2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining

Modeling Information Diffusion and Community Membership using Stochastic Optimization

Alireza Hajibagheri
Department of EECS
University of Central Florida
hajibagheri@knights.ucf.edu

Ali Hamzeh
Computer Science and Engineering Department
Shiraz University
ali@cse.shirazu.ac.ir

Gita Sukthankar
Department of EECS
University of Central Florida
gitars@eecs.ucf.edu

***Abstract*—Communities are vehicles for efficiently disseminating news, rumors, and opinions in human social networks. Modeling information diffusion through a network can enable us to reach a superior functional understanding of the effect of network structures such as communities on information propagation. The intrinsic assumption is that form follows function—rational actors exercise social choice mechanisms to join communities that best serve their information needs. Particle Swarm Optimization (PSO) was originally designed to simulate aggregate social behavior; our proposed diffusion model, PSODM (Particle Swarm Optimization Diffusion Model) models information flow in a network by creating particle swarms for local network neighborhoods that optimize a continuous version of Holland's hyperplane-defined objective functions. In this paper, we show how our approach differs from prior modeling work in the area and demonstrate that it outperforms existing model-based community detection methods on several social network datasets.**

I. Introduction

Due to its importance in everyday life, information diffusion has become a key area of research in social informatics. The physics of information diffusion has changed with the mainstream adoption of the Internet and the WWW. Until a few years ago, the major barrier for someone who wanted a piece of information to spread through a community was the cost of the technical infrastructure required to reach a large number of people. Today, with widespread access to the Internet, this bottleneck has largely been removed. More recently, there has been interest and attention not just in observing the flow of information and innovation, but also in creating, influencing, and simulating it.

Modeling information diffusion is a key component for addressing two research questions:

1) maximizing the spread of influence through opinions, ideas, and recommendations;
2) early containment of food contamination and disease.

In this paper, we investigate the use of information diffusion for a third problem: identifying the membership of communities in social networks. Our proposed community detection algorithm searches for a locally-optimal assignment of community labels that maximizes an agent's information flow. Our approach has two elements: 1) using stochastic optimization to calculate the information flow between agents that maximizes an objective function 2) a social choice process by which the agents select the community that increases their share in the simulated information economy. We demonstrate that improving the verisimilitude of the model of information diffusion helps our proposed method outperform several other model-based community-detection methods.

In the next section, we provide some background on existing models of information diffusion. Section III presents an overview of related work on community detection. Section IV-A introduces our proposed information diffusion model, Particle Swarm Optimization Diffusion Model (PSODM), and Section IV-B describes our community detection algorithm, GPSODM. We show our community detection results on both synthetic and real-world social networks in Section V, before concluding the paper.

II. Background

Information diffusion is often studied as a prerequisite of influence maximization. The problem is formally described as selecting the set of k nodes to target for initial activation such that it yields the largest expected spread of information, where k is constrained by a budget [1]–[3]. To study influence maximization, we require a deep understanding of the effects of micro and macro-level structures on information propagation. This in turn has focused attention on modeling and predicting information diffusion in social networks. These diffusion models are often inspired from research in various fields including epidemiology, sociology, marketing, and physics [4]–[7].

Two of the most widely used propagation models are the Linear Threshold Model (LTM) [8], which assumes that a node activates if the number of activated neighbors exceeds its private threshold, and the Independent Cascade Model (ICM), which assumes that when a node activates, each of its neighbors has a probability of activating, causing cascades of activation to sweep the network. Besides providing models for the propagation process, these techniques can be used for tasks like opinion leader detection [3]. Recently, Lahiri and Cerbian have shown that these popular models can be viewed as special cases of the more general Genetic Algorithm Diffusion Model (GADM) [9].

The main assumption of GADM is that individuals in the network communicate with the goal of increasing the value of their personal information as evaluated by a global objective function. The authors note that a key difference from previous models is that they can model multiple units of information using a hyperplane-defined function (HDF) [10] composed of multiple schemata. Each node in the network

ASONAM 2013, August 25-28, 2013, Niagara Falls, Canada
978-1-4799-1498-2/13/$31.00 ©2013 IEEE

is initialized with a binary string that represents the individual's personal information. An HDF is composed of multiple schemata for parsing binary strings and awarding scores; the overall fitness of the HDF is the sum of the schemata scores. Social interactions are emulated through a tail-swap cross-over interaction in which nodes exchange substrings; the resulting strings are retained if they have higher HDF scores than the parent string. Lahiri and Cerbian demonstrate that the final average normalized objective (ANO) of each network node as calculated by GADM cannot be simply predicted by network properties such as degree.

Although GADM is a very general model, it is not efficient at propagating information. The cross-over operation is inherently lossy, and social interactions are always assumed to be pairwise, rather than based on the local neighborhood. Inspired by GADM, we present a new method for modeling information diffusion processes in social networks using a stochastic optimization technique, particle swarm optimization, that has been successfully used to model the effects of aggregate social behavior. The use of PSO allows us to move from a binary string model of information storage to a more realistic real-valued vector. Each node of the underlying social network is considered as a particle in a swarm; the current information state of each node is represented by the particle position, encoded as a real-valued vector. Information exchanges are modeled by computing the relative value of a node's information compared to its local neighborhood using a modified HDF. This modification to the information diffusion model enables our community detection algorithm to outperform existing model-based community detection approaches, including the use of GADM [11].

