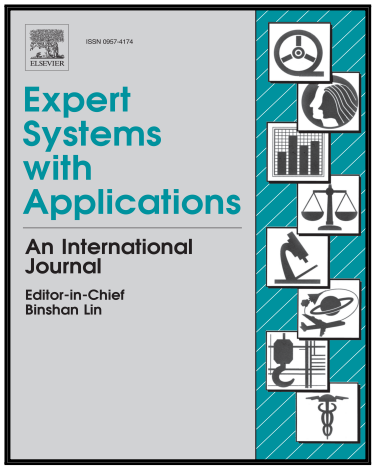
Journal Pre-proof



Node Importance based Label Propagation Algorithm for overlapping community detection in networks

Imen Ben EL Kouni, Wafa Karoui, Lotfi Ben Romdhane

PII: S0957-4174(19)30737-7
DOI: https://doi.org/10.1016/j.eswa.2019.113020
Reference: ESWA 113020

To appear in: *Expert Systems With Applications*

Received date: 25 January 2019
Revised date: 10 October 2019
Accepted date: 11 October 2019

Please cite this article as: Imen Ben EL Kouni, Wafa Karoui, Lotfi Ben Romdhane, Node Importance based Label Propagation Algorithm for overlapping community detection in networks, *Expert Systems With Applications* (2019), doi: https://doi.org/10.1016/j.eswa.2019.113020

Highlights

- We propose a label propagation method called NI-LPA.
- This method detects Overlapping communities on networks.
- NI-LPA is able to uncover logical partitions on real networks.
- Experiments are performed on complex networks
- NI-LPA obtains accurate results compared to several methods.

Node Importance based Label Propagation Algorithm for overlapping community detection in networks

Imen Ben ELKouni[a,b,*], Wafa Karoui[a,c] and Lotfi Ben Romdhane[a,b]

[a] Université de Sousse, Laboratoire MARS LR17ES05, ISITCom, 4011, Sousse, Tunisie
[b] Université de Sousse, ISITCom, 4011, Sousse, Tunisie
[c] Université de Tunis El Manar, Institut Superieur d'Informatique, 2080, Tunis, Tunisie

ARTICLE INFO

Keywords:
Complex networks
Communities
Overlapping
Label propagation
Node importance

Abstract

The enormous growth of the Web led to the birth of different network structures. Therefore, one of the important issues in the field of complex network analysis is to find and exploit the structure. Many studies have been carried out in this sense. The Label Propagation Algorithm (LPA) is among the most recognized approaches to detect disjointed communities. It is a simple and fast method, but its major disadvantage lies in its instability due to a random update. In this paper, we introduce a Node Importance based Label Propagation Algorithm (NI-LPA), a new algorithm for detecting overlapping communities in networks. As indicated in its name, NI-LPA is an improved version of LPA which maintains its simplicity and enhances its accuracy. In fact, we adopt the LPA strategy to allow a node to contain a set of labels. Moreover, the algorithm simulates a special propagation and filtering process using information deduced from the properties of nodes. Experimental results on artificial and real-world networks with different sizes, complexities and densities show the efficiency of our approach to detect overlapping communities.

1. Introduction

A complex system of nature, technology or society can be considered as a network. Especially with the progressive evolution of the web, its network structure has evolved to lead to a huge network, corresponding to large numbers of nodes and links. The existence of more densely connected areas than others is one of the common features of many networks. Detecting these areas called communities in complex networks has become a fundamental problem. Persons belonging to the same community have generally similar properties. Even if most existing approaches focus on the disjointed communities, people in the same network may have multiple relationships. Hence, we should address overlapping communities. Several algorithms have been developed to detect overlapping communities in complex networks. It is difficult to classify community detection algorithms. Therefore, in literature, many research scholars propose different categorizations. In (Orman and Labatut, 2009), the authors classify them in three different classes. The first class is the hierarchical approaches whose results are a dendrogram of communities. The idea is to divide or merge communities by considering a defined measure of similarity between them (Pons and Latapy, 2005; Clauset et al., 2004; Newman, 2004). The second class is the optimization approaches. This type of method uses given criterion, defined measure, or fitness function to estimate the quality of the partitions (Reichardt and Bornholdt, 2006; Lancichinetti et al., 2011; Duch and Arenas, 2005). The third class comprises all the remaining approaches which use different principles like agent-based methods, probabilistic, and density-based clustering.

In (Xie et al., 2013), the authors categorize the community detection algorithms into five families. First, the clique percolation method (CPM) (Palla et al., 2005) which is the most famous technique. The second category focuses on links partitioning where the idea is to detect groups of links instead of nodes. Third, the local expansion and optimization algorithms which select the central nodes, then, add consecutively and/or eliminate nodes within the community to improve the fitness function value. The fourth family consists of fuzzy detection algorithms to detect overlapping communities. One of the best-known algorithms in this category is FCM (Bezdek et al., 1984). Finally, agent-based algorithms which use the labels of nodes to determine communities as the Label Propagation Approach (LPA) proposed by Raghavan et al. (2007), called also the RAK (Raghavan, Albert, and Kumara) algorithm (Gregory, 2010b; Wen et al., 2014). This community detection algorithm is simple and fast with near linear time complexity. LPAs are such type of methods which exploit only the network structure to find communities. They require neither previous information about the communities nor a predefined fitness function. Their random update sequence expected some instability detection. Over the last decade, label propagation technology has been an active research direction. Thus, various modifications have been implemented to rectify its robustness and stability. But, there are very few algorithms that can uncover overlapping communities. In this context and in order to tackle the instability and improve the accuracy of LPA and to design an efficient LPA extension that is more stable to identify overlapping communities, we propose NI-LPA (Node Importance-Label Propagation Algorithm). Our model characterizes each node of the network with a weight to take into account its importance. We summarize the main contributions of this paper as follows:

*Corresponding author. Tel.: +216 56440830.
imenelkouni@gmail.com (I.B. ELKouni); karoui.wafa@gmail.com (W. Karoui); lotfi.ben.romdhane@gmail.com (L.B. Romdhane)
ORCID(s): 0000-0003-3240-9647 (I.B. ELKouni)

i We propose to sort nodes in a fixed order. This can solve the LPA instability problem and avoid the random selection of the node to be processed first. Also, it allows the algorithm to converge to a stable result.

ii Unlike traditional LPA, we propose a new LPA with extra information to detect groups using node importance. It conserves not only the simplicity and effectiveness of LPA but also introduces accuracy and robustness. The higher the importance of a node is, the higher the labels sent by this node have priority over the propagation phase.

iii Similar to traditional LPA, our algorithm finds communities using only network structure and requiring neither a fitness function to satisfy nor previous information about the communities. Besides, we propose to add a filtering step because we find that some labels are useless which can affect efficiency. These labels are those who have low coefficients compared to the other labels.

The remainder of the paper is organized as follows. Section 2 introduces the LPA process, its extensions, and the node and edge importance for clustering. Section 3 is the core of the paper: it describes our model NI-LPA. Sections 4 and 5 report experimental settings and results, respectively. The conclusion is given in Section 6.

2. Related Work

2.1. Label Propagation Algorithm

The literature on community discovery offers many different approaches. In this context, LPA was proposed as a simple, near-linear and fast algorithm for group detection. Its core idea is to assign each node in a network to the same community as most of its neighbors. The procedure of LPA can be described as follows:

1. Initialize each node in the network with a unique label.
2. Put the nodes of the network in random order and set it to X.
3. For each node chosen from the specific order X, set its label as the label occurring with the highest frequency among its neighbors.
4. If each node has the same label as most of its neighbors, then stop the algorithm. Else return to step (2).

In comparison, with other community detection algorithms, LPA relies only on local information that can be quickly computed. Each node makes its own decision about the community to which it belongs. Its decision is based on the communities of its neighbors. The label propagation process uses only the network structure to guide its progress and requires no external parameter settings. It requires neither a pre-defined fitness function nor prior information about the communities. The updating process can be done either synchronously or asynchronously. Synchronous updating approach is to change a node's label in the t^{th} iteration based on the neighbor's label at the $(t-1)^{th}$ iteration. Since synchronous updating may occur an oscillation phenomenon, asynchronous updating can fix this problem. Asynchronous updating is to update node labels in t^{th} iteration according to the labels of its already updated and not yet updated neighbors in the present iteration.

A major inconvenient is that LPA suffers from some instability detection because of its random update sequence. Therefore, label propagation technology has been an active research direction. Over the last decade, many research articles about LPA have been published and several ameliorations have been developed to improve its robustness and stability. So a great number of algorithms have been developed using label propagation techniques to detect either disjoint or overlapping communities.

