ing to the set \{public, private, secret\} and \textit{interests} is a set of topics. The value \(availability = false\) means that \(n\) does not desire to join with any group, while \(availability = true\) means that \(n\) is available to join with a new group under the condition that both the topics of the group are contained in the set \textit{interests} and the access type of the group is equal to the value of the parameter \textit{accessibility}.

A software agent \(a_n\) is also associated with each node of \(N\), able to perform the following two tasks:

- It can be delegated by its node \(n\) to form a group of nodes, based on a set of requirements \(RSET\). To this aim, the agent of \(n\) sends a joining request to all the nodes belonging to the admissible region \(S(RSET)\), determined by using the \textit{HySoN} protocol (see Section 4). The joining request is a tuple \(JR = \langle t, a \rangle\), where \(t\) is a set of topics associated with the group and \(a\) is the access type of the group.
- It negatively responds to a request of joining \(JR = \langle t, a \rangle\) with a group coming from another agent \(a_m\) if the parameter \(availability = false\). Otherwise, if \(availability = true\), it positively responds if both the topics \(t\) of the group are contained in the set \textit{interest} and the access type \(a\) of the group is equal to the value of the parameter \textit{accessibility}. After this joining, the agent of \(n\) periodically verifies that at least the \(\alpha\) percent of the group members’ profiles satisfy at least the \(\beta\) percent of the requirements of \(R_n\). If this verification fails, the agent automatically deletes the user \(u_n\) from the group.

4 The \textit{HySoN} Protocol

According to the schema reported in the Section above, any agents can be delegated by its user to form a new group basing on some properties of interest. Therefore the delegated agent will look for a set of suitable nodes according to a set of requirements \(RSET\). Since we are considering a large and very dynamic social network, as for instance Facebook, and the data to be analysed by the software agents is distributed over the users machines, in order to support such a finding process we adopt a \textit{peer-to-peer} approach since it is known to be more efficient and scalable than a centralized solution [26, 15].

The finding schema adopted in this paper is derived from a fully decentralized resource finding approach for large scale distributed systems developed and studied by the authors in the recent years [35, 37], and it is based on the construction of an overlay network of software agents which is the building base of the \textit{HySoN} (Hyper-space Social Network) protocol. As we discuss in Section 4.2, the \textit{HySoN} topology reflects the distribution of the selected properties over the users. This characteristic, as we show in Section 5, is the basis for an efficient finding process.

The \textit{HySoN} protocol includes three phases which are depicted in Figure 1. Once user data have been analysed, the \textit{HySoN} protocol starts with the dissemination (the dashed narrow (1) in Figure 1) of a gossip message among the agents; this phase is detailed in Section 4.1.
The $HySoN$ overlay construction (the dashed narrow (2) in Figure 1) represents the $HySoN$ link established between agent $A_U$ and the generic agent already reached by the gossip message. Although the overlay construction partially overlaps with the dissemination of the gossip message (see Section 4.1), it can be considered the second phase. As we discuss into Section 4.2, the $HySoN$ construction algorithm is fully decentralized and is executed on the nodes reached by the gossip protocol. The algorithm used for the $HySoN$ construction is also the basis for the adopted finding process, i.e. the third phase (the dashed narrow (3) in Figure 1), which is discussed in Section 4.3.

### 4.1 Disseminating $HySoN$ construction request

In the proposed model it is assumed that whenever two users are friends in the social network, their agents are able to communicate and exchange information. Therefore, the overlay topology used in the dissemination phase reflects the friendship of the social network.

The construction request process begins when an agent is delegated by its user to start the construction of the $HySoN$ overlay network and it is explained below:

1. The user agent sends a “gossip” message to its own neighbors (i.e. the software agents connected to its own friends) by a simple gossip-based protocol [28], whose details are shown in Section 5.
2. The gossip message contains a list of properties $(p^1, p^2, ..., p')$ to be used by the agents to compute and expose their own coordinates into the $HySoN$ hyperspace (see Section 4.2).
3. Once the gossip message is received by the generic agent $n$, user data are analysed to obtain values for the properties $(p_1, p_2, ..., p_l)$ specified in the message.
4. Agent $n$ send the gossip message to some of its neighbours according to the gossip algorithm described in Section 5.
5. Finally, to start with the second phase (overlay construction), agent $n$ connects itself to the sender and begin to run the HySoN overlay construction algorithm as described in Section 4.2.

