Community detection in dynamic graphs with missing edges

Oualid Benyahia, Christine Largeron, Baptiste Jeudy

Community Detection in Dynamic Graphs with Missing Edges

Oualid Benyahia, Christine Largeron and Baptiste Jeudy
Univ Lyon, UJM-Saint-Etienne, CNRS, Institut d Optique Graduate School,
Laboratoire Hubert Curien UMR 5516, F-42023, SAINT-ETIENNE, France

Abstract

Social networks are usually analyzed and mined without taking into account the presence of missing values. In this article, we consider dynamic networks represented by sequences of graphs that change over time and we study the robustness and the accuracy of the community detection algorithms in presence of missing edges. We assume that the network evolution can provide a complementary information allowing to neutralize the missing data. To confirm our hypothesis, we designed an experimental framework to simulate the missing data and compare the communities identified by the methods, with or without missing links. We explore two types of methods. The first ones, based on tensor decomposition, are adapted for dynamic networks. The second ones correspond to conventional community detection algorithms able to handle simple graphs. In our framework, the latter ones are adapted to dynamic graphs, either by merging the data during the preprocessing step or by merging the partitions during a post-processing step. The experimentation was conducted on synthetic and real dynamic networks for which the ground truth is available. The results confirm the best performances of the methods suited for dynamic networks when they present a complex community structure.

Keywords — Dynamic Graph, Social Network, Missing Data, Community Detection.

1 Introduction

In the recent few years, social data has attracted great interest in the data mining community and a variety of methods and software solutions have been proposed to ease their analysis. However, many of these tools have been designed on the assumptions that the network is totally known whereas, in practice, real networks are dynamic and they cannot be fully observed. Missing data can be due to various reasons. First, the large size of the network leads to consider only a sample. For instance, in the case of Twitter, a distributed crawler took four months to complete the data collection [14]. Second, even in

a subgraph, the relationship between two entities may be unobservable during the data acquisition, even if it exists. If this issue is well known in data mining and machine learning, it has been less studied in social data.

Previous works have mainly investigated and confirmed the impact of missing data on network properties such as the diameter, the centrality or the degree distribution [28, 20, 8]. However, this problem has not been well studied for other network analysis tasks.

In this article, we consider the community detection task which aims at grouping the nodes into sets with dense connections internally and sparser connections between groups [13]. We investigate, through extensive experiments done on dynamic networks, the ability of classical algorithms to detect the underlying community structure in presence of missing links.

The next section is dedicated to related work. The experimental framework is detailed in Sect. 3 and 4, Sect. 5 presents the experiments. Finally, Sect. 6 states our conclusion.

2 Related Work

Dealing with the missing data is a common problem in data mining [33, 32]. These missing values can arise from different sources such as measurement errors, non-responses, privacy settings, *etc.* Several techniques can be applied to deal with missing values in collected data [33, 39]. The easiest solution consists in ignoring these values or the records containing at least one missing value. However, this can lead to a loss of information and serious bias. Other approaches aim at replacing the missing data using various techniques [33, 37, 21] based notably on optimization methods [30, 40]. In the specific context of social mining, the problems of link prediction [31] and network reconstruction [48, 6, 24] are closely related to the study of missing data. Another related task is the matrix completion or recovery problem [23] where the objective is to fill out the missing entries of a matrix based on the observed ones. However, for the other tasks, the problem of missing data has been less studied even though the networks are most often partially observable and consequently represented by incomplete graphs. Thus, a main challenge is to estimate the impact of this missing data on the outcome of the mining process and the accuracy of the results.

Indeed, recent studies showed a negative effect of missing data on the structural properties of social networks [28, 38, 20]. In [28], the author highlights the problem of missing data in social network analysis. Through an exploratory analysis, he studies the effect of missing data on usual metrics (average degree, clustering coefficient, assortativity of nodes, number and size of components and average path length). The sensitivity of the metrics was assessed by simulating different forms of missing data in social networks describing scientific collaborations. In [41], the authors extend the work of [28] to understand the effects of different types of missing data in the case of multilayer networks. In [12] as well as in [8], the authors focus on the robustness of the centrality measures under

the constraints of missing data. These works confirm that the analysis of the network is severely biased by the presence of missing links.

Finally, the treatment of missing values and the solutions to replace them remains seldom studied. In [20], the author explores a variety of data imputation procedures, similar to those introduced in [38], like unconditional mean substitution, preferential attachment substitution (preserving the degree distribution) or the hotdocking substitution technique. He also studied the impact of these solutions on network's properties such as the clustering coefficient, the assortativity, the reciprocity and the inverse geodesic distance. Several imputation procedures have been designed in an application context: in [3], a model-based imputation is proposed for predicting information in an incomplete road network whereas in [18], the authors introduced a latent space imputation model to predict links in a political network.

Furthermore, Kim and Leskovec [25] study the network completion problem which consists in inferring the unobserved part of the network whereas Yan and Gregory [49] explore two tasks (link prediction and the community detection) with different scenarios to simulate the missing data. For these tasks, they conclude that the performance of the methods is differently affected depending on the type of missing edges.

However, none of these works take into account the network dynamic. In this article, we assume that the network evolution can provide a complementary information allowing to neutralize the missing data. We study the effect of missing data in dynamic networks and in the specific context of community mining. If our hypothesis is true, the community detection algorithms able to handle dynamic networks should be less penalized by missing links than algorithms designed for static graphs. Through intensive experiments, our aim is to analyze the robustness and the accuracy of the community detection algorithms in presence of missing edges. For this purpose, an experimental framework has been designed allowing to simulate the missing data according to two scenarios. It is presented in the next section.

3 Methodology

In this section we introduce the proposed experimental framework.