III. Related Work

Detecting communities in real-world data, such as social networks, the WWW, and biological networks, is an important problem that has been attracting a great deal of attention in recent years [12]–[15]. A network community, sometimes referred to as a cluster, is typically thought of as a group of nodes with more and/or better interactions amongst its members than between its members and the remainder of the network.

Community detection provides us a valuable tool with which to better understand the function of complex networks. Many real-world graphs decompose naturally into communities where nodes are densely connected within the community but with much sparser connections between the communities. The communities from large networks carry great scientific and practical value because they typically correspond to functional units of the network, such as social groups or protein modules. A more extensive survey on community detection can be found at [13], [16]. Also, a useful listing of a large number of community detection methods appears in the supplementary material of [17].

Model-based community detection describes a broad category of methods that either consider the operation of a dynamic process using the network or the underlying generative model of network formation. Examples of dynamic processes are label propagation [18]–[20], diffusion flow, better known as the Markov Cluster Algorithm [21], and the popular spin model [22]. Community detection can also be cast as an inference problem [23], by assuming that some underlying probabilistic model, such as the planted partition model, led to the formation of the network community structure; community detection is performed by estimating the hidden parameters of this model. Other model-based approaches rely on the principle that a good clustering is determined by a low encoding cost, thus they perform community detection by finding the cluster structure that results in the lowest possible encoding cost. In this paper, we assume that communities are formed by human groups to optimize personal information flow and compare our method to several model-based community detection methods.

IV. Method

In this section of the paper, we provide the description and mathematical formulation for our proposed information diffusion model based on PSO (Particle Swarm Optimization Diffusion Model). The information flows discovered with this diffusion model serve as the input for our game-theoretic algorithm for community detection GPSODM (Game + Particle Swarm Optimization Diffusion Model). The implementation for these algorithms is publicly available at: http://www.eecs.ucf.edu/~alireza/psodm/.

A. Particle Swarm Optimization Diffusion Model

As discussed earlier, GADM suffers from information loss created by the simple one point cross-over model of social interactions. To address this, we introduce a model of information transmission that considers the effect of the entire local neighborhood on updates to an agent's personal information. Moreover, personal information is represented as a real-valued vector of knowledge to model a continuum of expertise.

A social network is a directed or undirected graph, $G = (V, E)$, where a vertex $v \in V$ represents an actor in a social network and an edge $(u,v) \in E$ represents an interaction between these two individuals. On the other hand, a *dynamic social network* is a multi-graph $G = (V, E)$, where E is a set of edges, and each timestamped edge $(u,v)_t \in E$ represents an interaction between u and v that occurred at time t. A *diffusion model* accepts a graph $G = (V, E)$, a *state vector* $S_v^{(t)}$ for every vertex $v \in V$ at time t as input. Based on the state of all interacting vertices, the model outputs a new state vector $S_v^{(t+1)}$ for every vertex at time $t+1$. For a dynamic social network, the graph at each timestep is defined as $G_t = (V, E_t)$, where $E_t = \{(u,v) : (u,v)_t \in E\}$ is the set of edges at timestep t.

A dynamic social network consists of a set of vertices $V = \{v_1, ..., v_n\}$ interacting over a period of T discrete timesteps. Each of these vertices are treated as particles by the PSO algorithm which we use to model information propagation. The standard PSO [24] algorithm maintains a population of candidate solutions known as particles and moves particles around in the search space to satisfy an objective function. The movements of each particle are based on the particle's position and velocity. Particle movement is influenced by two parameters: the best known position of the specified particle (*pbest*) and the best known position found by the population of particles (*gbest*). Velocity and position updates are done as

follows:

$$v[i+1] = v[i] + (C_1 R_1 \times (\text{pbest}[i] - \text{position}[i])) + (C_2 R_2 \times (\text{gbest}[i] - \text{position}[i]))$$

and

$$\text{position}[i+1] = \text{position}[i] + v[i+1]$$

C and R are the PSO algorithm parameters which can be tuned for performance.

To model the individuals within a social network as particles in a swarm, our proposed PSODM algorithm does the following:

1) Consider each node $v \in V$ in graph $G = (V, E)$ as a particle in PSO.
2) We assign a state vector with length β to each node to be used as its position by PSO.
3) Particles selfishly try to maximize their information state. Hence for every particle, *pbest* will be the current position of the particle, eliminating the first term in the update equation. A particle is only allowed to interact with its neighbors in the social network. As a result, *gbest* for a particle will be one of its neighbors.