2.2. LPA for disjoint communities

Many algorithms have been developed using label propagation techniques to detect disjoint communities. Zhang et al. (2014) propose a new LPA based on edge clustering coefficient. The node selects the neighbor associated to the highest edge clustering coefficient to update its label rather than a random neighbor node. Chin and Ratnavelu (2016) propose a constrained LPA named CLPA-GNR. This method detects the main communities and assigns nodes into groups using constrained LPA. Then, it is possible to improve the quality of the initial detected communities at various steps of the algorithm. It can remove a node from its actual community, add or switch it to another community. Zhang et al. (2017) propose LPA-NI algorithm based on label influence and node importance. LPA-NI has better accuracy, stability, and improves the quality of community detection. Berahmand and Bouyer (2018) propose LP-LPA algorithm which computes the link weights and the influence of node's label. This method adopts a defined descending order of influence value of the node's label. It selects the label with the highest influence value among labels. Zhang et al. (2018) propose a node weight-based label propagation strategy for large-scale complex network community detection. It selects the core nodes with more influence in the network, which depends of their similarities and their degrees. Also, the authors propose a weighted compactness function as fitness function. Jokar and Mosleh (2019) propose a balanced link density-based label propagation (BLDLP). This method calculates the weights for each edge based on the density of the links. This weight is the criterion that replaces the random selection of labels. If the maximum label of neighboring nodes is not unique, this method selects the label with the highest weight. Deng et al. (2019) improve label propagation and fuzzy C-means to detect communities. This algorithm assigns initial labels of nodes using a neighbor evaluation method. The labels with a large diversity in each community are revised by fuzzy C-means membership vectors. Then, the parameters are updated until an objective function is satisfied. However, most complex networks contain nodes that are characterized by multiple community memberships. A

node can belong to multiple communities and the number of communities to which it belongs is unlimited, hence overlapping communities. In (Kelley et al., 2012), the authors demonstrate that overlap is an important feature of many complex networks.

2.3. LPA for overlapping communities

Among the recent algorithms proposed to detect overlapping communities, COPRA proposed by Gregory (2010b) is the first method using the label propagation. This method adopts a synchronous updating strategy in the propagation step. Each node updates its belonging coefficients which depend of the coefficients over all its neighbors. Its time complexity is $O(v^3 n)$ per iteration where v is the maximum number of communities per vertex. In (Xie et al., 2011), the authors propose a fast Speaker-listener LPA. This method uses interaction rules to propagate labels between nodes. Each node has a memory to save received information, i.e., labels are received with their occurrence probabilities. Its time complexity is $O(Tn)$ where T is the maximum iteration constant. In (Wu et al., 2012), the authors propose a balanced multi-label propagation algorithm called BMLPA. In this algorithm, the initialization of labels takes benefits from a new method to generate "rough cores". Furthermore, BMLPA proposes a new update strategy with balanced belonging coefficients to find overlapping groups in networks. Its time complexity is $O(n \log n)$ per iteration. In (He-Li et al., 2015), the authors recommend a dominant label propagation algorithm to identify groups. This algorithm simulates a special voting process to detect simultaneously disjoints and overlapping community in complex networks. Its time complexity is $O(m)$ with m the total number of edges. In (Tong et al., 2015), a weighted LPA is used by authors to find overlapping community. This method assigns to each label a weight to measure node importance based on degree centrality. Also, the nodes are arranged in descending order according to their importance to improve the stability. In (Liu et al., 2016), the authors improve LPA and propose an algorithm called LPPB. This method uses label propagation probability to discover overlapping communities. In (Sun et al., 2017), the authors propose LinkLPA. This method transforms the node partition problem into the link partition problem. Then, it uses a new LPA with a preference on links instead of nodes to detect communities. Its time complexity is $O(m + cn)$ with m the number of edges and c the number of clusters before merging. In (Chen et al., 2017), the authors introduce information entropy as the measurement of the relationship between direct and indirect neighbors. This algorithm fixes a threshold to keep one or more labels to constitute an overlapping community. Its time complexity is $O(n * d^2)$ with d is the average degree of the network. In (Lu et al., 2019), the authors propose an LPA with neighbor node influence called LPANNI. This method puts the nodes in the ascending order of node importance and propagates labels. The update of the label is based on its neighbor node influence and historical label preferred strategy. Its time complexity is linear complexity $O(n)$.

2.4. Node and edge importance for clustering

Since the paper focuses on the LPA model, it concentrates on nodes' importance for clustering purposes and community detection. Besides, relevant works exist on clustering by first calculating edge importance and using such information for clustering purposes. Although, modularity has some problems, such as the resolution limit (Fortunato and Barthélemy, 2007). In (Berry et al., 2011), the authors propose a weighted modularity maximization to improve the accuracy on many real networks. CNM algorithm computes the change in modularity associated with all possible mergers of two existing communities. The optimal weighting would assign 1 to each intra-clique edge and 0 to each connecting edge. Then, they use local computations to derive new edge weights and reward an edge for each short cycle connecting its endpoints. However, the number of modules finding is an important point to improve the results of community detection. In (Meo et al., 2014), the authors propose a K-path edge centrality algorithm to calculate edge importance and to compute distances between nodes in the graph. They use multiple random walks to simulate the propagation of a message between nodes. After this, they use these distances in conjunction with the Louvain Method (Blondel et al., 2008) to identify communities in a graph. This approach improves the modularity measure, but it still suffers instability. In (Ouyang et al., 2018), the authors define the most important edges to be the ones which, if removed, would break down the network to the most extent. The proposed importance measure, nearest-neighbor connectivity based edge importance (NNCEI), can quantify the importance of a single edge or a set of edges. With the notion of nearest-neighbor connectivity, one can write the probability for any connected sub-networks. The proposed measure is more efficient compared with some widely adopted measures, by removing the most important edges according to each measure, and examining the size of the giant component.

A lot of algorithms have been developed to tackle the community detection problem, but this issue is still open. Until now, this field suffers many problems to identify the appropriate partition for a network. Besides, some methods follow a random order or make random choices, and a few of these algorithms take into account the effect of a node's importance and its neighborhood. These limits interrupt the accuracy and quality of detected partitions, particularly with the complex nature of the large networks. In this paper, we propose a new algorithm called NI-LPA based on label propagation to detect overlapping communities. We describe our proposed method in section 3.

3. Node Importance-Label Propagation Algorithm

3.1. Problem formulation and basic definitions

Given a graph represented by $G = (V, E)$ where:
V is the finite set of vertices of G and E represents the set of edges of G.

Overlapping communities detection in G is defined as a partition $P = \{C_1, C_2, ..., C_c\}$ where these classes may be joined to each other. The vertices within the same class are strongly connected, the vertices of different classes are poorly connected, and an overlapping node belongs to over one class. Otherwise, a class or a community is a group of nodes that share common properties. The nodes of the same group are densely linked to each other and less connected to the other nodes of the network.

Definition 1. *"Neighbor"*
Given a graph $G = (V; E)$, two vertices u and v linked by a link $(u, v) \in E$ are called neighbors. (Souam et al., 2014).

Definition 2. *"Neighborhood"*
In the graph $G = (V; E)$, the neighborhood of a node v is the set of all its adjacent nodes noted $N(v)$ and defined as:

$$N(v) = u \in V \mid (v, u) \in E \quad (1)$$

Definition 3. *"Clustering coefficient of node"*
The node clustering coefficient also called the agglomeration, connection, grouping, aggregation or transition coefficient, is a measure of the grouping of nodes in a network (Watts and Strogatz, 1998). More precisely, this coefficient measures how close the neighborhood of a node is. This value is between 0 and 1 (1 if the neighborhood is well connected, 0 if there is almost no connection between neighbors). In the following, $cfc(v)$ is the clustering coefficient of node v. Thus, for a node v of degree greater than or equal to 2, we have:

$$cfc(v) = \frac{2 \mid e_{jk} \mid}{\mid B(v) \mid (\mid B(v) \mid -1)}; v_j, v_k \in V, e_{jk} \in E \quad (2)$$

where $\mid B(v) \mid$ is the degree of v and e_{jk} is the edge that connects the node v_j to v_k.

Definition 4. *"Overlap between two communities"*
This is the number of nodes shared between two communities C_i et C_j (Rhouma and Romdhane, 2014).

$$O(C_i, C_j) = \{v | v \in C_i and\ v \in C_j\} \quad (3)$$

Definition 5. *"Label"*
Each node of the network has a specific identifier. It is useful for indicating the community to which this node belongs (Raghavan et al., 2007).