3.1 Dynamic Network Representation.

A dynamic network $\mathcal{G}$ is a pair $(\mathcal{V}, \mathcal{E})$ where $\mathcal{V}$ is a set of N nodes (denoted v_i) and $\mathcal{E} \subset \mathcal{V}^2 \times \{1, \cdots, T\}$ is a set of undirected edges. Each edge is defined by two nodes of $\mathcal{V}$ and a timestamp in $\{1, \cdots, T\}$. We define the graph $\mathcal{G}_t = (\mathcal{V}, \mathcal{E}_t)$ describing the state of the network at a given timestamp (referred to as a snapshot) where $(v_i, v_j) \in \mathcal{E}_t$ iff $(v_i, v_j, t) \in \mathcal{E}$.

This leads to a natural formalization of this network as a tensor where the successive adjacency matrices $\mathbf{X}_t = \{x_{ijt}\}$ of $\mathcal{G}_t$ are combined in a three-way tensor $\mathcal{X} \in \mathbb{R}^{N \times N \times T}$. A tensor element x_{ijt} is equal to 1 if there exists an edge

between the nodes v_i and v_j in the graph $\mathcal{G}_t$ and 0 otherwise. We suppose that all the graphs (and thus tensors) are symmetric ($x_{ijt} = x_{jit}$).

We also define the aggregated graph $\mathcal{G}^{aggr} = (\mathcal{V}, \mathcal{E}^{aggr})$, where $\mathcal{E}^{aggr} = \cup_t \mathcal{E}_t$ (there is an edge (v_i, v_j) in $\mathcal{G}^{aggr}$ iff there is an edge (v_i, v_j, t) in any of the $\mathcal{G}_t$).

3.2 Missing Edges Representation.

In a real dynamic network, some edges of $\mathcal{G}$ may be missed (be unobserved) and eventually be considered incorrectly nonexistent. To simulate these missing/unobserved edges, we use a symmetric tensor $\mathcal{W} \in \mathbb{R}^{N \times N \times T}$ of the same size as $\mathcal{X}$. Its values w_{ijt} are equal to 0 if the edge between v_i and v_j at time t is missing/unobserved and 1 otherwise. Another way to interpret $\mathcal{W}$ is that $w_{ijt} = 0$ if the value of x_{ijt} is unknown.

We refer to the observed incomplete version of the initial dynamic network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ as $\mathcal{G}^m = (\mathcal{V}, \mathcal{E}^m)$ where $\mathcal{E}^m \subset \mathcal{E}$ denotes the subset of the remaining observed edges. The induced adjacency tensor $\mathcal{X}^m$ of $\mathcal{G}^m$ is therefore:

$$\mathcal{X}^m = \mathcal{W} * \mathcal{X}, \text{ i.e., } x^m_{ijt} = w_{ijt}.x_{ijt} \tag{1}$$

We assume that the missing edge tensor $\mathcal{W}$ may be known or unknown depending on the experimental scenario. The tensor $\mathcal{X}$ of the real network is always unknown (except, of course, if there is no missing edge).

3.3 Generating Missing Edges.

To generate incomplete data, we start from an artificial or real-world network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Then, given a desired ratio τ of missing edges, we choose $\tau.|\mathcal{E}|$ edges as missing using one of the two following scenarios.

Unbiased Missing Edges

In this scenario, we make the assumption that the missing edges are uniformly and randomly sampled from all observed edges in the dynamic graph. For instance, data may be observed with sensors (as in the classroom dataset) that may fail to detect an interaction between two nodes with a given probability. So, given τ, the $\tau \cdot |\mathcal{E}|$ missing edges are chosen, with an uniform probability, from the observed edges $\mathcal{E}$ of the original dynamic network.

With this scenario, missing data is not really a so difficult problem, because we can expect that the structure of the network remains stable in time. If it is the case, this means that if an edge is missing at the time t, there is a high probability for this edge to be present at another time.

Biased Missing Edges

In this second scenario, more difficult, we consider that if an edge between two nodes v_i and v_j is missing at time t, then it is also missing at other timestamps. In this case, all the information about the relation between v_i and v_j is lost. This

means that the missing value tensor $\mathcal{W}$ does not depend on time: $w_{ijt} = w_{ijt'}$ for all t and t'. This can model the fact that the network is only a partial view of the world. For instance, two people working in the same office may communicate verbally. Thus the relation between them is not present in a graph of email communications.

To achieve this, missing edges are sampled uniformly from the aggregated graph $\mathcal{G}^{aggr}$. For each sampled edge (i, j), we set $w_{ijt} = 0$ for all t. This sampling is performed until a proportion τ of edges is selected in $\mathcal{G}$, i.e., $\sum_{(ijt)}(1 - w_{ijt})x_{ijt} \geq \tau \cdot |\mathcal{E}|$ (in general, we cannot achieve equality because each sampled edge in $\mathcal{G}^{aggr}$ corresponds to several edges in $\mathcal{G}$).

4 Approaches for Community Mining in Dynamic Networks

Several approaches have been introduced to detect communities in dynamic networks among which, we can mention stochastic blockmodels [43], [26], [44] clique percolation method extension [35], quality function optimization notably adaptations of CNM algorithm [34], Louvain or Infomap methods [4, 2, 17]. In this article, we suppose that the community detection in a dynamic network aims to identify a unique partition of the vertices into non overlapping clusters. We explore two approaches: The first one, based on tensor factorization methods, is well adapted to dynamic networks since it processes all timestamps simultaneously. It provides a global partition of the vertices into communities, defined over the whole network. The second approach is to adapt conventional algorithms able to find the communities on each snapshot graph $\mathcal{G}^t$. It is based either on partition aggregation or on graph aggregation $\mathcal{G}^{aggr}$.

4.1 Methods Adapted for Dynamic Networks

The aim of tensor factorization methods [27] is to compute a low rank approximation of the dynamic graph tensor. This approximation captures the underlying latent structures describing the temporal and structural correlation between vertices.

