Direction matters in complex networks: A theoretical and applied study for greedy modularity optimization

Nicolas Dugué, Anthony Perez

Physica A 603 (2022) 127798

Contents lists available at ScienceDirect

Physica A

journal homepage: www.elsevier.com/locate/physa

Direction matters in complex networks: A theoretical and applied study for greedy modularity optimization ☆

Nicolas Dugué [a,*], Anthony Perez [b]

[a] Le Mans Université, LIUM, EA 4023, Le Mans, 72000, France
[b] Univ. Orléans, INSA Centre Val de Loire, LIFO EA 4022, F-45067 Orléans, France

ARTICLE INFO

Article history:
Received 23 April 2022
Received in revised form 16 June 2022
Available online 22 June 2022

Keywords:
Complex networks
Community detection
Directed networks
Louvain
Modularity
Graphs

ABSTRACT

Many real-world systems can be modeled as *directed* networks, such as transportation, social, collaboration or vocabulary networks. However, direction is often neglected or even ignored in community detection algorithms. This is in particular the case on large networks, since there are only a few scalable algorithms at the time. One of the most used scalable algorithm, `Louvain`'s algorithm, is based on modularity maximization and commonly used for directed networks by forgetting direction. We show that this oversimplification in the modeling process may alter the quality of the results both theoretically and practically. Moreover, we introduced in a complementary version of this work the `Directed Louvain` algorithm based on directed modularity that found various successful applications that enlighten the importance of direction when detecting communities. We hence propose an overview of selected applications within some of the aforementioned fields. We hope that this study will encourage researchers to use directed modularity whenever it is relevant.

1. Introduction

In many applications involving the study of complex systems, agents and their interactions are modeled as networks, nodes being the agents and links the communication between these agents. It is often natural to consider links as *directed* to better represent reality. For instance, when dealing with online social networks with subscription or follow relationships [1], transportation networks describing travel from a departure to a destination [2–5], biological protein-to-protein interaction networks with signal transduction [6], words co-occurrence networks with the precedence relation of words [7], and so on. A common approach in all these various research fields is to use complex network analysis tools to understand and exploit the inherent structure of such networks. In particular, community detection algorithms are widely used to uncover the underlying mesoscopic structure, giving insights about the latent organization of the system. While there is no unique formal definition of the notion of communities, the idea is to group nodes in different parts that are densely connected, the density *between* parts being smaller. The never-ending growth in data to process and the constant increase in their dimensions force researchers of the field to produce scalable algorithms to detect such densely connected groups. To that extent, algorithms maximizing *modularity* have received a lot of attention due to their efficient implementation. Modularity was designed by Newman [8] to measure the quality of a *partition* as a community structure. It values the existence of an edge within a community compared to the probability of having

☆ A complementary version of this work is available as an unpublished open access research report: https://hal.archives-ouvertes.fr/hal-01231784/.

* Corresponding author.
E-mail addresses: nicolas.dugue@univ-lemans.fr (N. Dugué), anthony.perez@univ-orleans.fr (A. Perez).

https://doi.org/10.1016/j.physa.2022.127798
0378-4371/

such an edge (regardless of communities) between the corresponding vertices in a random model following the same degree distribution. While modularity is known to have some limitations [9,10] and its maximization to be NP-hard [11], algorithms based on heuristics maximizing this measure remain to this day the most efficient on large networks [12,13]. One of the most used algorithm maximizing modularity is the Louvain's algorithm [12]. A major disadvantage of the algorithm in its most common shape is that it does not deal with *directed* networks. More generally, despite some work that was done evaluating *directed modularity* and its relevance [13–15], algorithms maximizing *directed modularity* have received very little attention so far. Instead, the practical applications that need direction in their representation and scalable community detection algorithms often forget the direction to use Louvain's algorithm [16–20]. This is actually even more problematic when considering weighted networks, since an undirected edge uv representing arcs (u, v) and (v, u) may need to be weighted by the sum of both arcs. Both observations can introduce a tremendous bias in the experiments. The use of community detection algorithms in aforementioned fields of research is well illustrated in a recent survey by Javed et al. [21]. However, despite their importance, directed networks are only mentioned.

Our contribution. We provide a theoretical and applied analysis to emphasize the importance of direction in complex networks. We focus on the scalable Directed Louvain method based on modularity optimization that offers a great trade-off between running time and results [14]. We begin by considering related work in Section 2 and thus illustrating the relevance of greedy modularity maximization. Then, with a theoretical analysis, we illustrate the importance of considering direction when maximizing modularity by focusing on the difference that arises between the classical Louvain's algorithm and Directed Louvain when one forgets the direction of a given network (Section 3). We would like to mention that in their survey on community detection algorithms, Fortunato et al. [22] presented several metrics on networks and stated that *extensions of the metrics [in directed networks] are fairly simple to implement, though their usefulness is unclear*. Our analysis thus states the usefulness of considering direction regarding modularity and hence degrees of vertices. Besides, we illustrate the meaningfulness of using a directed version of Louvain's algorithm by providing an overview of practical applications that used the Directed Louvain algorithm [14] (Section 4). Indeed, since its release, this algorithm has been successfully used in dozens of practical applications (see e.g. [2–5,7,23–27]). We focus in particular on transportation networks, which are directed by nature (Section 4) and thus well-suited for our study. We finally summarize our observations in a Conclusion section.