The overall effect of the algorithm is to move each node towards the vector with the best information value in the local network. A node v's personal information vector is a string S_v which contains values in $[0, 1]$ and is initialized as:

$$S_v[i] = \text{random}(0, 1)$$

For example, the following table depicts a node state vector with length $\beta = 10$:

1.0	0.8	0.5	0.0	0.2	0.7	0.2	0.6	0.6	0.9

During the simulation of the information diffusion process, nodes interact and try to maximize their information level. The process of interacting and updating information is based on the PSO algorithm. In a normal PSO algorithm, we update particles' velocity and position iteratively. However, in the proposed algorithm, particles' positions are only updated when a new edge arrives at time t, based on the observed social media data.

Similar to GADM, Holland's *hyperplane-defined functions* (HDFs) are used to generate information schemata. An HDF is constructed by defining schemata (short substrings with wildcards that start at a specific position) randomly and hierarchically, starting with relatively short schemata of order 1 occurring at random starting positions within the string. Pairs of such schemata are concatenated to generate schemata of order 2, with each schema receiving an individual positive or negative score, also specified randomly from some range. Similarly, higher order schemata can be constructed using the same approach. HDFs take a binary string as input and return an objective score that is the sum of the scores of all the individual schemata contained within it. Obviously, we can only use the original style of HDFs to generate information schemata; the continuous state vector representation requires a new objective function. In order to find similarity between each node's state vector and the set of global information schemata, we perform the following steps:

1) Calculate the difference between each element of a state vector with its equivalent in the pattern generated by the HDF. More specifically, this difference for bit x is defined as $|0 - x|$ if the corresponding bit in the pattern is 0 and otherwise is simply x.
2) After computing all differences without considering the zero states (i.e., when two corresponding bits are equal), we calculate the score of an input string w.r.t a given schema. To do so, we use the product of the calculated differences as a weight w for that schema. By multiplying w and the given pattern's score s, the score S for the corresponding input string will be:

$$S_i = w s_i$$

We repeat the same process for every pattern generated by HDF leading to a total score of:

$$\text{total\_score} = \sum_{i=1}^{n} S_i \tag{1}$$

The performance of PSODM is not very sensitive to the choice of function; potentially other commonly used distance metrics could be employed to compute the objective function.

Information updates occur when a new edge arrives at time t and are performed as follows. First, we update the velocity of the particle:

$$\text{velocity}_v^{(t+1)}[i] = \text{velocity}_v^{(t)}[i] + (C_1 R_1 (\text{pBest}[i] - \text{position}_v^{(t)}[i])) + (C_2 R_2 (\text{gBest}[i] - \text{position}_v^{(t)}[i])) \tag{2}$$

Recall that $pBest$ always shows the current position of the node, so we have:

$$\text{velocity}_v^{(t+1)}[i] = \text{velocity}_v^{(t)}[i] + (CR \times (\text{gBest}[i] - \text{position}_v^{(t)}[i])) \tag{3}$$

Then, based on velocity we update the position which represents the information level of a node:

$$\text{position}_v^{(t+1)}[i] = \text{position}_v^{(t)}[i] + \text{velocity}_v^{(t+1)}[i] \tag{4}$$

Each time a pair of individuals interact, the information exchange is modeled by proposed PSO algorithm. This process can result an increase in a node information level. Using PSO enables us to model partial knowledge and avoid information loss which increases model verisimilitude. Algorithm 1 summarizes PSODM.

Algorithm 1 PSODM

Input: Initial state vector $S_v^{(t)}$ for node v
Output: New state vector $S_v^{(t+1)}$
position$_v^{(t)} = S_v^{(t)}$
gBest $= \emptyset$
Neighbors $=$ all neighbors of v
for $u \in$ Neighbors **do**
  if (position$_u^{(t)}$>position$_{gBest}$) **then**
    gBest $\leftarrow u$.
  end if
end for
Update velocity$_v^{(t+1)}$ based on (3)
Update position$_v^{(t+1)}$ based on (4)

B. Community Detection: GPSODM

Previous work showed that GADM paired with game theory can form a strong framework to identify the community structure of the underlying network [11]. Each node of the graph is modeled as a selfish agent with a utility function calculated from the information exchanged with neighboring agents. Agents opt to join or leave communities until they reach an equilibrium state. The Nash equilibrium of this game corresponds to the community structure of the network. In this framework, GADM is used to obtain the amount of information exchange between the nodes which in turn is used as input for the community detection algorithm.

To evaluate the performance of our diffusion model at the community detection task, we created a similar framework which uses PSODM to create the needed input for the community detection algorithm GPSODM (Game + PSODM, shown in Algorithm 2). Each vertex of our network is modeled as a selfish agent who locally maximizes its utility. Initially the utilities for all of the agents are initialized to zero. The community membership modulates an agent's information flow. Whenever an agent is sampled from the population, it has the opportunity to join a new community, leave one of its communities, or switch from a community to a new one, based on its current utility U. None of the other agents modify their state during this time period. If no utility increase is possible, the default is that the agents are restless and opt to join a new community. An agent's utility, U is calculated as:

$$U_i(S) = \frac{1}{m} \sum_{j=1, j \neq i}^{n} I_{ij}\delta_{ij} \tag{5}$$

where I_{ij} is the amount of information that agent i receives from agent j, δ_{ij} is equal to 1 if agents i and j are in the same community. Also, n and m show number of nodes and edges respectively.