Given the observed adjacency tensor $\mathcal{X}^m$ and a parameter R (the number of components), we compute the canonical decomposition (CANDECOMP) [11], also known as parallel factorization (PARAFAC) [16]. We refer to the canonical decomposition as CANDECOMP/PARAFAC (CP) model. The factorization of $\mathcal{X}^m$ leads to factor matrices $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ all with R columns.

$$x^m_{ijt} = \sum_{r=1}^{R} a_{ir} b_{jr} c_{tr}$$

In our case, since $\mathcal{X}^m$ is symmetrical, $\mathbf{A}$ and $\mathbf{B}$ are equal. The matrix $\mathbf{A}$ describes the latent structures in the original graph and the matrix $\mathbf{C}$ the activity

pattern of these components through time. Finding the right value for parameter R is a difficult problem [10]. In our case, since we know the ground truth and since we focus on the missing data problem, we use the real number of communities as R.

When the missing edge tensor $\mathcal{W}$ is known, several extensions of the original CP methods have been proposed: EM-ALS [9, 47], *CP-INDAFAC* [45] and *CP-WOPT* [1].

Extracting Community Partition from Tensor Factors

In order to match the factor components to distinct communities we use a clustering algorithm (*Kmeans*) on the rows of the factor matrix A to get R distinct clusters, describing membership of each node to his community. By this way, we obtain a global partition $\mathcal{P}^{CP}$ describing R non overlapping communities defined on the whole evolving network:

$$\mathcal{P}^{CP} = \{C_1, ..., C_R\} \tag{2}$$

Each node v_i of the network $\mathcal{G}$ belongs to exactly one community C_i.

4.2 Methods Adapted for Static Networks.

We also use well-known community detection algorithms *Louvain* and *Infomap*. The *Louvain* algorithm is a greedy agglomerative hierarchical clustering method which optimizes the modularity measure [7]. Infomap exploits data compression for community identification by employing random walks to analyze the information flow through a network [36].

These two algorithms are designed to identify community structures on static graphs and in general, they are not adapted to deal with dynamic networks. For this reason, we adapt them in two different ways:

Independent Community Mining

The static algorithms are executed independently on each observed graph snapshot $\mathcal{G}_t^m$ of the observed dynamic network $\mathcal{G}^m$. We thus obtain a sequence of independent partitions $\mathcal{P}_t^{static}$ describing the communities of each graph snapshot:

$$\mathcal{P}^{static} = \{\mathcal{P}_1^{static}, \cdots, \mathcal{P}_T^{static}\} \tag{3}$$

Community Mining on Aggregated Graph

Another approach consists in applying these static algorithms on the aggregated graph $\mathcal{G}^{aggr}$. This approach provides directly a global partition $\mathcal{P}_{aggr}^{static}$.

4.3 Evaluation

In the experiments, we study synthetic and real networks for which ground truth is available. This allows to compare the partitions built by the algorithms with those of the original networks. Depending on the datasets, the ground truth is available either as a sequence of partitions $\mathcal{P}^* = \{\mathcal{P}_1^*, \mathcal{P}_2^*, \cdots, \mathcal{P}_T^*\}$ (for the synthetic networks) or as a unique global partition $\forall t \in \{1, \cdots, T\}, \mathcal{P}_t^* = \mathcal{P}_1^* = \mathcal{P}^*$ (for the primary school dataset).

The assessment of the accuracy of the different algorithms requires the use of a distance or similarity measures between pairs of (sequence of) partitions. In our experiments, we use the *Normalized Mutual Information* [42], the *Jaccard Index* [22] and the *Adjusted Rand Index* [19].

However, as shown before, the ground truth and the outputs of the algorithms may be either a unique partition or a sequence of T partitions (one for each snapshot). To be able to compare these two kinds of results, we transform sequences of partition $\mathcal{P} = \{\mathcal{P}_1, \mathcal{P}_2, \cdots, \mathcal{P}_T\}$ into a *consensus partition*, denoted $Consensus(\mathcal{P})$, as explained below.

Consensus Partition

Given a sequence of partitions $\mathcal{P} = \{\mathcal{P}_1, \cdots, \mathcal{P}_T\}$, we construct a feature matrix $\mathbf{F}$ of size $N \times T$. The value $\mathbf{F}_{it}$ is equal to j if node v_i belong to community C_j in $\mathcal{P}_t$. The nodes are then assigned to a unique community by performing a clustering (with the Kmeans algorithm) on the rows of matrix $\mathbf{F}$. The result of the clustering is the consensus partition $Consensus(\mathcal{P})$.

With this approach, each time the ground truth or the result of an algorithm is a sequence of partitions, we apply the consensus to obtain a unique global partition for the period $\{1, ..., T\}$.

To summarize, the global unique ground truth partition for each dataset can thus be compared to the output of the algorithms:

- For the tensor algorithms (*CP-ALS*, *CP-WOPT* and *CP-INDAFAC*), the output is always a global unique partition.

For the static algorithms *Louvain* and *Infomap*, we use two approaches:

- **Validation on aggregated partitions :** in this case, the ground truth is compared to the consensus partition $Consensus(\mathcal{P}^{static})$ built on the sequence of partitions $\mathcal{P}^{static}$.
- **Validation on aggregated graph :** in this case, the static algorithms are applied on the aggregated graph. So, the ground truth is compared to the global partition $\mathcal{P}_{aggr}^{static}$.

5 Experiments

We experimentally evaluate the performance of the algorithms (*CP-ALS*, *CP-WOPT*, *CP-INDAFAC*, *Louvain* and *Infomap*) in presence of missing edges. We

use one real network and two sets of synthetic networks built with generators which also give the ground truth. Thus, we can compare this last one with the partitions provided by the algorithms by means of the NMI and ARI scores and Jaccard Index but, for sake of brevity, we only present the results according to the NMI since they are approximatively the same for the other scores. We consider the scenarios described previously to generate the missing edges with values for the ratio τ varying from 10% to 90%. Moreover, for the static methods (*Infomap* and *Louvain*) suited to handle simple graphs, we present the results obtained either by aggregating the partitions built on each snapshot (denoted Validation on aggregated partitions in the captions and reported in sub-figures (a)) or by aggregating the snapshot graphs (denoted Validation on aggregated graph on sub-figures (b)). The results of the tensor algorithms (CP-*) are reported on both sub-figures (a) and (b) (they are the same on both since the "aggregated graph" and "aggregated partitions" apply only to static methods).