2. Related work

Existing algorithms such as Oslom [28] can deal with directed networks to detect communities and are well-known for the quality of the communities they return. However, as shown in a complementary version of this work [14], their complexity is much higher than modularity-based algorithms, Oslom [28] requiring more than 10 h to compute the communities of a network with 325 k arcs. Recent results by Singha et al. [29] on Intel(R) Xeon(R) E5-2687 W v3 processor with 32GB of RAM show that it runs up to 90 times slower than its competitors integrated in Cytoscape. Algorithms such as Label Propagation are much more scalable with quasi-linear complexity, and Li [30] adapted this framework for directed graphs. However, these algorithms are very sensitive to the order of execution and may not converge. Another option is to use InfoMap [31] which is one of the competitors of Oslom [28] integrated in Cytoscape that scales efficiently according to Singha et al. [29]. Furthermore, Li [30] shows it outperforms Directed LPA algorithm most of the time, emphasizing that InfoMap [31] provides an interesting trade-off between running time and results. However, according to Singha et al. [29], the fastest algorithm remains Louvain (with an $O(m)$ complexity, m being the number of edges [32]), which enlightens the relevance of the Directed Louvain algorithm (introduced by the same set of authors in a complementary work [14]) w.r.t. computation time. Moreover, a recent study on Twitter data compared several approaches to detect communities in directed graphs, showing with quality measures the good results of Directed Louvain that ranked second or third out of nine methods for each criteria [33]. For the sake of completeness, let us mention that there exist few other approaches that optimize modularity. Kim et al. [34] described LinkRank and showed that optimizing Pagerank for links on a directed network is equivalent to maximize directed modularity, while no experiments are operated on real graphs. Yang et al. [35] used mathematical models based on integer linear programming to compute a non-overlapping partition that maximizes modularity. Their approach is divided in two steps, namely MINLP (Mixed Integer Non-linear Programming Model) and MIP (Mixed Integer Linear Programming Model). In a first step, the MINLP is solved quickly but may lead to local optimal region. To overcome this issue, a second step is applied that redefine non-linear constraints (one being within the objective function) into linear constraints. The authors mention that this part is harder to solve, and they hence provide an initial network division produced by the first step. While this method leads to significant results compared to other existing methods such as Extremal Optimization [36] or Fast Algorithm [8],[1] the computation time needed to obtain the community partition seems really high. Indeed, even if the authors do not explicitly evaluate the running time, they consider networks with a small number of vertices (6500) and edges (21,000) and set the limit for the resolution of *each model* to 1500 s. This seems to indicate that such a method is not suitable to deal with large networks, a feature that is known to exist for Directed Louvain. Osaba

[1] These two methods are designed for undirected networks and used forgetting direction, a classical methodology as mentioned in the introductory section.

et al. [37] propose nature-inspired algorithms such as evolutionary simulated annealing or water cycling algorithm. Such algorithms can perform very well but are usually highly dependent on their initialization parameters. Furthermore, authors do not compare their algorithms to state-of-the-art algorithms that do not require any parameter. Finally, prior to our implementation of `Directed Louvain` [14], and to the best of our knowledge, there was no reference implementation of `Louvain`'s algorithm for directed networks. The only known implementation was an unmaintained `MATLAB` (proprietary environment) implementation [38]. Since its introduction, the properties of the `C++` implementation[2] we developed have been integrated in the `scikit-network` python framework [39].

3. Maximizing modularity in (directed) networks

Modularity. A classical way of detecting communities in an undirected graph $G = (V, E)$ is to find a *partition* of the vertex set that maximizes some optimization function. One of the most famous optimization function to measure the quality of a community partition is called *modularity* [40]. Roughly speaking, given a partition of the vertices, this function values the existence of an edge within a part compared to the probability of having an edge (regardless of parts) between the corresponding vertices in a random model following the same degree distribution. Formally, the modularity Q of a partition $\mathcal{C} = \{C_1, \ldots, C_p\}$ of G is defined as follows [40,41]:

$$Q = \frac{1}{2m} \sum_{u,v} \left[E_{uv} - \frac{d_u \cdot d_v}{2m} \right] \delta(C_u, C_v)$$