Here S is set of strategies of all agents. In this framework, the best response strategy of an agent i with respect to the strategies S_i of other agents is calculated by:

$$\arg\max_{s_i' \subseteq [k]} U_i(S_{-i}, s_i') \tag{6}$$

The set of all feasible communities of the network is denoted by $[k] = 1, 2, ..., n$ where k is polynomial in n, however the number of our final communities may be much smaller than k. The strategy profile S forms a pure Nash equilibrium of the community formation game if all agents play their best strategies. In the Nash equilibrium no agent can improve its own utility by changing its strategy; that is each agent is satisfied with the current utility:

$$\forall i, s_i' \neq s_i, U_i(S_{-i}, s_i') \leq U_i(S_{-i}, s_i) \tag{7}$$

Since reaching global Nash equilibrium is not feasible in this game, we calculate a local Nash equilibrium [25]. The strategy profile S forms a local equilibrium if all agents play their locally optimal strategies. Here $ls(s_i)$ refers to local strategy space of agent i:

$$\forall i, s_i' \in ls(s_i), U_i(S_{-i}, s_i') \leq U_i(S_{-i}, s_i) \tag{8}$$

Algorithm 2 GPSODM

Input: underlying network graph G
Output: *communities* as a final division of G
$communities = \{\}$.
Agents = $\{agent_1$,$agent_2$,...,$agent_n\}$
repeat
  $agent_i$ = Random_Select(*Agents*)
  $action_i =$ Best_Operation (*join, leave, switch*)
  u_i'= Utility_Calculate($agent_i, action_i$)
  if $(u_i < u_i')$ **then**
    $u_i \leftarrow u_i'$.
    Update s_i
    Update *communities*
  else $action_i =$ *no Operation*
  end if
until (*local equilibrium* is reached)

V. Experimental Results

As stated before, PSODM can be used to model information diffusion in social networks. To illustrate this, we apply PSODM to simulate the information diffusion process on a real dynamic social network. This network was extracted from the Enron e-mail data set by Lahiri and Cerbian for their evaluation as described in [9]. We show the final results of both models (GADM and the PSODM) on this dataset to illustrate the differences between the two algorithms.

A. Information Diffusion with PSODM

The Enron e-mail dataset is a repository of e-mails exchanged between the executives of the former Enron Corporation. It is one of the largest public datasets of a corporate e-mail environment, and is naturally represented as a dynamic social network. Each e-mail sent by or to an executive can be considered as a directed and timestamped edge in a dynamic social network. Also, each email address in this dataset is a vertex in the corresponding graph. Lahiri and Cerbian parsed the headers of all e-mails and obtained 1,326,771 timestamped edges corresponding to individual e-mails sent between 84,716 e-mail addresses. The dataset contains 215,841 unique timestamps which non-uniformly cover a period of approximately four years. We apply both PSODM and GADM on the Enron Dataset to simulate the information diffusion process.

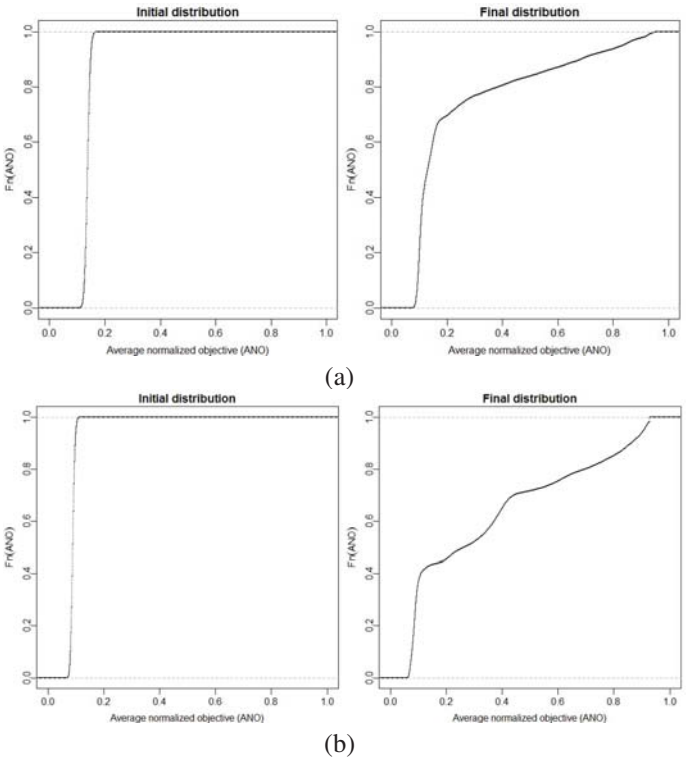


Fig. 1: Cumulative distribution of initial (left) and final (right) ANO values for the Enron e-mail dataset (a) Cumulative distribution of PSODM (b) Cumulative distribution of GADM. PSODM exhibits less information elitism due to the more efficient propagation model.