3.2. Node importance

The original LPA and some of its proposed extensions consider that nodes have the same importance (Gregory, 2010b; Xie et al., 2011; Xie and Szymanski, 2013; Chin and Ratnavelu, 2016). However, other extensions associate a weight to each node to express its importance (Xing et al., 2014; He-Li et al., 2015; Zhang et al., 2017). In fact, certain properties can characterize each node such as its degree and its neighborhood. These properties can be used to give additional information to each node in the graph.
In the propagation step, each vertex v updates its label and replaces it by the label of the greatest number of neighbors. We must agree that the decision to select the random or the majority label for a node among many labels received by its neighbors may affect obtained communities. This step is sensitive, especially when we work on overlapping community detection, i.e. lets a vertex label be a set of community identifiers (over one label in each node). This means it is illogical to receive labels sent by many nodes and treat them with the same importance.
In our method NI-LPA, we are interested in node features. The importance of node v in a graph is a measure that plays an essential role. It calculates the tendency of v to be the priority source. The higher the importance of a node is, the higher the labels sent by this node have priority over the propagation phase. Generally, when the neighbors of a node v are densely connected, v is the appropriate choice to be the *influential node*. We have observed that the node importance rises proportionally with the increase of its neighborhood and when these neighbors are well connected.

The importance of v, denoted NI, is calculated as follows:

$$NI(v) = \begin{cases} \deg(v) * cfc(v) & \text{if deg(v)>1} \\ 0 & \text{otherwise.} \end{cases} \quad (4)$$

where $cfc(v)$ is the clustering coefficient of v and $deg(v)$ its border size.

Obviously, both the clustering coefficient and the node degree determine node importance, since nodes with the same degree may not play the same important role in a complex network. What is more, the clustering coefficient decreases linearly with the degree (Ravasz and Barabási, 2003). In effect, nodes associated to a high degree have a small clustering coefficient. In other words, when a person has many friends, these friends have fewer relations among them. As well, a node with more neighbors is likely to be embedded in relatively fewer closed triplets, and therefore to have a smaller local clustering than a node connected to fewer neighbors (Opsahl and Panzarasa, 2009). In contrast, according to (Albert-László and Réka, 1999), the clustering coefficient is independent of node degree. For example, a star subgraph with a degree of 3 has a clustering coefficient equal to 0. To conclude, it is not easy to affirm anything about the relationship between node degree and clustering coefficient. In our method, we choose to combine the two concepts: the node degree and the clustering coefficient, de-

scribed in the sequel.

3.2.1. Node degree

The node degree is important information to characterize a graph node. It reflects the ability of the node to establish a direct connection with the other nodes. The nodes with the largest degrees in each group are effectively hubs that can be used to route to larger portions of the network (Hansen et al., 2011). The ones who have connections to many others might have more influence or more access to information than those who have fewer connections. In networks, it is interesting to focus on who has the most connections, the most friends, or who propagates information very fast.
Note that for some nodes, the node importance may tend to the node degree. In this case, the local clustering coefficient is close to 1, which means that this node forms a clique with its neighbors, that these nodes are densely connected, and that probably, they will be assigned to the same group. However, the degree indicator only uses the node's own information, and does not consider the location of the node in the graph, nor the connections between its neighbors, the clustering coefficient can reflect neighborhood connectivity to a certain extent.
The local clustering coefficient is only defined for nodes whose degree is larger than one and a value close to 0 means that there are any connections in the neighborhood. Therefore, it is reasonable to assume that the node with an importance value close to 0 can't quickly propagate the information or form a dense group.

3.2.2. Local coefficient of clustering

It is established that the degree of a node is not a deciding factor that impacts node importance. The nodes with the same degree can be considered with the same importance but in reality, the information about their neighbors can affect their importance in the network. The deciding factor is the number of connected sub-graphs formed between neighboring nodes. The information of agglomerations beside the degree of a node is very important to express its importance. The clustering coefficient assesses the connectivity in a node's neighborhood: a node has a high clustering coefficient if its neighbors tend to be directly connected with each other (Costantini and Perugini, 2014).

Fig. 1 shows an example. The labels are represented by the colors, and the purpose here is to select the appropriate label of the central node. Each node sends its label with its calculated importance value. The central node receives all the labels sent by its neighbors (in this example: red label with 3/2, blue with 2, purple with 0 and green label with 3/5). So, we notice that the blue node has the most important influence in this step, followed by the red node.

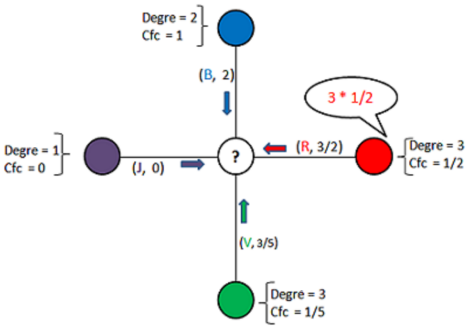


Fig. 1: Label selection of the central node

3.3. Description of our method NI-LPA

We propose an efficient algorithm, named NI-LPA, which uses label propagation techniques to detect overlapping communities. In fact, NI-LPA is described in Algorithm 1. It consists of three main components: initialization, propagation, and filtering.

Initialization

Firstly, we consider each node as a community. Similar to basic LPA, each node is initialized with a unique label also called an identifier. This label identifies the community to which this node belongs. Then we calculate the *node importance* of all vertices of the graph. If communities overlap, each vertex may belong to more than one community. Therefore, to find overlapping communities, we must allow a node to contain many community identifiers. In fact, we associate a set of pairs (c, b) at each node x where c is a community identifier, and b is the confidence coefficient of this direct neighbor. This coefficient b indicates the belonging coefficient of x to the community c and the importance of a node which sends this label. The idea behind the use of this coefficient is to associate a weight to each label. In general, the higher this measure is, the higher this label will be dominant in the propagation phase. Then we put nodes in a deterministic order.

Propagation process

This stage of the algorithm is described in Algorithm 2. This is the central phase of NI-LPA that contains multiple iterations. NI-LPA uses the asynchronous mode to avoid the oscillation problem (Cordasco and Gargano, 2010). To update node label in iteration t, this process is based on its neighbors' labels in iteration $(t-1)$ and the updated labels in the same iteration t. Therefore, we propose to sort the nodes in a fixed order to solve the LPA instability problem, avoid the random selection of the node to be processed first and allow the algorithm to converge to a stable result, unlike LPA which gives distinct results according to the processing order of the nodes.
Firstly, we select the node v from vector $Vect$. The processed node v receives sets of labels sent by its neighbors. A function $b_t(c; x)$ is applied as we show at line 5 of Algorithm 2. This function aggregates the coefficients of the labels associated by the neighbors of v at the iteration t and $(t-1)$.

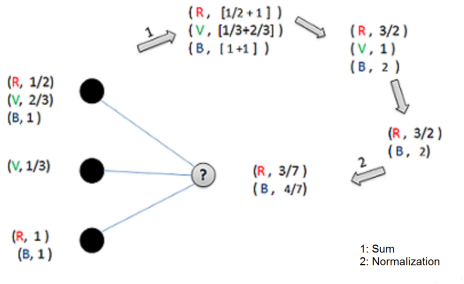


Fig. 2: Propagation step

$$b_t(c, v) = \sum_{y \in N(v)} (b_{(t-1)}(c, y) + b_t(c, y)) \quad (5)$$

At this level, two cases are possible:

- First case: if the node v contains the label c received with a determined coefficient then, we associate the sum of coefficients at the same label c.
- Second case: if the node v does not contain the label c among the set of labels already established, then, the couple (c, NI(c)) is added to its set.

At iteration t, when a coefficient is the highest compared to the other coefficients, this can be interpreted in two ways:

- This label is sent by more than one source node.
- This label is sent by an important source node.

Subsequently, unnecessary labels, i.e. labels with low coefficients, are eliminated, and we keep a subset that contains the labels with the K best-belonging coefficients. What is required is a way to retain more than one community identifier in each label without keeping all of them. We choose to set the K parameter to 2. The retained coefficients are normalized. Fig. 2 shows an example, when the treated node receives a set of labels (R, V and B) from its neighbors, then we sum the coefficients of the same label. The set becomes label 'R' with 3/2; label 'V' with 1 and 'B' with 2. Finally, we store labels with the best coefficients ('R=3/2'; 'B=2') and normalize them (see Algorithm 4).

Filtering process

This step is summarized in Algorithm 3. At the end of the propagation phase, each node contains a list of pairs (Label, Belonging coefficient). Among these labels, we find some that are useless because their coefficients are very low compared to the coefficients of the other labels. We delete the pairs with belonging coefficients less than some threshold. We choose in the rest of this work to set the threshold at 0.4. The choice of this threshold value is explained by the fact that any coefficient less than 0.4 is considered of minor importance.

Note that the threshold is used only in the filtering. It means that the process of NI-LPA is completely determined by the network structure.