where $m = |E|$ is the number of edges of G, E_{uv} represents the existence (0 or 1) of an edge between u and v, $d_u = |\{v \in V : uv \in E\}|$ is the *degree* of vertex u, C_u is the community of u and δ is the Kronecker delta function defined by $\delta(u, v) = 1$ if $u = v$, and 0 otherwise. Notice that the definition is given for unweighted networks but can be naturally extended when edges are weighted by some weight function $\omega : E \to \mathbb{R}^+$ by considering *weighted degrees*. However, for *signed* networks, *i.e* with a weighted function $\omega : E \to \mathbb{R}$ modularity requires a more intricate definition [42,43]. For the sake of simplicity, we henceforth consider *unweighted* networks. Arenas et al. [15] adapted the notion of modularity for directed graphs, which can be motivated by the following observation by Leicht and Newman [13]: consider two vertices u and v with u having small in-degree and large out-degree and v small out-degree and large in-degree. Then having an arc from v to u should be considered more surprising than having an arc from u to v. Taking this into account, the definition for the *directed modularity* of a partition of a directed network $D = (V, A)$ is formulated [15] as follows:

$$Q_d = \frac{1}{m} \sum_{u,v} \left[A_{uv} - \frac{d_u^{in} \cdot d_v^{out}}{m} \right] \delta(C_u, C_v)$$

where A_{uv} now represents the existence of an *arc* between u and v, and $d_u^{in} = |\{v \in V : (v, u) \in A\}|$ is the *in-degree* of u and $d_u^{out} = |\{v \in V : (u, v) \in A\}|$ its *out-degree*.

`Louvain`*'s algorithm.* We now briefly describe the behavior of `Louvain`'s algorithm to maximize modularity [12]. It relies on a greedy agglomerative procedure: starting from any partition of the vertices (usually the partition into singletons), the algorithm tries to increase the value of modularity by moving vertices from their community to any neighboring one. In other words, the algorithm computes the *gain* of modularity obtained by adding vertex u to community C as follows:

$$\Delta_Q = \frac{d_u^C}{2m} - \frac{\sum_{tot}^C \cdot d_u}{2m^2}$$

where $d_u^C = |\{v \in C : uv \in E\}|$ denotes the degree of node u in community C and $\sum_{tot}^C$ the number of edges incident to community C. The algorithm does a similar calculation to compute the *gain* obtained by *removing* vertex u from its own community C_u and then *agglomerates* all computed communities into a single vertex, resulting in a weighted network with self-loops. The procedure then carries on as long as there exists a move that improves the value of modularity, thus leading to a hierarchical community structure.

`Directed Louvain`*'s algorithm.* The behavior of the algorithm is the same in the directed case [14], the main difference lying in the calculation for the gain of modularity obtained by adding vertex u to community C, which can now be done using the following[3]:

$$\Delta_{Q_d} = \frac{d_u^C}{m} - \left[\frac{d_u^{out} \cdot \sum_{tot,in}^C + d_u^{in} \cdot \sum_{tot,out}^C}{m^2} \right]$$

where $\sum_{tot,in}^C$ (resp. $\sum_{tot,out}^C$) denotes the number of *incoming* (resp. *outcoming*) arcs incident to community C.

[2] https://github.com/nicolasdugue/DirectedLouvain.

[3] Notice that Leicht and Newman [13] provide a similar analysis but with a more intricate formulation based on so-called modularity matrix.

3.1. Theoretical comparison between directed and undirected modularity optimization

Arenas et al. [15] provide an expression of the directed modularity Q_D of a directed network w.r.t. modularity of the underlying undirected network. More precisely, given a (weighted) directed graph $D = (V, A)$ they consider the corresponding underlying (weighted) undirected graph $G = (V, E)$ where $uv \in E$ whenever (u, v) or $(v, u) \in A$. If both arcs (u, v) and (v, u) are present then one needs to weight the edge uv accordingly. The authors observe that the modularity Q_S of G is [15,44]:

$$Q_S = Q_D - \frac{1}{(4m)^2} \sum_{u,v} (d_u^{out} - d_u^{in}) \cdot (d_v^{out} - d_v^{in}) \cdot \delta(C_u, C_v)$$

Instead of comparing modularity *values*, we hereafter compare the conditions needed to merge communities during the greedy agglomerative procedure when maximizing (directed) modularity in both G and D. Let C_1 and C_2 be two communities uncovered. We name $A_{1,2}$ the arcs between communities C_1 and C_2, and $E_{1,2}$ the edges in the corresponding undirected graph G. Notice that we have $|E_{1,2}| = |A_{1,2}|$.

Undirected graphs. When C_1 and C_2 are considered as part of the same community, $|E_{1,2}|$ links contribute to increase modularity value, as shown in bold in the following formula:

$$Q^{C_1 \cup C_2} = \left(\frac{\sum_{in}^{C_1}}{m} + \frac{\sum_{in}^{C_2}}{m} + \frac{\mathbf{|E_{1,2}|}}{\mathbf{m}} \right) - \left(\sum_{u,v \in C_1} \frac{d_u \cdot d_v}{4m^2} + \sum_{u,v \in C_2} \frac{d_u \cdot d_v}{4m^2} + \sum_{\mathbf{u \in C_1, v \in C_2}} \frac{\mathbf{d_u \cdot d_v}}{\mathbf{2m^2}} \right)$$