For each dataset, we present the average and the standard deviation of the NMI as a function of the missing edge ratio. The averages and standard deviations are computed on ten simulations done with different missing values tensors $\mathcal{W}$ randomly generated for each scenario.

5.1 Primary School Network

Dataset

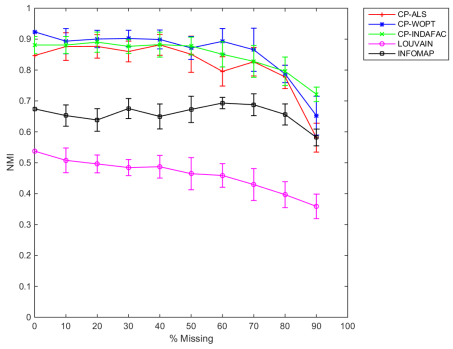
The first dataset [46] corresponds to a real temporal network describing the interactions between 231 children and 10 teachers equipped with a proximity sensor having an unique identifier (241 nodes). The sensor continuously monitored the close-range (less than 1.5 meters) face-to-face contacts of individuals and relayed the proximity relations to a receiving system that timestamps and logs the data.

The data, collected over two consecutive days (October 2009 from 8:30 am to 5:15 pm) with a temporal resolution of 20 seconds, has been aggregated to obtain a temporal network represented by $T = 18$ successive adjacency matrices. Each identifier is associated to the class of the participant, so that we have a global ground truth partition $\mathcal{P}^*$ composed of 11 communities for this dynamic network (10 communities for grouping children classes and an additional community for teachers).

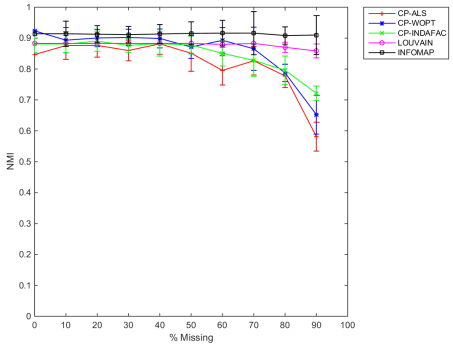
Results

Figure 1 and Fig. 2 show, respectively for the **unbiased scenario** and the **biased scenario**, the NMI scores in function of the missing edges rates obtained with the different algorithms.

The **unbiased scenario**, corresponding to missing edges selected randomly is depicted in Fig. 1. For missing edge ratios up to 70%, the tensor methods have good results but starts to decrease for higher missing edges ratios. With aggregated partitions (Fig. 1a), the static methods *Louvain* and *Infomap* obtain lower scores compared to the tensors methods, especially *Louvain* with an



(a) Validation on aggregated partitions



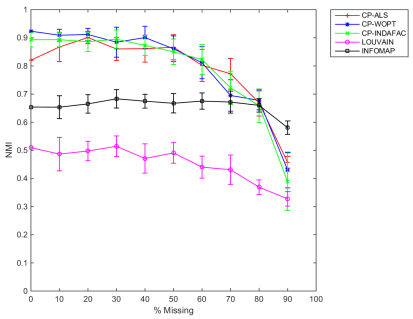
(b) Validation on aggregated graph

Figure 1: NMI results on the **primary school** graph with missing edges sampled according to the **Unbiased missing edges scenario**. In all figures, Validation on aggregated partitions and Validation on aggregated graph are the same for methods based on tensor decomposition (CP-*).

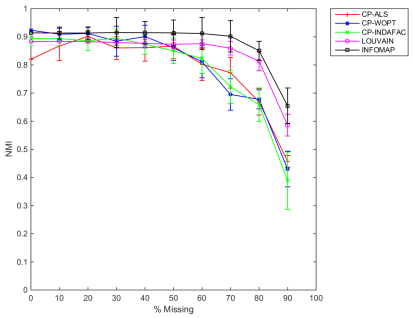
average score below 50% in most of the cases. As shown in Fig. 1b, when the static algorithms (*Louvain* and *Infomap*) are applied directly on the aggregated graph, they obtain results which remain stable even when 90% of the edges are missing, with a clear advantage for *Infomap* over the other methods.

Consequently, on this dataset, the static methods *Louvain* and *Infomap* are clear winner when applied on an aggregated graph. This is due to the fact that the aggregated graph remains very stable even with high missing edge ratios.

With the **biased scenario**, which is most difficult because a missing edge is selected over all the timestamps, we observe a degradation of all the results as shown Fig. 2. The NMI scores for the tensor methods start to decrease rapidly and severely when the ratio of missing edges is above 60%. For the static methods (*Louvain* and *Infomap*), we can remark the same effect on (Fig. 2b)



(a) Validation on aggregated partitions



(b) Validation on aggregated graph

Figure 2: NMI results on the **primary school** graph with missing edges sampled according to the **Biased missing edges scenario**.

when they are applied on the aggregated graph (but they remains better than tensor methods).

In conclusion, for this network presenting a relatively stable community structure, the more edges are considered as missing the lower is the efficiency of the tensor factorization algorithms. Moreover, these last ones are surpassed by conventional static algorithms applied on the aggregated graph.

5.2 Synthetic Dynamic Graphs

Dataset

We used the generator **DANCer** [29, 5] to construct synthetic datasets. A network is defined by a sequence of undirected attributed graphs having a well defined partition of the vertices into non-overlapping communities at each snapshot. Thus, the ground truth partition is given by $\mathcal{P}_t^*$ with $t \in \{1, \cdots, T\}$.