Algorithm 1: NI-LPA

Data: A network $G = (V, E)$, number of iterations Nb
Result: Communities $P = \{C_1, ..., C_n\}$

Begin
  $\mathbb{P} \leftarrow \emptyset$
  Initialize each node with a unique label $C_x = x$
  Calculate importance of all the nodes
  Save nodes on the vector $Vect$
  Sort nodes according to their degree in descending order
  for *i=1 To Nb* **do**
    while *there are vertices in* $Vect$ **do**
      foreach *node* $v \in Vect$ **do**
        $P_1 \leftarrow$ **Propagation**(v, L_v)
      end
    end
    Delete v from $Vect$
  end
  $P \leftarrow$ **Filtering** $(P_1$, threshold)
End

Algorithm 2: Propagation

Data: Node v, Node List L_v
Result: Node v updated

Begin
  foreach *u in Neighbor(v)* **do**
    if *label sent by u exist in List of v* **then**
      Apply equation (5)
      Sum the coefficients of the same label
    else
      Add the new label with its coefficient to List of v
    end
  end
  Save the labels with high coefficient in the set
End

Algorithm 3: Filtering

Data: Overlapping communities P, Threshold
Result: Overlapping communities P

Begin
  Normalization (P)
  foreach *Label c with coefficient b in* P **do**
    if $b < Threshold$ **then**
      Delete label c
    end
  end
End

Algorithm 4: Normalization

Data: Node List L_v
Result: Node List L_v normalized
Begin
  $S \leftarrow$ Sum the coefficients of all the nodes **foreach** *label c with coefficient b in* L_v **do**
    Divide b by S and save this label as $(c, \frac{b}{sum})$
  end
End

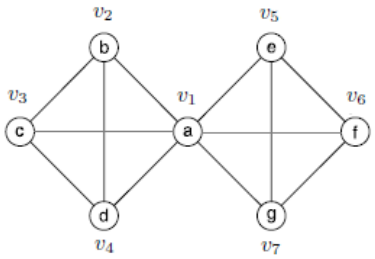


Fig. 3: Initialization of the network $G1$

3.4. Example

As an example, let consider the network $G1$ composed of seven nodes identified by letters from a to g (see Fig. 3). Initially, each node is identified by a unique label, and the belonging coefficient is set to 1.
Next, we compute the importance of each node, as shown in Table 1. Based on Table 1, each node is marked with a set of pairs (c; b) where c is a community identifier, and b is the importance of the neighbor who sent c. For example, neighbors of the node v_7 are v_1, v_5, and v_6. This node v_7 receives the labels of nodes v_1, v_5, and v_6 with their importances.
"v_7's labels": {(a, 2.4), (e, 3), (f,3)}

After preparation, the update rule is executed iteratively.
First, arrange nodes in descending order.
Second, we update the labels of nodes. Columns 2 and 3 in Table 2 denote the propagation step. We begin with node v_7 that will receive all the labels of its direct neighbors.
"v_7's labels": {(a, 7.2); (b, 3.0); (c, 3.0); (d, 3.0); (e, 9.0); (f, 9.0); (g, 9.0)}.
Then, we keep a subset of labels and eliminate unnecessary labels (label with low coefficients).
"v_7's labels": {(e, 9.0); (f, 9.0); (g, 9.0)}.
Node "v_6" updates its labels based on the labels of its neighbors updated in the current iteration and those that are not yet updated.
"v_6's labels": {(a, 4.8); (b, 3.0); (c, 3.0); (d, 3.0); (e, 6.33); (f, 6.33); (g, 9.33)}
After the elimination of labels with low coefficients:
"v_6's labels": {(e, 6. 33); (f, 6. 33); (g, 9.33)}
In the same way, we can update the labels of the other nodes with their belongings coefficients. Column 3 (iteration 2) in Table 2 shows the results after the second iteration. At the end of the algorithm, we normalize, compare the coefficients to a threshold and eliminate labels associated to coefficients that are insignificant relative to the threshold
The threshold is fixed to (0.4). Two overlapping communities C_1={v_1,v_2,v_3,v_4} and C_2={v_1,v_5,v_6,v_7} are identified. Node v_1 is identified as an overlapping node because it contains two labels (d and g) as shown in column 5 in Table 2.

Table. 1
Characteristics of nodes

Node	v_1	v_2	v_3	v_4	v_5	v_6	v_7
Degree	6	3	3	3	3	3	3
CC	2/5	1	1	1	1	1	1
Importance	12/5	3	3	3	3	3	3

4. Experimental settings

4.1. Experiment datasets

We use two types of dataset, including real-world and synthetic networks, to demonstrate the efficiency of our suggested NI-LPA algorithm. All the used datasets are undirected and unweighted.

- Real-world networks (see Table 3) are well-known in the literature and are used to check community detection algorithms' efficiency. We select models of both small and large sizes, but not all considered models have a ground truth partition structure for comparisons.

- LFR benchmark (Lancichinetti, 2013) is one of the most popular synthetic networks in the field of community detection. It is known to generate heterogeneous distributions of community sizes and node degrees. Furthermore, the graph's size and the density of its connections have a significant impact on the algorithm's efficiency. For the LFR benchmark, the generation of a network depends on specific parameters. N contains the number of nodes, K is the average degree, μ indicates the mixing parameter, $maxk$ is the maximum degree, community size is between $minc$ and $maxc$, O_n is the number of overlapping nodes and O_m specifies the number of memberships of overlapping nodes.

  In this work, we set $k = 10$; $maxk = 50$; $minc = 20$; $maxc = 100$ and we generate eight networks with different sizes, including 1000, 2000, 8000, 10.000, 50.000, 80.000, 100.000 and 150.000.

  The experiments are conducted on these generated networks, while we vary μ, O_n and O_m to create harder tests. Since the LFR benchmark is probabilistic, we

Table. 2
Definition of new labels for the network $G1$

Labels set	NI-LPA (Iteration 1)	NI-LPA (Iteration 2)	Normalization	Filtering
L_{v_1}	(d, 4.26) (g, 4.26)	(d, 2.34) (g, 2.39)	(d, 0.5) (g, 0.5)	(d, 0.5) (g, 0.5)
L_{v_2}	(c, 6.62) (d, 6.76)	(c, 1.24) (d, 2.25)	(c, 0.36) (d, 0.64)	(d, 0.64)
L_{v_3}	(b, 6.34) (c, 6.34) (d, 9.33)	(c, 1.17) (d, 2.04)	(c, 0.36) (d, 0.64)	(d, 0.64)
L_{v_4}	(b, 9) (c, 9) (d, 9)	(c, 1.12) (d, 1.76)	(c, 0.39) (d, 0.61)	(d, 0.61)
L_{v_5}	(f, 6.62) (g, 6.76)	(f, 1.24) (g, 2.25)	(f, 0.36) (g, 0.64)	(g, 0.64)
L_{v_6}	(e, 6.34) (f, 6.34) (g, 9.33)	(f, 1.17) (g, 2.04)	(f, 0.36) (g, 0.64)	(g, 0.64)
L_{v_7}	(e, 9) (f, 9) (g, 9)	(f, 1.12) (g, 1.76)	(f, 0.39) (g, 0.61)	(g, 0.61)

Table. 3
The characteristics of real-world networks

Network	$\|V\|$	$\|E\|$	Description
Karate club (Newman, 2013)	34	78	*The Zachary Karate Club*
Books (Newman, 2013)	105	441	*The network of American Politics Books*
PGP (Arenas, 2013)	10680	24316	*Pretty-Good-Privacy*
Ca-CondMat (Newman, 2013)	40421	175692	*Collaboration network of Arxiv Condensed Matter*

repeat the experiments many times (30 times) and report the mean to evaluate the real accuracy of our proposed method. The ground truth partition structure is given.

We test the efficiency of our algorithm in the experimental phase with different tasks known in the literature (Xie et al., 2011, 2013; Rhouma and Romdhane, 2014). We compare NI-LPA with four methods: the clique percolation algorithm CPM (Palla, 2011); COPRA (Gregory, 2010a) which uses label propagation technique; GCE, a greedy approach (Lee et al., 2013) and DOCNET, a local optimization algorithm.

All NI-LPA[1] algorithms are implemented in Java and tested in JDK8.0 platform, and simulations are performed on a notebook PC with Inter(R) Core(TM) CPU i3-2370M @ 2.40 GHz and 4 GB memory under Windows 7 OS.

4.2. Evaluation metrics

4.2.1. Normalized mutual information (NMI)

The normalized mutual information, NMI, is a common similarity metric between two partitions. It is applied to compare the partition detected by the algorithm to the partition one wishes to recover (McDaid et al., 2011). It allows knowing if a node resides in a good community. Also, this partition comparison measure takes the topological characteristics of both networks into account. In our work, we are looking at overlapping, a node can belong to over one group. Therefore, we present the measure NMI suggested by Lancichinetti et al. (2009) that can be used to compare divisions of a network into overlapping groups, i.e. covers of overlapping clusters. NMI ranges from 0 to 1 and equals 1 in the perfect case, i.e. the two partitions are identical. Given two partitions $X = \{X_1; X_2; :; X_i\}$ and $Y = \{Y_1; Y_2; :: Y_j\}$ of a graph G with i and j are the number of clusters. This function calculates the similarity between two random variables. Normalized mutual information is described as follows:

$$I_{norm}(X : Y) = \frac{H(X) + H(Y) - H(X,Y)}{(H(X) + H(Y))/2} \tag{6}$$

where H(X), (H(Y)), is the entropy of the random variable X, (Y), assigned to the partition $C\prime$, ($C\prime\prime$), whereas H(X, Y) is the joint entropy. This variable is in the range [0, 1] and equals 1 only when the two partitions $C\prime$ and $C\prime\prime$ are exactly coincident.