Setup. We start by generating random state vectors to initialize every node, along with a random HDF of arbitrary length. After PSODM was executed on timestamped edges, the final objective value (information level) of all nodes are normalized relative to the maximum objective value in the swarm population. A node can attain a higher objective value, if and only if it has accumulated a more valuable combination of information units compared with other nodes. As discussed in [9], in order to account for the latter bias, we run this algorithm with different HDFs and also new random state vectors for every node. We get the *average normalized objective* (ANO) for each node over many trials. This ANO represents the relative information value of an individual, and a particle with a higher ANO possesses more valuable information units.

Result. Figure 1 shows both initial and final cumulative distributions of the ANO values of all individuals in the network for both PSODM (top) and GADM (bottom). The initial cumulative distribution of the ANO values are strongly clustered and have very low dispersion for both models. As mentioned in [9], a similar final ANO distribution is expected if all individuals were comparable in terms of their network position, i.e., no vertex would consistently end with high normalized objective values from different starting states. However, it is obvious from both final distributions that this is not the case. We can see the final distributions of ANOs for both models contain a small number of vertices that consistently end with the highest-scoring schemata, regardless of their initial state. This feature of the network is called *information elitism*.

TABLE I: Parameters used in experiments

Algorithms	Parameters
GPSODM	$C = 2.0$
GGADM [36]	None
HA [40]	$m = 0.1, \delta = 0.05$
MMC [41]	$\alpha = 2.65, \beta = 2, \rho = 0.9, \mu = 1.08, \eta = 0.7, \gamma = 0.1$
LPA [42]	None
InfoMap [43]	None

As discussed by Lahiri and Cerbian, information elitism can not be explained by trivial graph features like degree. They show that there is no correlation between the final ANO values of the top 2% of vertices and several simple network features of each node, such as the in-degree (number of incoming email partners), the total number of emails received from all neighbors, and the time at which a vertex was first observed in the dataset. This phenomenon is reduced when we apply PSODM, as shown in Figure 1(a). In contrast to GADM, PSODM nodes exchange information with their best neighbor (*gbest*). By locally optimizing the information propagation process, information elitism in the network is substantially reduced.

B. Community Detection Algorithm (GPSODM)

Although the information diffusion results yield interesting insights, the aim of our research is to improve the community detection process. Here, we show the results for our community detection framework, GPSODM, compared with GGADM (Game + GADM) and four other state-of-the-art algorithms on three real world and two synthetic datasets. Two benchmark social media datasets from [26] are used to show the scalability of GPSODM. Two evaluation metrics are used to show the performance of our algorithm: Normalized Mutual Information (NMI) [27] and Modularity (Q) [28]. Table I summarizes the algorithms and parameter settings used for our experiments. All of the algorithms were implemented in Java and executed on a system with 4G of RAM and an Intel 2.53 GHz CPU. The following community detection approaches are used as benchmarks:

- **MMC** [29]: This algorithm is a label propagation method which uses labels to show information diffusion in social networks. Each node is able to communicate and exchange information with existing neighbors. By employing the inflation-like operation in the Markov clustering algorithm to update transmission probabilities, information flow will result in "dominant" information staying in each node, which can be used to determine community structure.
- **LPA** [18]: In this algorithm, every node is initialized with a unique label and at every step each node adopts the label that most of its neighbors currently have. During this iterative process densely connected groups of nodes form a consensus on a unique label to form communities.
- **HA** [19]: HA is an extended form of LPA. Suggested potential heuristics, presented in this work, can be applied to improve LPA average detection performance and adaptability. Also, by tweaking of parameters the new algorithm can be adapted to different types of networks.

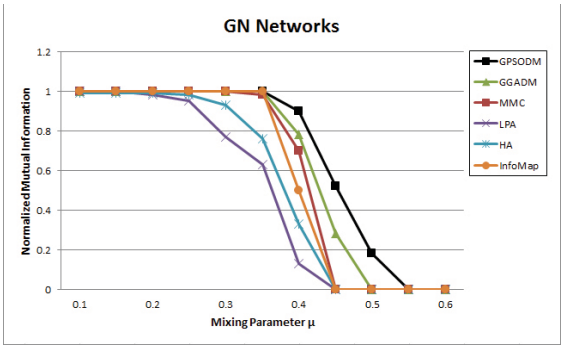
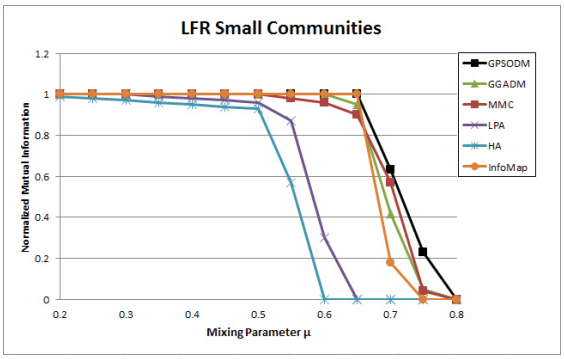
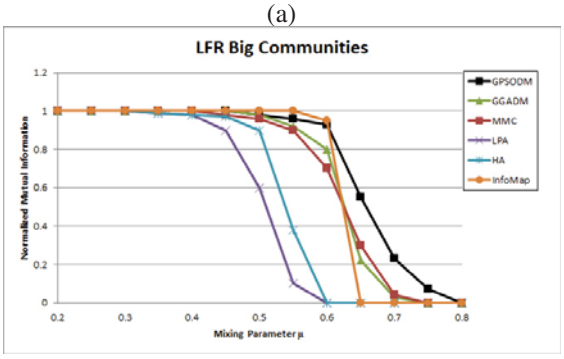