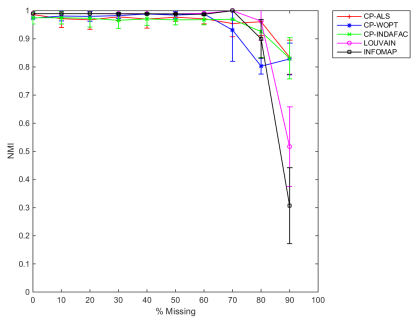
The evolution of the network is obtained by removing or adding edges, by migrating nodes from a community to another one, by splitting a community

into two new sub-communities or by merging two existing communities into a single community.

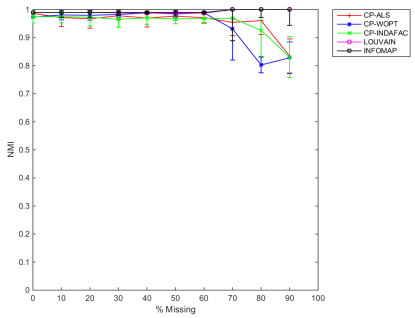
We generated 10 dynamic networks with the same set of parameters[1] but with different seeds. Each of them have 100 nodes, 2000 initial edges and an initial community structure composed of 4 groups. These networks evolved over $T = 10$ time steps.

The results are computed with 10 missing values tensors $\mathcal{W}$ randomly generated for each scenario.

Results



(a) Validation on aggregated partitions

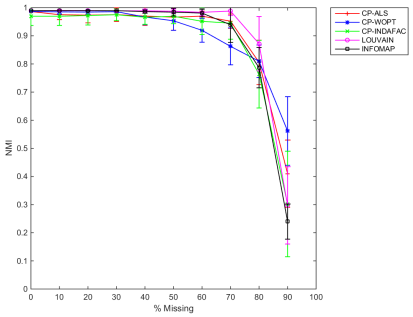


(b) Validation on aggregated graph

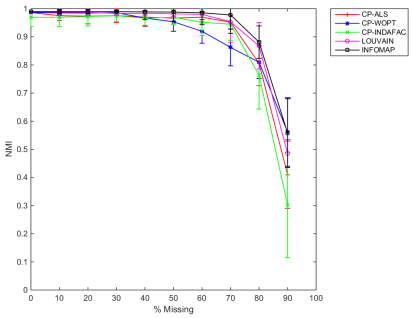
Figure 3: NMI results on the synthetic graph **DANCer Graph** with missing edges sampled according to the **Unbiased missing edges scenario**.

For the **unbiased scenario** of missing edges depicted in Fig. 3, the NMI scores of all the tensor decomposition methods remain stable with high accuracy,

[1] K=4; n=100; Nb. Rep=10; Edges Within=10; Edges Between=3; MTE=2000; Proba Micro=0.5; Add Btw. Edges=0.5; Remove Btw. Edges=0.5; Add Wth. Edges=0.3; Remove Wth. Edges=0.5; Timestamps=10; Proba Merge=0.2; Proba Split=0.2. All other parameters are set to 0.



(a) Validation on aggregated paritions



(b) Validation on aggregated graph

Figure 4: NMI results on the synthetic graph **DANCer Graph** with missing edges sampled according to the **Biased missing edges scenario**.

above 80%, and a low standard deviation, even with a large ratio of missing edges.

For the static algorithms (*Infomap* and *Louvain*), we obtain different results depending on the aggregation. When the partitions are determined at each time step and then aggregated into a consensus partition, as shown Fig. 3a the average NMI score remains good and stable until to a ratio τ equals to 70%. Then it decreases severely with a high standard deviation. This emphasizes the prevalence of the tensor decomposition methods, even with a great number of missing edges. Conversely, as shown Fig. 3b, when an aggregated graph is used to detect the communities, the static algorithms (*Infomap* and *Louvain*) give the best NMI scores, with a slight advantage over the tensor methods, and their result remain stable even with a high ratio of missing edges.

For the **biased scenario**, Fig. 4, the NMI score is almost the same for all the algorithms. It starts to decrease rapidly when the ratio of missing edges is approximately equals to 60%. Moreover we can not observe a significant difference for the static algorithms when the validation is done on aggregated partitions or on aggregated graph.

The conclusion for this dataset is thus similar to the previous one: the static methods on the aggregated graph perform better even if their advantage is less pronounced than in the school dataset.

5.3 Synthetic Networks from Benchmark Model

Dataset

We also used a recent benchmark model for generating dynamic networks having a periodic evolution of the communities [15]. This cyclic evolution of the network is modeled by two dynamic processes applied to the communities: growing/shrinking and merging/splitting. In these two dynamic processes, two equal sized communities start off as random graphs with internal link density p_{in}, and with link density p_{out} between them (at $t = 1$).

In **Merging/splitting** process, edges are added between communities until they become a random graph with only one community with a link density of p_{in} (at $t = 0.5T$). Then, this process reverses (edges are removed) until the communities are split again at $t = T$.

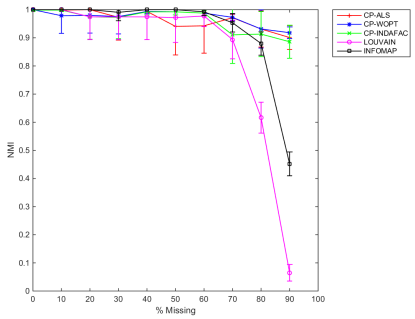
In **growing/shrinking** process, nodes are transferred from one community to the other, until a fraction f of the nodes of one community have been moved. At $t = 0.25T$, the size of the first community is maximal and and the size of the second is minimal. The process reverses, and at $t = 0.5T$ communities are of equal size again, and at $t = 0.75T$ the second community is at the maximal size. Finally, at $t = T$ the communities return to equal size. At all times, the internal link density of both communities is maintained at p_{in} and link density between the two communities is maintained at p_{out}.