4.2.2. Overlap modularity

Modularity is a widely used measure to describe the partitioning quality of graph nodes into clusters. Originally, Newman and Girvan (2004) suggested it for non-overlapping community structures. Later, modularity is mainly used in complex network analysis as the evaluation criteria on real networks when the ground truth is unknown (Fortunato, 2010; Gregory, 2010b; Steinhaeuser and Chawla, 2010; Rhouma and Romdhane, 2014; Zhang et al., 2017). Newman's modularity is a function that evaluates the relative density of edges between and within communities. However, in the case where the nodes may belong to more than one group, i.e. communities overlap, Shen et al. (2009) suggest an extension of the classical modularity. This function is based on the number of overlapping memberships to measure the performance of overlapping community detection algorithms. A good modularity value indicates a significant network partition and modularity equals to 0 implies that all nodes belong to the same community.

In this paper, we adopt the extended modularity which is expressed as follows:

$$Q_{ov} = \frac{1}{2m} \sum_{c} \sum_{i,j \in C_c} [A_{ij} - \frac{K_i K_j}{2m}] \frac{1}{O_i O_j} \tag{7}$$

[1] https://github.com/imenkn/NI-LPA-code

Table. 4
NMI results for test with different threshold

Networks	0.2	0.3	0.4	0.5	0.6
N=2000	0.77	0.89	**0.95**	0.88	0.78
N=5000	0.87	0.90	**0.98**	0.94	0.83

Given a cover of a network, let m represents the number of links in the graph. A_{ij} is the element adjacency matrix. It takes 1 if there is a link between nodes i and j. K_i and K_j are, respectively, the degrees of nodes i and j. O_i and O_j represent the numbers of communities to which the nodes i and j belong respectively.

5. Numerical results and discussion

5.1. Synthetic networks

In this section, we examine our algorithm NI-LPA in terms of accuracy and stability. Therefore, to evaluate its performance, we create different types of a test while we vary network complexity (see Section 5.1.4), density (see Section 5.1.5), and size (see Section 5.1.6). Also, NI-LPA is compared to other extensions of LPA in the sequel (see Section 5.3).

5.1.1. Threshold definition

In our method, in the filtering phase, we use a threshold in the filtering step to eliminate labels with low coefficients, and we choose to set it to 0.4. This value's selection is clarified by the fact that each coefficient lower than 0.4 is considered low. So, in this section, we try to verify experimentally this hypothesis. We test our algorithm with distinct thresholds from 0.2 to 0.6. The parameters of networks are as follows: μ= 0.2; k=10; O_m=2; O_n=10%.
According to the results of Table 4, we see that our method is more efficient to detect overlapping communities when the threshold is set at 0.4 for the two network samples.

5.1.2. Degree centrality vs. Node importance

In this section, we experiment the effect of using the clustering coefficient beside the node degree index instead of node degree. In Table 5, we record the NMI mean values (denoted as *NMI*) and the standard deviations (denoted as *SD*) of networks with distinct parameters.
Experimentally, we can conclude that the clustering coefficient improves the stability of this method and provides high NMI values compared with node degree. In fact, the combination of node degree and clustering coefficient improves the community structure detection.

5.1.3. Results effectiveness verification with NMI

Since the LFR benchmark is probabilistic, we should repeat the experiments many times and report the mean and standard deviations in order to evaluate the accuracy of our suggested technique. There are studies which demonstrate that the performances of the classifiers measured across the same datasets use preferably the same splits into training and testing sets (Demsar, 2006). Therefore, we repeat the tests 32 times with networks of two different sizes (N=2000 and N=5000). Table 6 shows the results. We know that the more dispersed the distribution is, the less the values are concentrated around the mean, the higher the standard deviation will be.
According to the obtained results, we show that the standard deviations are near to 0 which implies that the values are very little dispersed around the mean. We can therefore conclude that NI-LPA provides stable outcomes and solves LPA's randomness problem.

Table. 5
NMI results for networks with 2000 and 5000 nodes

Network		Importance node		degree centrality	
Size	parameters	NMI	SD	NMI	SD
2000	mu=0.2	0.98	0.007	0.96	0.02
	ON=50%	0.60	0.009	0.48	0.03
	OM=4	0.88	0.01	0.84	0.02
5000	mu=0.2	0.99	0,002	0.97	0,02
	ON=50%	0.63	0,07	0.59	0,05
	OM=4	0.89	0,003	0.82	0,03

Table. 6
NMI results for effectiveness verification of NI-LPA

Networks	O_n	NMI mean	standard deviations
N=2000	10%	0.98	0.004
N=2000	50%	0.60	0.05
N=5000	10%	0.99	0.002
N=5000	50 %	0.65	0.03

5.1.4. NI-LPA performance with the network complexity variation

In this section, we vary μ, which indicates the fraction of links that are between communities in the graph. This parameter affects the network structure. The network is much more complex as μ rises, and the boundaries between groups become unclear. According to the result illustrated in Fig. 4, we can observe that the increase of the ratio μ affects the quality of detected partition for all the algorithms. When μ varies from 0.1 to 0.5, the GCE algorithm outperforms all other methods. The CPM algorithm gives a poor performance in complicated networks. The COPRA algorithm is efficient for simple graphs, but tends to a zero solution when $\mu = 0.4$. The NI-LPA method keeps good NMI values and a uniform behavior. Then, we can say in this case that the NI-LPA algorithm is less efficient than GCE, and it is more efficient than COPRA and CPM to find partitions in a complex network.

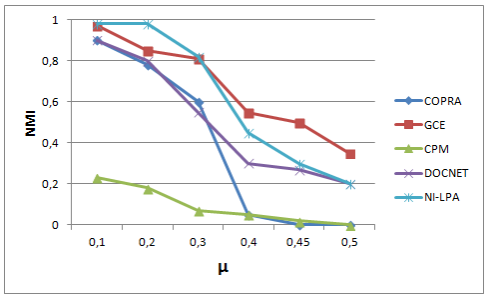


Fig. 4: Variation of NMI in function of μ, N=1000

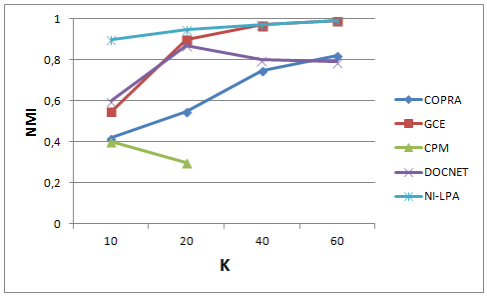


Fig. 5: Variation of NMI in function of K

5.1.5. NI-LPA performance with the network density variation

We evaluate the influence of the graph density (K from 10 to 60) on a network with 8000 nodes and $\mu = 0.2$. The NMI values are illustrated in Fig. 5. CPM has difficulties in detecting the exact partitions. Both GCE and COPRA maintain good results even when the network density increases. Even if DOCNET gives a satisfactory solution, the NI-LPA algorithm outperforms the other considered algorithms.

5.1.6. NI-LPA performance with different network sizes

NI-LPA performance with thousand nodes graphs

We generate a graph with 1000 nodes. μ is set to 0.2. The results are illustrated in Table 7 and Fig. 6. We notice that COPRA keeps good results for simple graphs, but it tends to zero solution from $O_n = 40\%$. CPM gives the furthest NMI values. GCE can find a partition close to the correct one tracking by NI-LPA, and then the DOCNET algorithm. Indeed, NI-LPA is very competitive, even though it did not outperform them in some cases.

The experimental results of Table 8 and Fig. 7, while we vary O_m from 2 to 8, show that NMI decreases according to the variation of O_m. We consider that GCE presents NMI's best values. NI-LPA shows good performance and gives results that are almost constant and better than other methods.