Fig. 2: Community detection (NMI) on GN Synthetic Networks. GPSODM outperforms the other methods in the more challenging cases with a high mixing parameter.



(a)



(b)

Fig. 3: Community detection (NMI) on LFR Synthetic Networks (a) Small communities (20-50) 10,000 nodes (b) Big communities (20-100) 10,000 nodes. GPSODM outperforms the other methods in the more challenging cases with a high mixing parameter.

- **InfoMap** [30]: InfoMap is a centralized method which is based on Minimum Description Length (MDL). It uses the probability flow of random walks on a network as a proxy for information flows in the real system and decomposes the network into modules by compressing a description of the probability flow. The result is a map that both simplifies and highlights the regularities in the structure and their relationships.

1) Evaluation: In order to measure the similarity between the identified community partition and the ground truth partition we can use *normalized mutual information* (NMI). However, when the ground truth is unknown, modularity can be used to evaluate the quality of the community detection. Modularity is the most popular qualitative measure for detecting communities in social networks, although this measure has drawbacks and becomes unreliable when our networks are too sparse.

TABLE II: Community detection results on real-world data (average of 100 runs)

	NMI			Modularity		
	Football	Dolphin	Karate	Football	Dolphin	Karate
	Football	Dolphin	Karate	Football	Dolphin	Karate
GPSODM	**1.000**	0.723	**1.000**	0.583	**0.580**	**0.412**
GGADM	0.910	**0.736**	1.000	**0.598**	0.544	0.381
HA	0.907	0.707	0.754	0.566	0.449	0.300
MMC	0.885	0.579	1.000	0.595	0.526	0.371
LPA	0.927	0.710	0.751	0.597	0.450	0.362
InfoMap	0.899	0.695	0.643	0.575	0.514	0.354

2) Datasets: Experiments have been performed using two synthetic and three real world networks. Results on these datasets demonstrate that our approach performs well on all datasets. We also investigate our results in terms of modularity and time complexity on social media datasets. First, we presents results on the GN and LFR synthetic networks in order to show that our method performs well on large datasets, since ground truth is only available for small real-world datasets.

GN Synthetic Networks: The GN synthetic network algorithm is probably the most popular community generation model [15]. This model contains fixed communities and 128 nodes in which each node has the same expected degree. The mixing parameter μ controls the ratio between the external degree with respect to community and degree of a node. Figure 2 shows the performances of all algorithms on a GN synthetic network. Every point in this figure is the average of 100 runs for each algorithm. GPSODM and GGADM detect only four communities in each run while other algorithms occasionally find more than four communities. As we can see in Figure 2, GPSODM performs outstandingly well when the value of mixing parameter is lower than 0.55. The NMI value approaches zero when $\mu = 0.55$ while this happens to other algorithms when $\mu \leq 0.5$.

LFR Synthetic Networks: Lancichinetti [31] proposed a method for generating synthetic networks while allowing control over degree distribution and community sizes. The LFR network generation method is popular since it is able to generate realistic networks with overlapping communities. The maximum degree of each node is capped at 50, and the average degree is set to 20 for this experiment. Also, the exponent of the degree distribution is set to -2. Community sizes range from 20 to 50 nodes (small) and 20 to 100 (large) respectively. Figure 3 shows the performance of different benchmarks compared with our algorithm, applied to two different networks with small and big communities. As can be seen in this figure, GPSODM performs better than all the other methods when $\mu \leq 0.8$ while LPA and HA reach zero at $\mu = 0.60$.

Finally we use three real world datasets, for which we have ground truth, to illustrate the performance of our algorithm. Since PSODM uses a more realistic information propagation model, it can achieve better results on these datasets. The

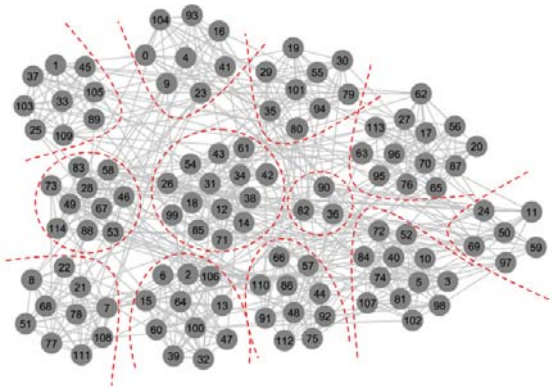


Fig. 4: The best community structure of American College Football network found by our method with NMI = 1.