Overall phase 1 (dissemination) and phase 2 (overlay construction) overlap, i.e. while the gossip message is spreading into the social network, the nodes already reached by the gossip are executing the overlay construction algorithm, which is described in the Section below.

### 4.2 HySoN construction algorithm

The HySoN overlay construction is the key for a fast and effective fully decentralized finding process, and it is performed by means of a decentralized algorithm which runs on the nodes reached by the gossip message.

We assume that the generic agent $n$ has been reached by the gossip message and retrieved the values for the requested properties. Property values are mapped into “agent coordinates” in a multidimensional space, and are used to compute the distance between nodes as specified in the construction algorithm listed below:

1. agent $n$ contacts its neighbours in order to obtain, in turn, their own neighbours; in this way the agent can compute the whole set of agents at 2-hops;
2. the set is ordered by using the Euclidean distance of each node from $n$;
3. based on some threshold parameters $\deg_{\text{min}}$ and $\deg_{\text{trg}}$, node $n$ rearranges its links, interconnecting itself with at least $\deg_{\text{min}}$ near nodes, but at most $\deg_{\text{trg}}$ nodes.

As detailed in [35], during the last step of the construction algorithm, the so-called “essentially critical” neighbours are preserved in order to make sure that the network stays connected during the process. The effect of the said steps is to create clusters of nodes featuring a short intra-cluster distance, while keeping long links between clusters (which actually are the essentially critical neighbours). Therefore the HySoN construction algorithm features a structure quite similar to a small-world [48], i.e. a network which shows a high clustering degree and a very low average path length; as we show in Section 4.3, these characteristics are very important to make resource finding effective. Since the Euclidean distance is a similarity measure for the properties mapped on the coordinates, such clusters are characterized by nodes with a resource status very closed to each other. By exploiting short links, a fast navigation inside the cluster is possible to e.g. refine the finding process, while, by using long links, the region (i.e. the cluster) in the hyperspace where the nodes offering the requested properties reside is quickly reached.

Some steps in the construction of an overlay network – in this case two properties/attributes have been mapped – are depicted in Figure 2.
4.3 Finding suitable nodes with HySoN

According to the said hyperspace abstraction, not only a node can be viewed as a point in the metric space but also a finding request can be represented in the same way. Such a request carries a set of requirements on the set of properties which are mapped as coordinates of the hyperspace.

In this way the request denotes a point representing a corner of a region or semi-space whose internal nodes are those exploiting suitable properties for the given finding request. Such a semi-space is called the admissible region $S(RSET)$.

The finding process is based on a fully decentralized finding algorithm, which is based on the following check-and-forward model:

1. a node receiving the request checks if its properties satisfy the set of requirements in $RSET$; if this is the case, the node belongs to the admissible region; therefore the finding process continues with step 4;
2. if the set of requirements $RSET$ does not hold for the node, the request is submitted to one of its neighbours, which is selected on the basis of appropriate heuristics; such heuristics are chosen in order to help the request to reach the admissible region as soon as possible [37];
3. the algorithm keeps track of all the nodes visited (this set is carried together with the request), if all the possible nodes to jump onto are already analysed, the system does not include a node suitable to host the request, so the algorithm terminates with failure;
4. when we found a node belonging to the admissible region, by suitably navigating through links[37] the algorithm can reach other valid nodes, in order to build the needed set for the group formation.

Since we simulated all the phases of the proposed approach (i.e. dissemination, overlay construction and finding process), in the next section we discuss some experimental results.
5 Experimental results

We evaluated the proposed approach in a social network of $10^6$ users which was simulated with the ComplexSim simulation platform [36, 38]. The topology adopted for the social network was generated through the scale free model discussed by Barabasi [5].

For the first phase (overlay construction request dissemination), a simple probabilistic gossip protocol [28] has been adopted. Listing 3a reports a frame of the behaviour of the HySoN agent, while Listing 3b reports function `check_gossip()`, used to disseminate the gossip message. Its schema is simple, and its behaviour is basically tuned by the threshold parameter $v$. When $v$ is close to 1, the gossip message is likely to be propagated to the whole agent’s neighbourhood; in this case the hubs of the network generates, in average, huge bags of messages which can involves in excessive overheads. Conversely, in order to reach most of the nodes, the threshold should not be too low. Moreover, a TTL (Time-To-Live) and a message cache are needed to stop the process in a few steps and to limit the number or duplicated messages.