When C_1 and C_2 are two different communities, both terms in bold disappear:

$$Q^{C_1, C_2} = \left(\frac{\sum_{in}^{C_1}}{m} + \frac{\sum_{in}^{C_2}}{m} \right) - \left(\sum_{u,v \in C_1} \frac{d_u \cdot d_v}{4m^2} + \sum_{u,v \in C_2} \frac{d_u \cdot d_v}{4m^2} \right)$$

Thus, if summing these bold terms results in a positive number, C_1 and C_2 are merged. At the contrary, if the sum is negative, C_1 and C_2 are considered as two distinct communities. Hence, communities C_1 and C_2 are merged when the following holds:

$$\frac{1}{m} \left(|E_{1,2}| - \sum_{u \in C_1, v \in C_2} \frac{d_u \cdot d_v}{2m} \right) > 0$$

$$\Leftrightarrow |E_{1,2}| > \sum_{u \in C_1, v \in C_2} \frac{d_u \cdot d_v}{2m} \tag{1}$$

Directed graphs. A similar result holds for directed graphs, both communities C_1 and C_2 being merged when:

$$\frac{1}{2m} \left(|A_{1,2}| - \sum_{u \in C_1, v \in C_2} \left(\frac{d_u^{in} \cdot d_v^{out} + d_u^{out} \cdot d_v^{in}}{2m} \right) \right) > 0$$

$$\Leftrightarrow |A_{1,2}| > \sum_{u \in C_1, v \in C_2} \left(\frac{d_u^{in} \cdot d_v^{out} + d_u^{out} \cdot d_v^{in}}{2m} \right) \tag{2}$$

Comparison. One can compare the choices made by algorithms by replacing the vertex degree of Eq. (1) by its incoming and outcoming counterparts: $d_u = (d_u^{in} + d_u^{out})$. We thus obtain the following condition for merging C_1 and C_2 in G:

$$|E_{1,2}| > \underbrace{\sum_{\substack{u \in C_1 \\ v \in C_2}} \left(\frac{d_u^{in} \cdot d_v^{out} + d_u^{out} \cdot d_v^{in}}{2m} \right)}_{S} + \underbrace{\sum_{\substack{u \in C_1 \\ v \in C_2}} \left(\frac{d_u^{in} \cdot d_v^{in} + d_u^{out} \cdot d_v^{out}}{2m} \right)}_{T}$$

In the undirected case, C_1 and C_2 are thus merged when $|E_{1,2}| > S + T$ while in the directed case, the fusion is done when $|A_{1,2}| > S$ (Eq. (2)), T being absent from the equation. Since $|E_{1,2}| = |A_{1,2}|$ this may have a significative impact in the computed communities. Notice that the term S confirms the observation made by Leicht and Newman [13]. Furthermore, we can see that term T is not relevant: multiplying the incoming degrees of both nodes in one side and their outgoing degrees on the other side does not allow to estimate the probability of the existence of an arc between u and v in a random model.

Table 1
Comparison between directed and undirected algorithms on directed graphs with groundtruth communities using Normalized Mutual Information (NMI) and Adjusted Mutual Information (AMI).

	Louvain		Louvain-dir		ECG		ECG-dir	
	NMI	AMI	NMI	AMI	NMI	AMI	NMI	AMI
Pol. blogs	**0.63**	**0.63**	0.62	0.61	0.64	0.64	**0.65**	**0.65**
Eu-core	0.54	0.51	**0.64**	**0.61**	0.56	0.54	**0.63**	**0.60**
Eurosis	0.84	0.83	0.84	0.83	0.84	0.83	**0.86**	**0.85**

Table 2
Average number of communities detected for each algorithm, with $|C|$ the groundtruth.

	$\|C\|$	Louvain	Louvain-dir	ECG	ECG-dir
Pol. blogs	2	9	14	8	10
Eu-core	42	6	18	7	10
Eurosis	12	16	22	16	20

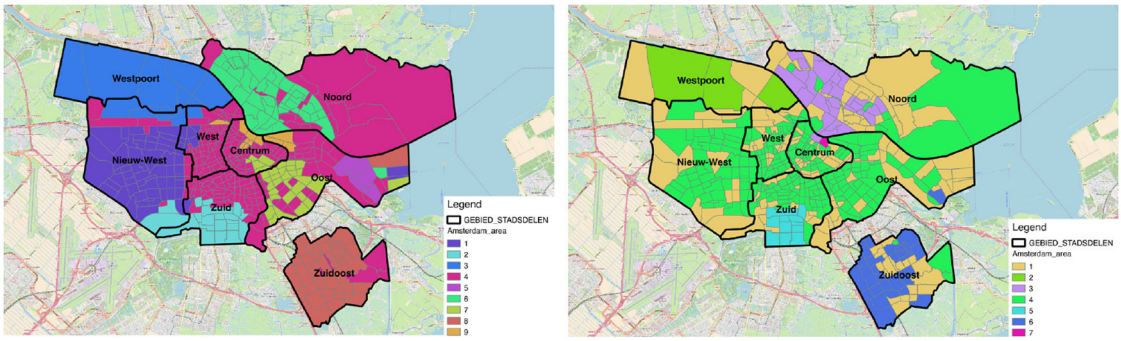
Experiments on real-graphs. In Section 4, we will report various experiments showing the relevance of the Directed Louvain approach. Most of these results being based on expert assessment, we first run some experiments on directed graphs with groundtruth communities to evaluate more thoroughly the performances of Directed Louvain when compared to its undirected version. Poulin and Théberge [45] showed that the nondeterministic feature of Louvain's algorithm can actually be used to design a more efficient algorithm based on consensus, the *Ensemble Clustering for Graphs* (ECG). We thus report results with the classic Louvain algorithm, Directed Louvain (Louvain-dir), ECG, and ECG-dir, the ECG algorithm we adapted to make it run with Directed Louvain. Considering that these algorithms are nondeterministic, we report results averaged on 50 runs for each method (even ECG and ECG-dir). For one run of ECG, because it is based on consensus, we run 64 times the Louvain algorithm to uncover the consensus partition. We consider three datasets, the Political Blogs of Adamic and Glance [46], Eurosis introduced by Van Welden [47] and Eu-core from Snap [48].