TABLE III: Statistical properties of BlogCatalog and Flickr datasets

Dataset	# Categories	*# Nodes*	*# Edges*	Max Degree
BlogCatalog	39	10,312	333,983	3,992
Flickr	195	80,513	5,899,882	5,706

performances of a variety of algorithms on these networks for the Modularity and NMI metrics are shown in Table II.

American College Football Network: The American college football network represents the schedule of games during the 2000 season. This network was previously used by Girvan and Newman [14] and its community structure is known. The network contains 115 nodes and 616 edges. Each node represents a football team and each edge shows a game between two teams connected to each other. Teams are divided into 12 conferences, and we consider each conference as a community in this case. Games held between teams of the same conference are held more frequently than games played between different conferences. The results on this network indicate that while all the other approaches erroneously classify a few teams, such as Connecticut (42) and Navy (80) from IA Independent and Middle Tennessee State (63) from Sunbelt, GPSODM correctly identifies all communities. Figure 4 shows communities found by our algorithm on the American college football network.

Bottlenose Dolphin Network: The dolphin network represents the relationships of 62 bottlenose dolphins, introduced by Lusseau [32]. This network consists of 62 nodes and 159 edges. Based on the Lusseau observations, these dolphins were divided into two groups.

Zachary Karate Club Network: As a final test on real-world data, we turn to the Zachary karate club network [18] which consists of 34 nodes and 78 edges and shows relationships between the standing of club members. This network is widely used as a benchmark for community detection algorithms. The Zachary karate club network split into two roughly equally-sized groups due to a disagreement between the club administrator and the principal karate teacher, represented by node 34 and node 1 respectively. Our approach reveals these two groups perfectly.

Benchmark Social Media Datasets: The amount of data in social media datasets is increasing drastically, hence an important feature of a community detection algorithm is scalability and ability to show plausible results on large datasets. To evaluate this feature of GPSODM, two benchmark social media datasets [26] downloaded from ASU[1] are used: BlogCatalog[2] and Flickr[3]. Since there is no ground truth about the number and structure of the communities in these datasets, we only show the results of GPSODM and GGADM.

TABLE IV: Computational time of the two methods and the number of extracted communities in the BlogCatalog dataset.

Method	Time (Seconds)	#Communities
GPSODM	8,834	15,340
GGADM	5,430	15,645

TABLE V: Computational time of the two methods and the number of extracted communities in the Flickr dataset.

Method	Time (Seconds)	#Communities
GPSODM	53,570	128,591
GGADM	36,029	202,171

BlogCatalog: BlogCatalog is a social blog directory which permits its users to create their own blogs under a set of predefined categories. A blog in BlogCatalog can be described by features such as the listing category, blog level tags, snippets of the five most recent posts, and post-level tags. Bloggers can designate metadata for improved access to their blogs. Also, users can specify their relationships with other bloggers. A blogger's interests can be gauged by the categories he publishes his blogs in. This dataset consists of 39 categories with a reasonably large blogger pool. Each blogger lists his/her blog under 1.6 categories on average.

Flickr: Flickr is a content sharing website for photos that is accompanied by an online community platform. Users can create profiles, upload their own photos and subscribe to different interest groups. To create this dataset, 195 interest groups were selected randomly and users with only single connection were removed from the dataset. The statistical properties of BlogCatalog and Flickr are depicted in Table III.

[1] www.socialcomputing.asu.edu/pages/datasets
[2] www.blogcatalog.com
[3] www.flickr.com

Also, Table IV and Table V show the computational time (for an Intel Core i5 2.5GHz CPU) and the number of communities extracted for GPSODM and GGADM on these datasets. The results show that the more realistic information fusion diffusion model renders GPSODM slower than GADM but still reasonably fast on these larger datasets.

VI. Conclusion and Future Work

Information can shape public opinion, cause panic and riots, change the outcome of elections, or motivate customers to adopt a product. It is disseminated to members of the social network through both natural and electronic media and can be in oral or written form. Information is critical to social organization in human societies, and online communities are formed for the efficient dissemination of information through bulletin boards, newsgroups, and blogs.

In this study, we proposed an information diffusion model based on particle swarm optimization that can also be used to identify the community structure of the underlying network. In this framework each node of the graph is considered as a selfish agent who locally optimizes a utility function to increase its information share. Our results show that our method outperforms existing model-based community detection methods on both synthetic and real-world datasets. Furthermore, our proposed method can easily be implemented in distributed processing environments such as Hadoop or MapReduce. For future work, we plan to support overlapping concepts by means of a predetermined control function.

Acknowledgments

The authors would like to thank Hamidreza Alvari. This research was funded in part by NSF award IIS-0845159.

References

[1] P. Domingos and M. Richardson, "Mining the network value of customers," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2001, pp. 57–66.

[2] M. Richardson and P. Domingos, "Mining knowledge-sharing sites for viral marketing," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2002, pp. 61–70.