We report in Figure 4a the experimental result (in logarithmic scale) of the dissemination phase in terms of number of nodes reached by the gossip, and generated messages, for different values of the threshold $v$ (see Listing 3b) and $TTL = 3$. We note that when the threshold span in the range $0.4 \div 0.5$, about the 80% of nodes can be reached with no more than (about) $3n$ messages (where $n$ is the total number of nodes). We do not show the cost, in terms of time-steps, of the gossip protocol, as it is comparable to the average path length of the network (< 10), hence it is negligible.

Table 4b reports the main characteristics of the resulting network in terms of average minimum path (see index $l$) and clustering coefficient (see parameter $C$) for two couples of target degree (parameter $deg_{\min}$ and $deg_{\text{trg}}$, see Section 4.2). To this aim a bidimensional hyperspace is build, hence two properties have been considered. We also observe that the HySoN clustering algorithm leads to a stable network.

```c
// Agent (n) behaviour
// ...
msg = receive_msg(n);

if (is_gossip (msg,n,v)==1)
   // Starts the hyson clustering
   // The sender is the first neighbour
   hyson(msg.sender , msg.properties);
else if ( is_searching (msg,n)==1)
   // distributed search
   // algorithm
   search (msg.request)
```

```c
int check_gossip (msg,n,v)
{
   if (!is_gossip (msg))
      return 0;
   if (!is_cached (msg,n) OR msg.ttl == 0))
      put_msg_in_cache (msg,n);
   msg = create_msg (msg.ttl - 1)
   for all nn in neighbours(n)
      if random_uniform(0,1) < v
         send(msg, nn)
      return 1;
   else return 0;
}
```

Fig. 3  a) Agent behaviour  b) Gossip function
after a few steps (about 10) and that the network exhibits the small world properties (i.e. low geodesic path, \(l\), and high clustering coefficient, \(C\)).

A set of experiments have been performed to evaluate the performance of the finding algorithm, in terms of necessary time-steps to reach the admissible region \(S(RSET)\). In this case the property values were appropriately changed for different experiments; in this way the ratio of suitable nodes (x-axis in Figure 5), i.e. nodes which satisfy the constraints of the finding requests, can assume the desired range.

As depicted in Figure 5, the finding algorithm performs well in terms of time-steps (see first quartile, median and third quartile). This result is due to the fact that the agents holding similar properties are connected with high probability, while the average geodesic path is maintained small enough.

Therefore, the cost of the HySoN network construction discussed before (see Figure 4a) in terms of number of messages, is compensated not only by the improvements in terms of privacy (users do not have to share sensitive data in the social network, as discussed in section 3) but also by the performance of the finding process. Indeed, once the network is built, multiple requests can be sent by different agents according to different requirements with the aim of forming a new group.
In on-line social networks, an important issue is that of preserving the users’ privacy. However, the possibility for the users of finding new profitable contacts strictly depends on information which the user is willing to share on the network. In this paper, we have presented an agent-based mechanism to support the users of an online social network in the formation of groups composed of users having similar interests. We have proposed to build the groups by using an overlay network of software agents, which allows the users to locally maintain sensitive information, on the one hand, and to offer an efficient mechanism for each user to reach promising interlocutors, on the other hand. The agents operate on the users’ behalf, analysing the local user’s information in order to extract relevant properties which are then exploited to find the appropriate interlocutors. The set of the interlocutors is organized into a custom overlay network called HySoN (Hyperspace Social Network), built by exchanging mutual messages between agents using a gossip protocol.

The so-built network provides the users with the possibility to send their requests over the network in order to find a set of agents having some required properties, in order to form a group of users.

We have shown by experiments performed on a large network of simulated users that our approach is capable of reaching approximately all the agents in a small number of steps and with a limited number of exchanged messages. The experiments also show that the search of suitable nodes is very efficient due to the topology of the network, which exhibits the small-world properties. The main advantage of our approach is that the properties involved in the user’s aggregation are inferred by local data not published in the social network, thus preserving the users’ privacy with respect to their sensitive data.

We note that a limitation of our approach is in the assumption that the properties declared by the users of the social networks are actually owned. If some user exhibits a misleading or fraudulent behaviour, the result of the group formation activity might be compromised. As for our ongoing research, we are planning to
HySoN: A Distributed Agent Gossip Protocol for Group Formation in Social Networks

introduce a trust mechanism in our approach, that will use trust measures in order to limit the effects of misbehaving nodes in the formation of groups.

References