NI-LPA performance with two thousand nodes graphs

We consider a graph with 2000 nodes generated by LFR benchmarks. The community's size ranged from 20 to 100 and a mixing parameter is equal to 0.2. The analysis in Table 9 and Fig. 8, varying O_n, indicate that NI-LPA allows

Table. 7
NMI for networks with N=1000

O_n	COPRA	GCE	CPM	DOCNET	NI-LPA
10%	0.61	0.90	0.30	0.79	**0.98**
20%	0.52	0.87	0.29	0.62	**0.95**
40%	0.00	0.59	0.07	0.41	**0.62**
60%	0.00	**0.35**	0.06	0.24	0.30
80%	0.00	**0.20**	0.02	0.15	0.10

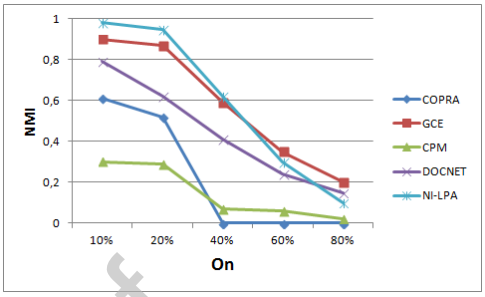


Fig. 6: Variation of NMI in function of O_n - 1000 N

Table. 8
NMI for networks with N=1000

O_m	COPRA	GCE	CPM	DOCNET	NI-LPA
2	0.61	0.90	0.33	0.75	**0.98**
3	0.56	**0.82**	0.07	0.62	0.77
5	0.49	**0.73**	0.30	0.50	0.69
8	0.00	**0.57**	0.32	0.42	0.47

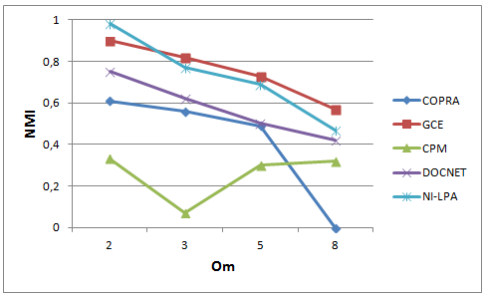


Fig. 7: Variation of NMI in function of O_m - 1000 N

the maximum value of NMI when $O_n = 10\%$ and $O_n = 20\%$. Then, the quality of the results decreases when we increase the number of overlapping nodes, but it remains acceptable. With high values of the O_n parameter (80%), GCE becomes the best. From the NMI values illustrated in Table 10 and Fig. 9, varying O_m, We note that all algorithms achieve good partitions. NI-LPA always admits the maximum values of NMI.

NI-LPA performance with eight thousands nodes graphs

Given a graph with 8000 nodes, from the results of Fig. 10 and Table 11 and while we vary O_n, we remark that NI-LPA is a highly recommended technique for community identification in a network.

By increasing the number of communities via 2 to 8, the analysis of Table 12 and Fig. 11 show that the NMI of all

Table. 9
NMI for networks with N=2000

O_n	COPRA	GCE	CPM	DOCNET	NI-LPA
10%	0.85	0.89	0.49	0.78	**0.95**
20%	0.72	0.82	0.37	0.72	**0.86**
60%	0.00	**0.58**	0.09	0.32	0.56
80%	0.00	**0.26**	0.04	0.08	0.21

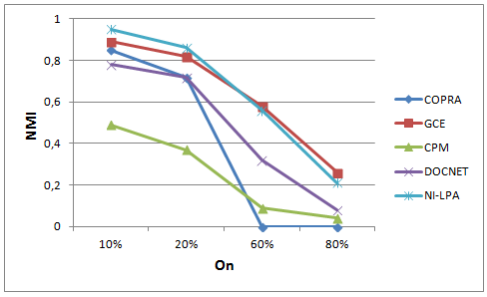


Fig. 8: Variation of NMI in function of O_n - 2000 N

Table. 10
NMI for networks with N=2000

O_m	COPRA	GCE	CPM	DOCNET	NI-LPA
2	0.85	0.89	0.49	0.78	**0.95**
3	0.60	0.77	0.42	0.57	**0.86**
5	0.52	0.72	0.41	0.47	**0.56**
8	0.30	**0.55**	0.38	0.43	0.21

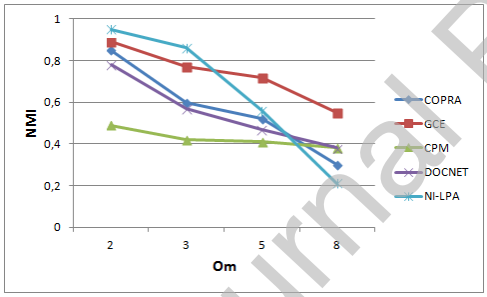


Fig. 9: Variation of NMI in function of O_m - 2000 N

Table. 11
NMI for networks with N=8000

O_n	COPRA	GCE	CPM	DOCNET	NI-LPA
10%	0.75	0.90	0.67	0.83	**0.96**
20%	0.63	0.85	0.55	0.70	**0.92**
40 %	0.42	0.42	0.38	0.49	**0.78**
60%	0.28	0.26	0.26	0.36	**0.59**
80%	0.02	0.00	0.01	0.20	**0.33**

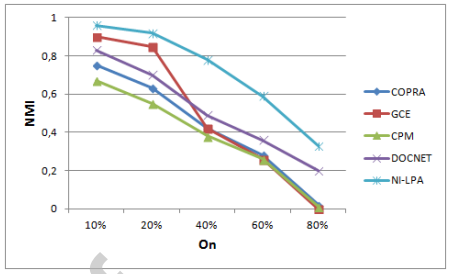


Fig. 10: Variation of NMI in function of O_n - 8000 N

Table. 12
NMI for networks with N=8000

O_m	COPRA	GCE	CPM	DOCNET	NI-LPA
2	0.74	0.90	0.68	0.82	**0.96**
3	0.60	0.81	0.60	0.58	**0.86**
5	0.52	0.68	0.52	0.48	**0.76**
8	0.35	0.53	0.40	0.35	**0.54**

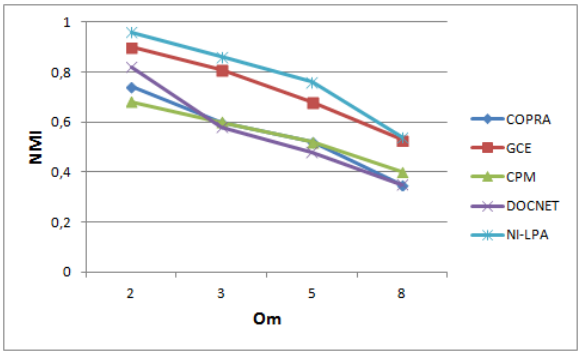


Fig. 11: Variation of NMI in function of O_m - 8000 N

the considered models are disturbed. Our method remains constant and reliable for community detection with a minimum value of NMI equal to 0.54 for $O_m = 8$. Therefore, our model is able to detect partitions that are nearly identical to the exact partitions..

NI-LPA performance with ten thousands nodes graphs
From the measurements given in Table 13 and Fig. 12, we notice that all the algorithms decrease in terms of NMI measurements, but NI-LPA always achieves the best results. However, it tends to a null partition, just like COPRA (the other methods do not converge anymore), when the number of overlapping nodes reaches 80% of the total number of nodes. While we vary O_m, according to Fig. 13 and Table 14, all the algorithms allow stable NMI values except COPRA. At the beginning (values O_m of 2 and 3) COPRA gives higher values, but presents a poor performance when O_m values equal to 5 and 8. NI-LPA always provides a partition close to the exact partition and remains reliable for O_m variations from 2 to 8.

NI-LPA performance with large graphs
To evaluate the effectiveness of NI-LPA to uncover overlapping communities in large networks, we generate in this section five large networks with different sizes N = 10,000; N=50,000; N=80,000; N=100,000 and N =150,000.
For each generated network, the number of overlapping nodes represents 10% of all nodes. Varying the network size from 10,000 to 150,000 nodes and show the results summarized in Fig. 14, we can see that the NMI values obtained by NI-LPA are between 0.95 and 0.97. To conclude, the NI-LPA algo-

Table. 13
NMI for networks with N=10000

O_n	COPRA	GCE	CPM	DOCNET	NI-LPA
10%	0.75	0.43	0.19	0.28	**0.95**
20%	0.71	0.19	0.12	0.21	**0.90**
40 %	0.00	0.32	0.08	0.17	**0.58**
80%	0.00	-	-	-	**0.47**

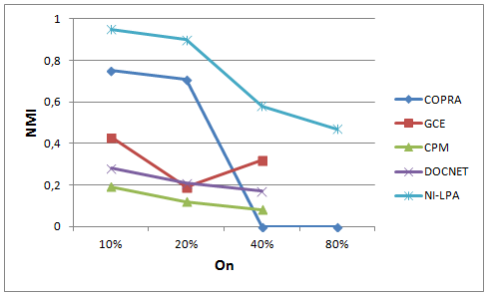


Fig. 12: Variation of NMI in function of O_n - 10000 N

Table. 14
NMI for networks with N=10000

O_m	COPRA	GCE	CPM	DOCNET	NI-LPA
2	0.74	0.42	0.18	0.26	**0.95**
3	0.69	0.34	0.18	0.21	**0.81**
5	0.00	0.37	0.14	0.22	**0.70**
8	0.00	0.45	0.13	0.22	**0.52**

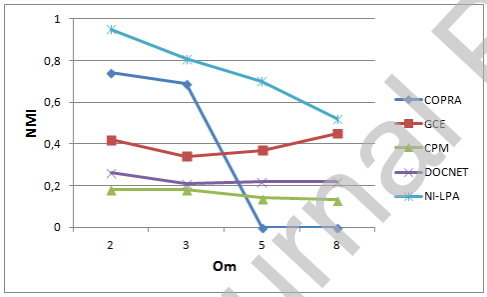


Fig. 13: Variation of NMI in function of O_m - 10000 N

rithm is able to detect a partition very similar to the exact partition, even for graphs that exceed 100 thousand nodes.