As one can see Table 1, except for the political blogs, Directed Louvain results are always better than for the undirected version. When considering ECG, results are always better (and even improved) when using the directed version except for Eu-core where the Directed Louvain itself obtains the best results. Furthermore, the directed version of the algorithm uncovers more communities (see Table 2), which is relevant in the case of Eu-core, but not consistent with the groundtruth for the other cases. The relatively low improvements may be related to the very symmetric nature of the graphs considered. In other cases, we may hope for better results. Unfortunately, there are only few directed networks with groundtruth communities available. Still, as we shall see in Section 4, results with Directed Louvain are more consistent according to experts, in particular with transportation networks.

4. Overview of applications

As mentioned in the introductory section, Directed Louvain has been successfully used in many real-life applications since its release (see *e.g.* [2–5,7,23–27]. In many practical applications the underlying graph has to be directed, a feature commonly ignored to exploit community detection algorithms (see for instance [18] where the authors use Louvain's algorithm to visualize scientific publications). Before focusing on applications related to transportation networks which are naturally directed, let us mention that Directed Louvain has recently been used in the field of scientometrics, an active and important research area. Using Directed Louvain, Pramanik et al. [49] obtained interesting observations on the migration of researchers across scientific domains, thus opening a new research orientation. Another example of application lies in the field of natural language processing. Gómez-Suta et al. [7] proposed a semi-automatic transformation of Spanish texts to ontology structures as terms, concepts and relations between them. In their work, the authors use community detection as a preprocessing step before semantic clustering [7].

In the remaining of this section we focus on applications related to transportation networks, which are directed by nature. Such applications enlighten the relevance of Directed Louvain since in several cases the results obtained with such an algorithm are more consistent than the ones obtained using classical Louvain's algorithm. Since its release, Directed Louvain found several applications in transportation networks [2–5,50]. In all cases, the method was used as a subroutine to analyze and understand various transportation networks. As a warm-up, let us mention a project for Stanford's course entitled *Analysis of networks, mining and learning with graphs* conducted by Daniel Gardner [23]. In this project, the author focused on home-to-work routes in the San Francisco Bay Area, the network being directed and weighted according to the frequency of the routes. The author used both classical and directed versions of Louvain's algorithm on the aforementioned network, where nodes correspond to neighborhood blocks and are connected when at least one worker commutes from one block to another. While the undirected (unweighted) version of Louvain's algorithm produces good results, including both weights and direction provides more insightful information. This study



(a) Louvain directed (b) Louvain undirected

Fig. 1. Communities obtained with respect to destination for each Amsterdam's district (extracted from [2]).

Table 3
Quality assessment of disjoint community detection (extracted from [3]).

Algorithms	Q	Number of communities	Time complexity
`OSLOM`	0.78	16	$\sim$ 40 min
`Directed Louvain`	0.78	17	$\sim$ 0.3 s

leads to the observation that `Directed Louvain` produces superior communities than the classical `Louvain`'s algorithm on the underlying undirected graph, with communities corresponding to spatially contiguous neighborhoods which made sense geographically [23]. We now provide more thoroughly other examples of applications of `Directed Louvain` in transportation networks.

Travel behavior in Amsterdam [2]. In this work, van Leeuwen et al. [2] analyze travel behavior in Amsterdam based on time dependent origin–destination electronic trace data. Using directed modularity, their work distinguishes spatially connected regions. More precisely, they make use of travel movements registered by Google on Android phones from the Amsterdam region [2]. This results in a directed weighted network where nodes represent neighborhoods of the Amsterdam regions and where arcs represent an origin–destination pair weighted according to the travel intensity across the respective pair. All weights are normalized with respect to the largest hourly intensity (over a 6 months period). In order to compute communities, the authors tried several heuristics such as `InfoMap` [31] and `Oslom` [28]. They observed that all such methods either failed to converge or returned modularity values close to 0 [2]. Due to its ability to find spatially connected clusters with no spatial information included [12], `Louvain`'s algorithm was a natural choice. However, the authors emphasize that forgetting direction and using `Louvain`'s algorithm provides worse results (see Fig. 1).