[3] D. Kempe, J. Kleinberg, and E. Tardos, "Maximizing the spread of influence through a social network," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2003, pp. 137–146.

[4] M. Cha, H. Haddadi, F. Benevenuto, and K. P. Gummadi, "Measuring user influence in Twitter: the million follower fallacy," in *Proceedings of International AAAI Conference on Weblogs and Social Media*, 2010.

[5] M. Goetz, J. Leskovec, M. McGlohon, and C. Faloutsos, "Modeling blog dynamics," in *Proceedings of the International Conference on Weblogs and Social Media*, May 2009.

[6] J. Leskovec, L. Backstrom, and J. Kleinberg, "Meme-tracking and the dynamics of the news cycle," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2009, pp. 497–506.

[7] D. Liben-Nowell and J. Kleinberg, "Tracing information flow on a global scale using Internet chain-letter data," *Proceedings of the National Academy of Sciences*, vol. 105, no. 12, pp. 4633–4638, 2008.

[8] M. Granovetter, "Threshold Models of Collective Behavior," *American Journal of Sociology*, vol. 83, no. 6, pp. 1420–1443, 1978.

[9] M. Lahiri and M. Cebrian, "The genetic algorithm as a general diffusion model for social networks," in *AAAI Conference on Artificial Intelligence*, 2010. [Online]. Available: http://www.aaai.org/ocs/index.php/AAAI/AAAI10/paper/view/1785

[10] J. H. Holland, "Building blocks, cohort genetic algorithms, and hyperplane-defined functions," *Evolutionary Computation*, vol. 8, pp. 373–391, 2000.

[11] A. Hajibagheri, H. Alvari, A. Hamzeh, and S. Hashemi, "Community detection in social networks using information diffusion," in *IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, 2012, pp. 702–703.

[12] A. Clauset, M. E. J. Newman, and C. Moore, "Finding community structure in very large networks," *Physical Review E*, vol. 70, no. 6, Dec. 2004.

[13] S. Fortunato, "Community detection in graphs," *Physics Reports*, vol. 486, p. 75, 2010.

[14] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," *Proceedings of the National Academy of Sciences*, vol. 99, no. 12, pp. 7821–7826, 2002.

[15] A. Lancichinetti and S. Fortunato, "Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities," *Physical Review E*, vol. 80, no. 1, p. 016118, 2009.

[16] S. E. Schaeffer, "Survey: Graph clustering," *Computer Science Review*, vol. 1, no. 1, pp. 27–64, Aug. 2007.

[17] I. A. Kovács, R. Palotai, M. S. Szalay, and P. Csermely, "Community landscapes: An integrative approach to determine overlapping network module hierarchy, identify key nodes and predict network dynamics," *PLoS ONE*, vol. 5, no. 9, 2010.

[18] U. Raghavan, R. Albert, and S. Kumara, "Near linear time algorithm to detect community structures in large-scale networks," *Physical Review E*, vol. 76, no. 3, 2007.

[19] I. Leung, P. Hui, P. Liò, and J. Crowcroft, "Towards real-time community detection in large networks," *Physical Review E*, vol. 79, no. 6, Jun. 2009.

[20] S. Gregory, "Finding overlapping communities in networks by label propagation," 2009.

[21] S. M. van Dongen, "Graph Clustering by Flow Simulation," Ph.D. dissertation, University of Utrecht, The Netherlands, 2000.

[22] J. Reichardt and S. Bornholdt, "Statistical mechanics of community detection," *Phys. Rev. E*, vol. 74, Jul 2006.

[23] M. B. Hastings, "Community detection as an inference problem," *Phys. Rev. E*, vol. 74, Sep 2006.

[24] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, vol. 4, 1995, pp. 1942–1948 vol.4.

[25] C. Alos-Ferrer and A. Ania, "Local equilibria in economic games," *Economics Letters*, vol. 70, no. 2, pp. 165–173, 2001.

[26] L. Tang and H. Liu, "Relational learning via latent social dimensions," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: ACM, 2009, pp. 817–826.

[27] L. Danon, A. Daz-guilera, and J. Duch, "Comparing community structure identification," *Journal of Statistical Mechanics: Theory and Experiment*, 2005.

[28] M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Phys. Rev. E*, vol. 69, no. 2, Feb. 2004.

[29] W. Chen, "Discovering communities by information diffusion," in *Proceedings of the International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*, vol. 2, 2011, pp. 1123–1132.

[30] M. Rosvall and C. T. Bergstrom, "Maps of random walks on complex networks reveal community structure," *Proceedings of the National Academy of Sciences*, vol. 105, no. 4, pp. 1118–1123, Jan. 2008.

[31] A. Lancichinetti and S. Fortunato, "Community detection algorithms: a comparative analysis," 2009. [Online]. Available: http://arxiv.org/abs/0908.1062

[32] D. Lusseau, "The emergent properties of a dolphin social network," July 2003. [Online]. Available: http://arxiv.org/abs/cond-mat/0307439