We used the **StdMixed** configuration [15] which combines the two previous processes with the parameter p_{in} (Internal Communities link density) set to 0.5 and the other link densities, notably p_{out}, set to 0.1. At $t = 1$, there are four equal size communities with 60 nodes. The first two communities follow the merging/splitting process, and the other two communities follow the expand/contract process. The number of snapshots is $T = 10$.

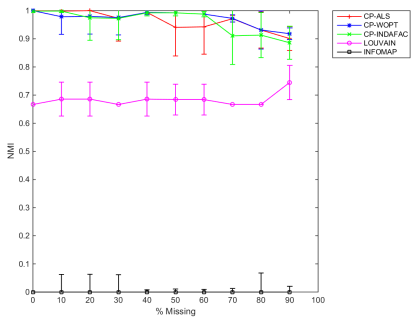
Results

For the **unbiased scenario**, where the missing edges are randomly selected, the partitions provided by the tensor decomposition methods have larger NMI, even with a large number of missing edges, as shown Fig. 5.

With the **validation on aggregated partitions**, illustrated by Fig. 5a, the static algorithms (*Infomap* and *Louvain*) also give good NMI scores when the percentage of missing edges is lower than 60%. Then the score decreases rapidly to reach 40% for *Infomap* and about 10% for *Louvain*. Figure 5b shows the NMI score when the static algorithms *Infomap* and *Louvain* are applied on an aggregated graph. The communities identified are less relevant compared to those provided by the tensors decomposition methods, especially for *Infomap* which fails completely to found the ground truth. We can note that the low score



(a) Validation on aggregated partitions



(b) Validation on aggregated graph

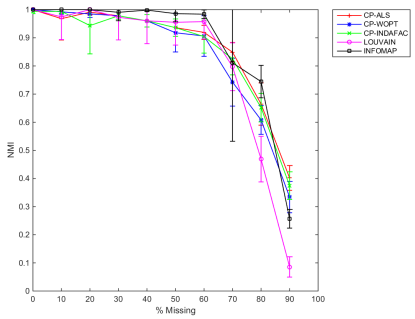
Figure 5: NMI results on the synthetic configuration **StdMixed** with missing edges sampled according to the **Unbiased missing edges scenario**.

obtained by Infomap (NMI = 0) for this very evolving community structure is consistent with other experimental results recently published [50].

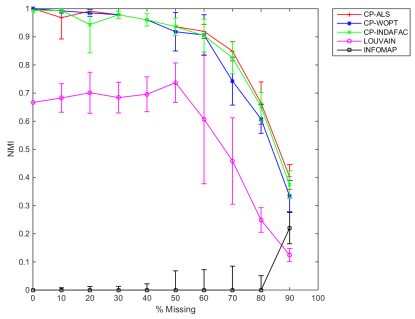
In fact, in this benchmark, the dynamic process governing the evolution of the communities membership has a negative impact on the results of the static methods.

For the **biased scenario**, the missing data impacts the accuracy of the different algorithms as shown in Fig. 6a. In this case, the static methods on the aggregated graph have the worse results (whereas they were better than the aggregated partitions in the previous graphs). This can be explained by the fact that since the structure of the communities changes a lot, the aggregated graph tends to become complete (i.e., it tends to contain all possible edges).

Thus, for this kind of dynamic network, where the structure of the communities changes a lot with a high discrepancy between different snapshots, the tensor decomposition methods perform better, even with a high number of missing edges.



(a) Validation on aggregated partitions



(b) Validation on aggregated graph

Figure 6: NMI results on the synthetic configuration **StdMixed** with missing edges sampled according to the **Biased missing edges scenario**.

5.4 Processing Time

To assess the processing time of the different static and dynamic community detection algorithms, we conducted an experimentation on different DANCer graphs with the same parameters except for the number of nodes (from 100 to 2000 nodes). The ratio of missing edges was between 10% and 90%. The results for 1500 nodes are presented in Fig. 7. The results for other numbers of node are similar.

In general, the static methods are faster (by a factor between 2 to 10) especially for larger number of nodes. Except for the highest level of missing edges (90%), computation times do not depend much on the missing edge ratio. For 90% of missing edges, all the methods except *Infomap* have a large increase in processing time. These are iterative methods that aim to optimize a function (which measures the quality of the communities). This increase in processing time means that these algorithms need more iterations to find a good partition. This is probably because the "gradient" of the optimized function (i.e., the increase of this function at each step of the algorithms) is small (the communities

are not well separated), and thus finding the optimal takes more iterations.

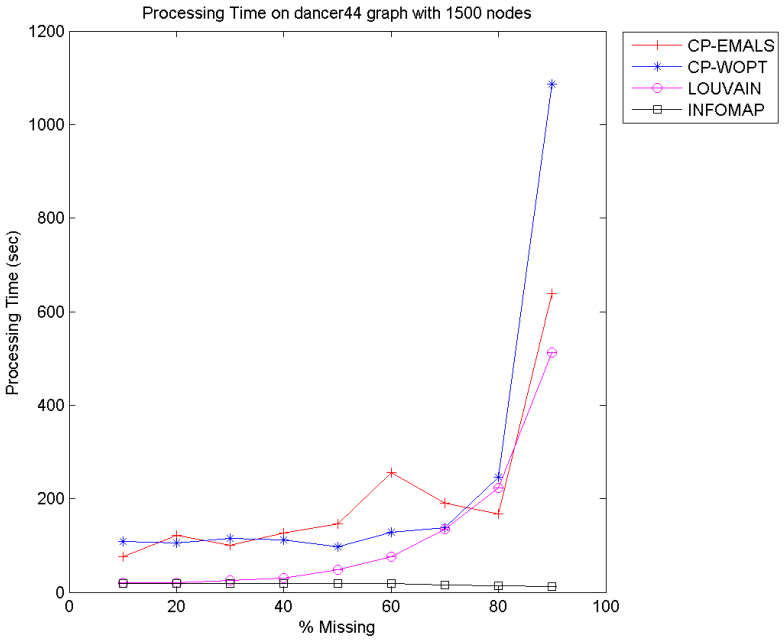


Figure 7: Processing Time for **DANCer Graph** with **1500** nodes

6 Conclusion

We investigated the impact of missing edges on the performance and the efficiency of several community detection algorithms in dynamic networks. The dynamic algorithms, based on tensor decomposition methods have been compared to two conventional algorithms dedicated the detection of communities in static networks: *Louvain* and *Infomap*.