5.2. Real-world networks

We test our algorithm on real networks listed in Table 3 known as several models in the literature. We choose real networks with both small and large sizes. Not all these models have a ground truth partition structure for comparisons. There are measures used in literature to demonstrate the performance of algorithms in real networks. In our experimentations, We use the modularity metric and the exact number of discovered communities (rated M) (Rhouma and Romdhane, 2014) to evaluate our model. According to (Fortunato and Barthélemy, 2007), There is a relationship between the number of communities found in one partition and the modularity measure, but an increase in the number of identified groups does not necessarily leads to an increase in modularity values. In contrast, Newman (2006) suggests that modularity values upper than 0.4 indicate that the network is modular. In Table 15, we present the number of groups detected by each algorithm, and the modularity values. For the Karate club network, COPRA detects six communities with modularity values equal to 0.04, which is a weak result compared to other examined algorithms. However, in large networks like PGP, the number of detected groups by all the algorithms is high, so high modularity values can give better results. Besides, we observe that the modularity measurements obtained by NI-LPA on real networks are acceptable and exceed 0.4 in some cases (PGP and Ca-CondMat). We remark that NI-LPA becomes more efficient to find network structure when the network size increases. Indeed, to conclude, the NI-LPA can detect logical partitions in real networks.

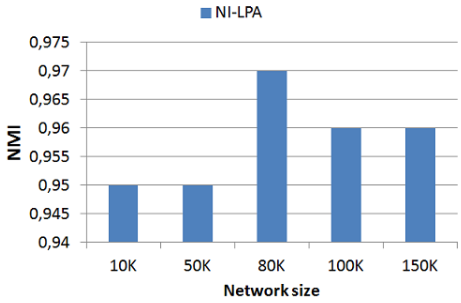


Fig. 14: NMI for large networks

5.3. Comparaison with label propagation based algorithms

We compare the performance of NI-LPA with other algorithms, which are an improvement of LPA to detect overlapping communities: COPRA, SLPA, and LinkLPA.
In the first task, we generate several graphs while varying μ and fix the rest of the parameters: $N = 5000$, $K = 10$, $O_n = 10\%$, and $O_m = 2$. The higher the mixture parameter of a network is, the more difficult it is to discover the community structure. The NMI values are shown in Fig. 15. We remark that when $\mu = 0.1$, all the algorithms can detect nearly the exact partition of the graph. Then, the achieved NMI values gradually decrease when μ increases, and in the case when the network is more complicated ($\mu = 0.5$), NI-LPA obtains the best NMI values followed by LinkLPA, SLPA, and COPRA.

In the second task, to compare the effectiveness of the algorithms on networks with many overlapping nodes, we generate a graph while increasing the fraction of overlapping nodes denoted O_n from 10% to 50%. In Fig. 16, we can see that the considered algorithms keep uniform behaviors and NI-LPA is better than other algorithms regardless of the level of overlap.

Table. 15
Results for real-world networks

Network	Metric	COPRA ($v = 3$)	GCE	CPM ($K = 4$)	DOCNET	NI-LPA
Karate club	Q_{ov}	0.04	0.26	0.32	0.24	0.30
	M	6.00	2.00	3.00	3.00	3.00
books	Q_{ov}	0.45	0.40	0.39	0.45	0.40
	M	2.00	5.00	4.00	3.00	4.00
PGP	Q_{ov}	0.39	0.23	-	0.23	0.5
	M	993.00	1200.00	-	1151.00	1020.00
Ca-CondMat	Q_{ov}	0.25	0.28	-	0.20	0.45
	M	1717.00	2257.00	-	1984.00	2110.00

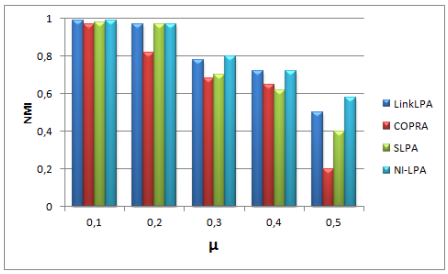


Fig. 15: NMI values for networks with N=5000

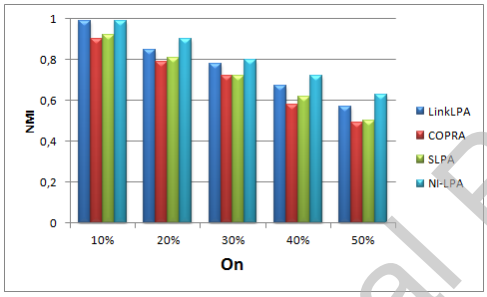


Fig. 16: Variation of NMI in function of O_n - N=5000

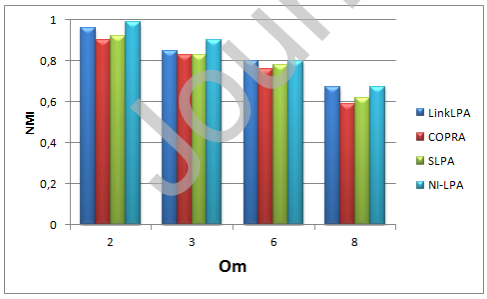


Fig. 17: Variation of NMI in function of O_m - N=5000

In the third task, we increase the value of O_m from 2 to 8. Generally, the accuracy goes down with the increase in the degree of overlapping nodes for all algorithms. In Fig. 17, we can see that for all the instances, all the extensions of LPA can detect an acceptable partition for the complex network. The results show that NI-LPA successfully outperforms the other algorithms.

6. Conclusion

In this paper, we present a novel method, Node Importance-Label Propagation Algorithm (NI-LPA), based on label propagation techniques to detect overlapping communities in complex networks. NI-LPA keeps the good efficiency advantages of LPA and exploits nodes' characteristics to improve the label propagation process. Thus, our algorithm uses a new filtering method to remove unnecessary labels. The proposed method improves the accuracy and robustness of community detection. An experimental study, carried out on a variety of artificial and real benchmark graphs, is proposed to situate NI-LPA among the most popular approaches. The quality of the community detection of NI-LPA was attested in this investigation.

We find that our algorithm identifies the nearest partition to the correct one where artificial graphs are concerned. It keeps high stability, particularly for complex networks. On real graphs, our method can uncover logical partitions. In the future, it will be interesting to adapt the NI-LPA model for directed and weighted networks. We can even improve it to discover communities in the context of a dynamic network.

References

Albert-László, B. and Réka, A. (1999). Emergence of scaling in random networks. *Science*, 286:509–512.

Arenas, A. (2013). Network data. http://deim.urv.cat/~alexandre.arenas/data/welcome.htm/.

Berahmand, K. and Bouyer, A. (2018). Lp-lpa: A link influence-based label propagation algorithm for discovering community structures in networks. *International Journal of Modern Physics B*, 32(06):1850062.

Berry, J. W., Hendrickson, B., LaViolette, R. A., and Phillips, C. A. (2011). Tolerating the community detection resolution limit with edge weighting. *Phys. Rev. E*, 83:056119.

Bezdek, J. C., Ehrlich, R., and Full, W. (1984). Fcm: The fuzzy c-means clustering algorithm. *Computers & Geosciences*, 10(2):191 – 203.

Blondel, V. D., Guillaume, J.-L., Lambiotte, R., and Lefebvre, E. (2008). Fast unfolding of communities in large

networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008.

Chen, N., Liu, Y., Chen, H., and Cheng, J. (2017). Detecting communities in social networks using label propagation with information entropy. *Physica A: Statistical Mechanics and its Applications*, 471:788 – 798.

Chin, J. H. and Ratnavelu, K. (2016). Detecting community structure by using a constrained label propagation algorithm. *PloS one*, 11(5):e0155320.

Clauset, A., Newman, M. E. J., and Moore, C. (2004). Finding community structure in very large networks. *Phys. Rev. E*, 70:066111.

Cordasco, G. and Gargano, L. (2010). Community detection via semi-synchronous label propagation algorithms. In *Business Applications of Social Network Analysis (BASNA), 2010 IEEE International Workshop on*, pages 1–8. IEEE.

Costantini, G. and Perugini, M. (2014). Generalization of clustering coefficients to signed correlation networks. *PLOS ONE*, 9:1–10.

Demsar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1–30.

Deng, Z.-H., Qiao, H.-H., Song, Q., and Gao, L. (2019). A complex network community detection algorithm based on label propagation and fuzzy c-means. *Physica A: Statistical Mechanics and its Applications*, 519:217 – 226.