They thus turned their attention to *directed modularity*, using the available `MATLAB` implementation of Scherrer [38]. As one can see Fig. 1, including direction in `Louvain`'s algorithm results in spatially close clusters, some communities having a close resemblance with the regional districts of Amsterdam. These findings are of important interest and may support policy makers in their decisions for future infra structural adjustments [2], thus illustrating the fundamental aspect of direction for modularity maximization heuristics.

Hangzhou's urban bus systems [3]. Wang et al. [3] conducted an analysis of the urban bus spatial network of the downtown area of Hangzhou, China. Networks are a natural representation of urban bus transportation systems since they comprise bus stations and routes that cover the entire urban area [3]. The routes being naturally oriented, the underlying network must be considered directed. The authors hence considered two distinct graphs, namely from bus station connections and from connections between hexagonal spatial units (with an area of 1 km^2). In both cases there is an arc between two nodes whenever there exists a directed route connecting both bus stations (respectively both spatial units). Moreover, arcs are weighted according to the number of routes existing between the corresponding nodes.

The authors then conduct a thorough analysis of such networks, and consider in particular their macroscopic characteristics. We focus on the spatial unit network which contains 3250 vertices and 133,539 arcs.

As mentioned by Wang et al. [3], community detection algorithms considering both directions and weights are relatively rare. They thus used `Directed Louvain` as well as `Oslom` [28] to compute disjoint communities. The obtained results are reported Table 3 and illustrate that `Directed Louvain` outperforms `Oslom` [28] in this particular case, mainly from the time complexity viewpoint.

The authors notice that although the geospatial distances and the spatial relationships between nodes were not considered in community detection algorithms (recall that arcs correspond to *routes* between spatial units and weights to the numbers of such routes) the results present apparent geographical proximity [3]. Such observations have insight and implications for spatial planning and development of urban bus systems.

Topological analysis of traffic pace [5]. Another notable example is the recent work of Carmody and Sowers [5] who provide a topological analysis of traffic pace using persistent homology. The field of *topological data analysis* proposes methods in order to identify objects which remain invariant under different perspectives [5]. More precisely, the authors aim at understanding macroscopic topological structures of *traffic networks* from local data. Based on persistent homology (used to identify features of a dataset in presence of topological noise [5]), this work aims at properly addressing the effect of directionality, which has been neglected so far. In order to conduct such a topological analysis, the authors first need to define so-called Rips complexes of directed networks, that is a family of simplicial complexes. To do so, Carmody and Sowers [5] begin with a weighted directed graph modeling a road network with nodes representing intersections and arcs and weights corresponding to roads and their respective traffic paces. Since topological data analysis depends on a notion of *nearness* [5], the authors need to exploit both weights and direction to define some *distance* (which in this case correspond to shortest weighted directed paths). The persistent homology of Rips complexes associated to directed graphs can thus be compared through a topological notion of distance. Unfortunately, the complexity of large road networks poses some significant computational challenges [5]. To circumvent this issue, the authors first apply the `Directed Louvain` algorithm to *coarse-grain* the network into statistically similar neighborhoods [5]. The hierarchical structure of the algorithm is also exploited. This is used as a preprocessing step, and the authors thus emphasize that coarse-graining a traffic network using `Directed Louvain` preserves important topological features [5]. They successfully illustrate their work on both Manhattan [51] and Chengdu [52] datasets.

5. Conclusion

Despite its obvious importance when modeling a complex network, direction has been commonly neglected (or simply ignored) when detecting communities. This is in particular the case when considering algorithms that maximize modularity [18]. However, as illustrated in the theoretical part of this work as well on the overview of applications, direction may have a great impact on results and it hence seems really important to use frameworks that can deal with directed networks. By providing the first stand-alone implementation of `Directed Louvain` we proposed a scalable solution to circumvent this issue [14]. Even if the improvements compared to the undirected version seem modest in our experiments on real graphs, this may be due to the fact that these graphs are somehow symmetric. Indeed, since its release, `Directed Louvain` found applications in various fields and helped develop better and more interpretable analysis of important problems in various fields of research [2–5,7,23–27]. We hence hope that this overview of applications as well as the provided theoretical analysis will encourage researchers to consider direction in their work detecting communities. To this day, direction is still commonly ignored [16–19] and it would be interesting to see if these results can be improved using `Directed Louvain`, even if modularity maximization has a resolution limit [10]. To circumvent this issue one may eventually consider variations of modularity such as the *modularity difference* introduced in [53]. Notice however that in most cases this would have a significant impact in the computation time.