The study confirms that the network evolution can provide a complementary information allowing to neutralize the missing data.

More precisely, in the case of a low discrepancy between the communities at different timestamps, the static algorithms perform well to find the ground truth partition, notably when they are applied on an aggregated version of the graphs composing the network and, especially if the number of missing edges is reasonably low. However, when the communities structure changes a lot, the tensor decomposition methods provide partitions that are more significant compared to those identified by the static algorithms.

It is worth noting than even with very high ratio of missing edges (80% or even 90%), the NMI score remains above 50% in many cases. This shows that

the dynamic networks contains lots of redundancy and that the algorithms are able to use it.

This work should be pursued further, notably by considering versions of Louvain or Infomap [4, 2, 17] recently introduced and dedicated to dynamic networks. However it is not certain that they achieve better results when the community structure is very evolving since they change the initialization in order to force stability by beginning the detection at time t with the partition obtained at time $t-1$.

Moreover, we studied two scenarios to generate incomplete graphs, unbiaised and biaised missing edges. We could consider more complex cases, like for instance when missing edges are concentrated around a node or when they edges are mainly removed at given timestamps. We have focused on non-directed and non-weighted networks, we could also study the case of directed and weighted graphs.

References

[1] Evrim Acar, Daniel M Dunlavy, Tamara G Kolda, and Morten Mørup. Scalable tensor factorizations for incomplete data. *Chemometrics and Intelligent Laboratory Systems*, 106(1):41–56, 2011.

[2] Riza Aktunc, Ismail Hakki Toroslu, Mert Ozer, and Hasan Davulcu. A dynamic modularity based community detection algorithm for large-scale networks: Dslm. In *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015*, ASONAM '15, pages 1177–1183, 2015.

[3] Muhammad Tayyab Asif, Nikola Mitrovic, Lalit Garg, Justin Dauwels, and Patrick Jaillet. Low-dimensional models for missing data imputation in road networks. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 3527–3531. IEEE, 2013.

[4] T. Aynaud and J.-L. Guillaume. Static community detection algorithms for evolving networks. In *Proceedings of the 8th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt)*, pages 513–519, 2010.

[5] Oualid Benyahia, Christine Largeron, Baptiste Jeudy, and Osmar R Zaïane. Dancer: Dynamic attributed network with community structure generator. In *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, volume 9853 of *Machine Learning and Knowledge Discovery in Databases*, pages 41 – 44, 2016.

[6] Kevin Bleakley, Gérard Biau, and Jean-Philippe Vert. Supervised reconstruction of biological networks with local models. *Bioinformatics*, 23(13):i57–i65, 2007.

[7] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008, 2008.

[8] Stephen P Borgatti, Kathleen M Carley, and David Krackhardt. On the robustness of centrality measures under conditions of imperfect data. *Social networks*, 28(2):124–136, 2006.

[9] Rasmus Bro. *Multi-way analysis in the food industry: models, algorithms, and applications*. PhD thesis, Københavns Universitet'Københavns Universitet', 1998.

[10] Rasmus Bro and Henk A. L. Kiers. A new efficient method for determining the number of components in parafac models. *Journal of Chemometrics*, 17(5):274–286, 2003.

[11] J Douglas Carroll and Jih-Jie Chang. Analysis of individual differences in multidimensional scaling via an n-way generalization of "eckart-young" decomposition. *Psychometrika*, 35(3):283–319, 1970.

[12] Elizabeth Costenbader and Thomas W Valente. The stability of centrality measures when networks are sampled. *Social networks*, 25(4):283–307, 2003.

[13] Santo Fortunato. Community detection in graphs. *Physics Reports*, 486(3):75–174, 2010.

[14] Maksym Gabielkov and Arnaud Legout. The Complete Picture of the Twitter Social Graph. In *ACM CoNEXT 2012 Student Workshop*, Nice, France, December 2012.

[15] Clara Granell, Richard K. Darst, Alex Arenas, Santo Fortunato, and Sergio Gómez. A benchmark model to assess community structure in evolving networks. *CoRR*, abs/1501.05808, 2015.

[16] Richard A Harshman. Foundations of the parafac procedure: Models and conditions for an" explanatory" multi-modal factor analysis. *UCLA working papers in phonetics*, 16:1–84, 1970.

[17] Pascal Held, Benjamin Krause, and Rudolf Kruse. Dynamic clustering in social networks using louvain and infomap method. *CoRR*, abs/1603.02413, 2016.

[18] Michael D Ward Peter D Hoff and Corey Lowell Lofdahl. Identifying international networks: Latent spaces and imputation. In *Dynamic Social Network Modeling and Analysis:: Workshop Summary and Papers*, page 345. National Academies Press, 2003.

[19] Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of classification*, 2(1):193–218, 1985.

[20] Mark Huisman. Imputation of missing network data: some simple procedures. *Journal of Social Structure*, 10(1):1–29, 2009.

[21] Joseph G Ibrahim, Ming-Hui Chen, Stuart R Lipsitz, and Amy H Herring. Missing-data methods for generalized linear models: A comparative review. *Journal of the American Statistical Association*, 100(469):332–346, 2005.

[22] Paul Jaccard. The distribution of the flora in the alpine zone. 1. *New phytologist*, 11(2):37–50, 1912.