Duch, J. and Arenas, A. (2005). Community detection in complex networks using extremal optimization. *Phys. Rev. E*, 72:027104.

Fortunato, S. (2010). Community detection in graphs. *Physics Reports*, 486(3):75 – 174.

Fortunato, S. and Barthélemy, M. (2007). Resolution limit in community detection. *Proceedings of the National Academy of Sciences*, 104(1):36–41.

Gregory, S. (2010a). Copra. https://http://gregory.org/research/networks/.

Gregory, S. (2010b). Finding overlapping communities in networks by label propagation. *New Journal of Physics*, 12(10):103018.

Hansen, D. L., Shneiderman, B., and Smith, M. A. (2011). Chapter 3 - social network analysis: Measuring, mapping, and modeling collections of connections. In Hansen, D. L., Shneiderman, B., and Smith, M. A., editors, *Analyzing Social Media Networks with NodeXL*, pages 31 – 50. Morgan Kaufmann, Boston.

He-Li, S., Jian-Bin, H., Yong-Qiang, T., Qin-Bao, S., and Huai-Liang, L. (2015). Detecting overlapping communities in networks via dominant label propagation. *Chinese Physics B*, 24(1):018703.

Jokar, E. and Mosleh, M. (2019). Community detection in social networks based on improved label propagation algorithm and balanced link density. *Physics Letters A*, 383(8):718 – 727.

Kelley, S., Goldberg, M., Magdon-Ismail, M., Mertsalov, K., and Wallace, A. (2012). *Defining and Discovering Communities in Social Networks*, pages 139–168. Springer US, Boston, MA.

Lancichinetti, A. (2013). Benchmark graphs to test community detection algorithms. https://sites.google.com/site/santofortunato/inthepress2.

Lancichinetti, A., Fortunato, S., and Kertész, J. (2009). Detecting the overlapping and hierarchical community structure in complex networks. *New Journal of Physics*, 11(3):033015.

Lancichinetti, A., Radicchi, F., Ramasco, J. J., and Fortunato, S. (2011). Finding statistically significant communities in networks. *PloS one*, 6(4):e18961.

Lee, C., Fergal, R., Aaron, M., and Neil, H. (2013). Greedy clique expansion. https://sites.google.com/site/greedycliqueexpansion/.

Liu, S.-C., Zhu, F.-X., and Gan, L. (2016). A label-propagation-probability-based algorithm for overlapping community detection. *Chinese Journal of Computers*.

Lu, M., Zhang, Z., Qu, Z., and Kang, Y. (2019). Lpanni: Overlapping community detection using label propagation in large-scale complex networks. *IEEE Transactions on Knowledge and Data Engineering*, 31(9):1736–1749.

McDaid, A., Greene, D., and Hurley, N. (2011). Normalized mutual information to evaluate overlapping community finding algorithms. *CoRR*.

Meo, P. D., Ferrara, E., Fiumara, G., and Provetti, A. (2014). Mixing local and global information for community detection in large networks. *Journal of Computer and System Sciences*, 80(1):72 – 87.

Newman, M. (2013). Network data. http://www-personal.umich.edu/~mejn/netdata/.

Newman, M. E. J. (2004). Fast algorithm for detecting community structure in networks. *Phys. Rev. E*, 69:066133.

Newman, M. E. J. (2006). Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, 103(23):8577–8582.

Newman, M. E. J. and Girvan, M. (2004). Finding and evaluating community structure in networks. *Phys. Rev. E*, 69:026113.

Opsahl, T. and Panzarasa, P. (2009). Clustering in weighted networks. *Social Networks*, 31(2):155 – 163.

Orman, G. K. and Labatut, V. (2009). A comparison of community detection algorithms on artificial networks. In Gama, J., Costa, V. S., Jorge, A. M., and Brazdil, P. B., editors, *Discovery Science*, pages 242–256, Berlin, Heidelberg. Springer Berlin Heidelberg.

Ouyang, B., Xia, Y., Wang, C., Ye, Q., Yan, Z., and Tang, Q. (2018). Quantifying importance of edges in networks. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 65(9):1244–1248.

Palla, G. (2011). Clusters et communities overlapping dense groups in networks. http://www.cfinder.org/.

Palla, G., Derényi, I., Farkas, I., and Vicsek, T. (2005). Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435(7043):814.

Pons, P. and Latapy, M. (2005). Computing communities in large networks using random walks. In Yolum, p., Güngör, T., Gürgen, F., and Özturan, C., editors, *Computer and Information Sciences - ISCIS 2005*, pages 284–293, Berlin, Heidelberg. Springer Berlin Heidelberg.

Raghavan, U. N., Albert, R., and Kumara, S. (2007). Near linear time algorithm to detect community structures in large-scale networks. *Physical review E*, 76(3):036106.

Ravasz, E. and Barabási, A.-L. (2003). Hierarchical organization in complex networks. *Phys. Rev. E*, 67:026112.

Reichardt, J. and Bornholdt, S. (2006). Statistical mechanics of community detection. *Phys. Rev. E*, 74:016110.

Rhouma, D. and Romdhane, L. B. (2014). An efficient algorithm for community mining with overlap in social networks. *Expert Systems with applications*, 41(9):4309–4321.

Shen, H., Cheng, X., Cai, K., and Hu, M.-B. (2009). Detect overlapping and hierarchical community structure in networks. *Physica A: Statistical Mechanics and its Applications*, 388(8):1706–1712.

Souam, F., Aïtelhadj, A., and Baba-Ali, R. (2014). Dual modularity optimization for detecting overlapping communities in bipartite networks. *Knowledge and information systems*, 40(2):455–48.

Steinhaeuser, K. and Chawla, N. V. (2010). Identifying and evaluating community structure in complex networks. *Pattern Recognition Letters*, 31(5):413 – 421.

Sun, H., Liu, J., Huang, J., Wang, G., Jia, X., and Song, Q. (2017). Linklpa: A link-based label propagation algorithm for overlapping community detection in networks. *Computational Intelligence*, 33(2):308–331.

Tong, C., Niu, J., Wen, J., Xie, Z., and Peng, F. (2015). Weighted label propagation algorithm for overlapping community detection. In *2015 IEEE International Conference on Communications (ICC)*, pages 1238–1243.

Watts, D. J. and Strogatz, S. H. (1998). Collective dynamics of 'small-world' networks. *Nature*, 393:440–442.

Wu, Z.-H., Lin, Y.-F., Gregory, S., Wan, H.-Y., and Tian, S.-F. (2012). Balanced multi-label propagation for overlapping community detection in social networks. *Journal of Computer Science and Technology*, 27(3):468–479.

Xie, J., Kelley, S., and Szymanski, B. K. (2013). Overlapping community detection in networks: The state-of-the-art and comparative study. *ACM Comput. Surv.*, 45(4):43:1–43:35.

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

Xie, J., Szymanski, B. K., and Liu, X. (2011). Slpa: Uncovering overlapping communities in social networks via a speaker-listener interaction dynamic process. In *Data Mining Workshops (ICDMW), 2011 IEEE 11th International Conference on*, pages 344–349. IEEE.

Xing, Y., Meng, F., Zhou, Y., Zhu, M., Shi, M., and Sun, G. (2014). A node influence based label propagation algorithm for community detection in networks. *The Scientific World Journal*.

Zhang, W., Zhang, R., Shang, R., and Jiao, L. (2018). Weighted compactness function based label propagation algorithm for community detection. *Physica A: Statistical Mechanics and its Applications*, 492:767 – 780.

Zhang, X.-K., Ren, J., Song, C., Jia, J., and Zhang, Q. (2017). Label propagation algorithm for community detection based on node importance and label influence. *Physics Letters A*, 381(33):2691–2698.

Zhang, X.-K., Tian, X., Li, Y.-N., and Song, C. (2014). Label propagation algorithm based on edge clustering coefficient for community detection in complex networks. *International Journal of Modern Physics B*, 28(30):1450216.

Authorship contributions

Category 1

Conception and design of study: Imen BEN ELKOUNI;
acquisition of data: Imen BEN ELKOUNI;
analysis and/or interpretation of data: Imen BEN ELKOUNI.

Category 2

Drafting the manuscript: Imen BEN ELKOUNI, Wafa KAROUI, Lotfi BEN ROMDHANE;
revising the manuscript critically for important intellectual content: Imen BEN ELKOUNI, Wafa KAROUI, Lotfi BEN ROMDHANE .

Category 3

Approval of the version of the manuscript to be published (the names of all authors must be listed): Imen BEN ELKOUNI, Wafa KAROUI, Lotfi BEN ROMDHANE.

Declaration of Conflict of Interest

All authors have participated in (a) conception and design, or analysis and interpretation of the data; (b) drafting the article or revising it critically for important intellectual content; and (c) approval of the final version.

This manuscript has not been submitted to, nor is under review at, another journal or other publishing venue.

The authors have no affiliation with any organization with a direct or indirect financial interest in the subject matter discussed in the manuscript