CRediT authorship contribution statement

Nicolas Dugué: Methodology, Software, Formal analysis, Writing and editing. **Anthony Perez:** Methodology, Software, Formal analysis, Writing and editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] M. Cha, H. Haddadi, F. Benevenuto, K.P. Gummadi, Measuring user influence in Twitter: The million follower fallacy, in: ICWSM '10: Proc. of Int. AAAI Conference on Weblogs and Social, 2010, pp. 1–17.
[2] D. van Leeuwen, J.W. Bosman, E.R. Dugundji, Network partitioning on time-dependent origin-destination electronic trace data, Pers. Ubiquitous Comput. 23 (5) (2019) 687–706.
[3] Y. Wang, Y. Deng, F. Ren, R. Zhu, P. Wang, T. Du, Q. Du, Analysing the spatial configuration of urban bus networks based on the geospatial network analysis method, Cities 96 (2020) 102406.
[4] A. Furno, N.-E.E. Faouzi, R. Sharma, E. Zimeo, Graph-based ahead monitoring of vulnerabilities in large dynamic transportation networks, PLoS One 16 (3) (2021) e0248764.
[5] D.R. Carmody, R.B. Sowers, Topological analysis of traffic pace via persistent homology, J. Phys.: Complexity 2 (2) (2021) 025007.
[6] A. Gitter, J. Klein-Seetharaman, A. Gupta, Z. Bar-Joseph, Discovering pathways by orienting edges in protein interaction networks, Nucleic Acids Res. 39 (4) (2011) e22.
[7] M. Gómez-Suta, J.D. Echeverry-Correa, J.A. Soto-Mejía, Semi-automatic extraction and validation of concepts in ontology learning from texts in spanish, in: WIMS, 2020, pp. 7–16.
[8] M.E. Newman, Fast algorithm for detecting community structure in networks, Phys. Rev. E 69 (6) (2004) 066133.
[9] S. Fortunato, M. Barthelemy, Resolution limit in community detection, Proc. Natl. Acad. Sci. 104 (1) (2007) 36–41.
[10] A. Lancichinetti, S. Fortunato, Limits of modularity maximization in community detection, Phys. Rev. E 84 (6) (2011) 066122.
[11] U. Brandes, D. Delling, M. Gaertler, R. Görke, M. Hoefer, Z. Nikoloski, D. Wagner, Maximizing modularity is hard, 2006, arXiv preprint Physics/0608255.