[23] Raghunandan H Keshavan, Andrea Montanari, and Sewoong Oh. Matrix completion from a few entries. *Information Theory, IEEE Transactions on*, 56(6):2980–2998, 2010.

[24] Myunghwan Kim and Jure Leskovec. The network completion problem: Inferring missing nodes and edges in networks. In *Society for Industrial and Applied Mathematics. Proceedings of the SIAM International Conference on Data Mining*, page 47. Society for Industrial and Applied Mathematics, 2011.

[25] Myunghwan Kim and Jure Leskovec. The network completion problem: Inferring missing nodes and edges in networks. In *Proceedings of the Eleventh SIAM International Conference on Data Mining, SDM 2011, April 28-30, 2011, Mesa, Arizona, USA*, pages 47–58, 2011.

[26] Myunghwan Kim and Jure Leskovec. Nonparametric multi-group membership model for dynamic networks. In *Conference on Neural Information Processing Systems 2013.*, pages 1385–1393, 2013.

[27] Tamara G Kolda and Brett W Bader. Tensor decompositions and applications. *SIAM review*, 51(3):455–500, 2009.

[28] Gueorgi Kossinets. Effects of missing data in social networks. *Social networks*, 28(3):247–268, 2006.

[29] Christine Largeron, Pierre-Nicolas Mougel, Reihaneh Rabbany, and Osmar R. Zaïane. Generating attributed networks with communities. *PLoS ONE*, 10(4):e0122777, 04 2015.

[30] Collins Leke, Bhekisipho Twala, and Tshilidzi Marwala. Modeling of missing data prediction: Computational intelligence and optimization algorithms. In *Systems, Man and Cybernetics (SMC), 2014 IEEE International Conference on*, pages 1400–1404. IEEE, 2014.

[31] David Liben-Nowell and Jon Kleinberg. The link-prediction problem for social networks. *Journal of the American society for information science and technology*, 58(7):1019–1031, 2007.

[32] Roderick J Little, Ralph D'Agostino, Michael L Cohen, Kay Dickersin, Scott S Emerson, John T Farrar, Constantine Frangakis, Joseph W Hogan, Geert Molenberghs, Susan A Murphy, et al. The prevention and treatment of missing data in clinical trials. *New England Journal of Medicine*, 367(14):1355–1360, 2012.

[33] Roderick JA Little and Donald B Rubin. *Statistical Analysis with Missing Data*. Wiley, 2002.

[34] Peter J Mucha, Thomas Richardson, Kevin Macon, Mason A Porter, and Jukka-Pekka Onnela. Community structure in time-dependent, multiscale, and multiplex networks. *science*, 328(5980):876–878, 2010.

[35] G Palla, Barabasi AL, and Vicsek T. Quantifying social group evolution. *Nature*, 446:664–667, 2007.

[36] Martin Rosvall and Carl T Bergstrom. Multilevel compression of random walks on networks reveals hierarchical organization in large integrated systems. *PloS one*, 6(4):e18209, 2011.

[37] DB Rubin. *Multiple Imputation for Nonresponse in Surveys*. J. Wiley & Sons, New York, 1987.

[38] Jeanne A Saunders, Nancy Morrow-Howell, Edward Spitznagel, Peter Doré, Enola K Proctor, and Richard Pescarino. Imputing missing data: A comparison of methods for social work researchers. *Social work research*, 30(1):19–31, 2006.

[39] Joseph L Schafer and John W Graham. Missing data: our view of the state of the art. *Psychological methods*, 7(2):147, 2002.

[40] Peter Schmitt, Jonas Mandel, and Mickael Guedj. A comparison of six methods for missing data imputation. *Journal of Biometrics & Biostatistics*, 2015, 2015.

[41] Rajesh Sharma, Matteo Magnani, and Danilo Montesi. Investigating the types and effects of missing data in multilayer networks. In *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2015)*, pages 392–399, 2015.

[42] Alexander Strehl and Joydeep Ghosh. Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *The Journal of Machine Learning Research*, 3:583–617, 2003.

[43] Yizhou Sun, Jie Tang, Jiawei Han, Manish Gupta, and Bo Zhao. Community evolution detection in dynamic heterogeneous information networks. In *Proceedings of the Eighth Workshop on Mining and Learning with Graphs*, MLG '10, pages 137–146, 2010.

[44] Xuning Tang and Christopher C. Yang. Detecting social media hidden communities using dynamic stochastic blockmodel with temporal dirichlet process. *ACM Trans. Intell. Syst. Technol.*, 5(2):36:1–36:21, 2014.

[45] Giorgio Tomasi and Rasmus Bro. Parafac and missing values. *Chemometrics and Intelligent Laboratory Systems*, 75(2):163–180, 2005.

[46] Nicolas Voirin, Cécile Payet, Alain Barrat, Ciro Cattuto, Nagham Khanafer, Corinne Régis, Byeul-a Kim, Brigitte Comte, Jean-Sébastien Casalegno, Bruno Lina, and Philippe Vanhems. Combining high-resolution contact data with virological data to investigate influenza transmission in a tertiary care hospital. *Infection Control and Hospital Epidemiology*, FirstView:1–7, 1 2015.

[47] B Walczak and DL Massart. Dealing with missing data: Part i. *Chemometrics and Intelligent Laboratory Systems*, 58(1):15–27, 2001.

[48] Yoshihiro Yamanishi, J-P Vert, and Minoru Kanehisa. Protein network inference from multiple genomic data: a supervised approach. *Bioinformatics*, 20(suppl 1):i363–i370, 2004.

[49] Bowen Yan and Steve Gregory. Finding missing edges and communities in incomplete networks. *Journal of Physics A: Mathematical and Theoretical*, 44(49):495102, 2011.

[50] Z Yang, R Algesheimer, and CJ Tessone. A comparative analysis of community detection algorithms on artificial networks. *Scientific Reports*, 6:30750, 2016.