[12] V.D. Blondel, J.-L. Guillaume, R. Lambiotte, E. Lefebvre, Fast unfolding of communities in large networks, J. Stat. Mech. Theory Exp. 2008 (10) (2008) P10008.
[13] E.A. Leicht, M.E.J. Newman, Community structure in directed networks, Phys. Rev. Lett. 100 (11) (2008) 118703.
[14] N. Dugué, A. Perez, Directed Louvain: maximizing modularity in directed networks, Research Report, Université d'Orléans, 2015, URL https://hal.archives-ouvertes.fr/hal-01231784.
[15] A. Arenas, J. Duch, A. Fernández, S. Gómez, Size reduction of complex networks preserving modularity, New J. Phys. 9 (6) (2007) 176.
[16] B. Evkoski, I. Mozetic, N. Ljubesic, P.K. Novak, Community evolution in retweet networks, 2021, arXiv:2105.06214.
[17] P. Schmid, A. García-Gutierrez, J. Jiménez, Description and detection of burst events in turbulent flows, J. Phys.: Conf. Series 1001 (1) (2018) 012015.
[18] Q. Ping, C. Chen, LitStoryTeller+: an interactive system for multi-level scientific paper visual storytelling with a supportive text mining toolbox, Scientometrics 116 (3) (2018) 1887–1944.
[19] C. Unger, D. Murthy, A. Acker, I. Arora, A. Chang, Examining the evolution of mobile social payments in Venmo, in: International Conference on Social Media and Society, 2020, pp. 101–110.
[20] T. Prouteau, V. Connes, N. Dugué, A. Perez, J.-C. Lamirel, N. Camelin, S. Meignier, SINr: Fast computing of sparse interpretable node representations is not a Sin!, in: International Symposium on Intelligent Data Analysis, 2021, pp. 325–337.
[21] M.A. Javed, M.S. Younis, S. Latif, J. Qadir, A. Baig, Community detection in networks: A multidisciplinary review, J. Netw. Comput. Appl. 108 (2018) 87–111.
[22] S. Fortunato, D. Hric, Community detection in networks: A user guide, Phys. Rep. 659 (2016) 1–44.
[23] D. Gardner, Evolving community structures in a geographic commuting graph, 2018, Project for Stanford course CS224W — Analysis of Networks, http://snap.stanford.edu/class/cs224w-2018/projects.html.
[24] K.J. Dooley, S.D. Pathak, T.J. Kull, Z. Wu, J. Johnson, E. Rabinovich, Process network modularity, commonality, and greenhouse gas emissions, J. Oper. Manage. 65 (2) (2019) 93–113.
[25] P. Umar, C. Akiti, A. Squicciarini, S. Rajtmajer, Self-disclosure on Twitter during the COVID-19 pandemic: A network perspective, in: ECML KDD, 2021, pp. 271–286.
[26] A.N. Wickramasinghe, S. Muthukumarana, Social network analysis and community detection on spread of COVID-19, Model Assist. Stat. Appl. 16 (1) (2021) 37–52.
[27] A. Wickramasinghe, S. Muthukumarana, Assessing the impact of the density and sparsity of the network on community detection using a Gaussian mixture random partition graph generator, Int. J. Inf. Technol. (2022) 1–12.
[28] A. Lancichinetti, F. Radicchi, J.J. Ramasco, S. Fortunato, Finding statistically significant communities in networks, PLoS ONE 6 (5) (2011).
[29] A. Singhal, S. Cao, C. Churas, D. Pratt, S. Fortunato, F. Zheng, T. Ideker, Multiscale community detection in cytoscape, PLoS Comput. Biol. 16 (10) (2020) e1008239.
[30] X. Li, Directed LPA: Propagating labels in directed networks, Phys. Lett. A 383 (8) (2019) 732–737.
[31] M. Rosvall, D. Axelsson, C.T. Bergstrom, The map equation, Eur. Phys. J. Spec. Top. 178 (1) (2009) 13–23.
[32] V.A. Traag, Faster unfolding of communities: Speeding up the Louvain algorithm, Phys. Rev. E 92 (3) (2015) 032801.
[33] I. Blekanov, S.S. Bodrunova, A. Akhmetov, Detection of hidden communities in Twitter discussions of varying volumes, Future Internet 13 (11) (2021) 295.
[34] Y. Kim, S.-W. Son, H. Jeong, Finding communities in directed networks, Phys. Rev. E 81 (1) (2010) 016103.
[35] L. Yang, J.C. Silva, L.G. Papageorgiou, S. Tsoka, Community structure detection for directed networks through modularity optimisation, Algorithms 9 (4) (2016) 73.
[36] J. Duch, A. Arenas, Community detection in complex networks using extremal optimization, Phys. Rev. E 72 (2) (2005) 027104.
[37] E. Osaba, J. Del Ser, D. Camacho, A. Galvez, A. Iglesias, I. Fister, Community detection in weighted directed networks using nature-inspired heuristics, in: International Conference on Intelligent Data Engineering and Automated Learning, Springer, 2018, pp. 325–335.
[38] A. Scherrer, MATLAB implementation of Louvain's algorithm, https://perso.uclouvain.be/vincent.blondel/research/louvain.html.
[39] T. Bonald, N. de Lara, Q. Lutz, B. Charpentier, Scikit-network: Graph analysis in python, J. Mach. Learn. Res. 21 (185) (2020) 1–6.
[40] M.E. Newman, M. Girvan, Finding and evaluating community structure in networks, Phys. Rev. E 69 (2) (2004) 026113.
[41] A. Clauset, M.E. Newman, C. Moore, Finding community structure in very large networks, Phys. Rev. E 70 (6) (2004) 066111.
[42] S. Gómez, P. Jensen, A. Arenas, Analysis of community structure in networks of correlated data, Phys. Rev. E 80 (1) (2009) 016114.
[43] V.A. Traag, J. Bruggeman, Community detection in networks with positive and negative links, Phys. Rev. E 80 (3) (2009) 036115.
[44] F.D. Malliaros, M. Vazirgiannis, Clustering and community detection in directed networks: A survey, Phys. Rep. 533 (4) (2013) 95–142.
[45] V. Poulin, F. Théberge, Ensemble clustering for graphs: comparisons and applications, Appl. Netw. Sci. 4 (1) (2019) 1–13.
[46] L.A. Adamic, N. Glance, The political blogosphere and the 2004 US election: divided they blog, in: 3rd International Workshop on Link Discovery, 2005, pp. 36–43.
[47] D. Van Welden, Mapping system theory problems to the field of knowledge discovery in databases, in: Fubutec'2004, Eurosis, 2004, pp. 55–59.
[48] J. Leskovec, J. Kleinberg, C. Faloutsos, Graph evolution: Densification and shrinking diameters, TKDD 1 (1) (2007) 2–es.
[49] S. Pramanik, S.T. Gora, R. Sundaram, N. Ganguly, B. Mitra, On the migration of researchers across scientific domains, in: Proceedings of the International AAAI Conference on Web and Social Media, Vol. 13, 2019, pp. 381–392.
[50] X. Liu, J. Huang, J. Lai, J. Zhang, A.M. Senousi, P. Zhao, Analysis of urban agglomeration structure through spatial network and mobile phone data, Trans. GIS (2021).
[51] B. Donovan, A. Mori, N. Agrawal, Y. Meng, J. Lee, D. Work, New york city hourly traffic estimates (2010–2013), 2016.
[52] F. Guo, D. Zhang, Y. Dong, Z. Guo, Urban link travel speed dataset from a megacity road network, Scientific Data 6 (1) (2019) 1–8.
[53] F.N. Silva, A. Albeshri, V. Thayananthan, W. Alhalabi, S. Fortunato, Robustness modularity in complex networks, Phys. Rev. E 105 (5) (2022) 054308.