Ergodic sets in directed networks: a dynamics-based simplification

Erik Hörmann Renaud Lambiotte

November 7, 2022

Keywords: ergodic sets, directed networks, random walks,coarse-graining

Abstract

We introduce ergodic sets, strongly connected components of directed networks [1] that are connected in either direction to the rest of the graph, but not in both directions. We use these newly introduced structures to propose a coarse-graining method for large complex directed networks, which preserve the random walk dynamics of the original network.

1 Introduction

We start by formulating a mathematical definition of the generalized ergodic sets, from a purely structural construction that only depends on the network wiring.

Definition 1. *Let $\mathcal{G} = (V, E)$ be a directed graph. We say $X \subseteq V$ is an* ***ergodic set*** *if X is strongly connected and* ***either*** *of the two following sets of conditions hold:*

1. $X = V$
2. $X \subset V$ *and* ***all*** *the following conditions hold:*
   (a) *if $\exists x^* \in X$, $v^* \in V \setminus X$ such that $(x^* \to v^*) \in E$, then*
   $$\forall x \in X, \forall v \in V \setminus X \Rightarrow (v \to x) \notin E \tag{1}$$
   (b) *if $\exists x^* \in X$, $v^* \in V \setminus X$ such that $(v^* \to x^*) \in E$, then*
   $$\forall x \in X, \forall v \in V \setminus X \Rightarrow (x \to v) \notin E \tag{2}$$

Furthermore, we call input an ergodic set that has out-edges to the rest graph (i.e. one for which equation (2) is non-empty), and an output an ergodic set that has in-edges from the rest graph (i.e. one for which equation (1) is non-empty). Note that an ergodic set must either be an output, or an input, or a strongly connected component disjoint from the rest of the graph. Furthermore, an ergodic set X cannot be both an input and an output, as the two conditions are mutually contradicting.

2 Coarse-graining with ergodic sets

The ergodic sets has been purposely defined so that they can be used to simplify the structure of complex directed networks. One of the issues of directed networks are the presence of sources and wells, nodes that have zero in- and out-degree, respectively. These structures have risen to prominence in the seminal paper of Page and Brin [2], where these types of nodes precluded web crawler to explore the entire word wide web network, and this limitation prompted the development of the well-known PageRank algorithm by the same authors. The concept of ergodic sets allows us to generalise sources and wells to groups of more than one node.

2.1 Detecting ergodic sets

The first step is to find the strongly connected components of the network. To do so, we rely on the algorithm implemented in the standard library. For example, the NetworkX algorithm uses Tarjan's algorithm [3] with Nuutila's [4] modifications.

We now need to implement an algorithm to efficiently check whether the strongly connected components satisfy the additional conditions in the definition of ergodic sets, equations (1) and (2).

We use the sum of the in- and out-degrees of all nodes in the ergodic set $v \in X \subseteq V$,

we compare these two quantities in the subgraph X properly and in X as induced subgraph X_I. More formally, the following relations hold:

$$\left(\sum_{v \in X} d_{out}(v) - \sum_{v \in X_I} d_{out}(v) \neq 0 \Rightarrow \sum_{v \in X} d_{in}(v) - \sum_{v \in X_I} d_{in}(v) = 0 \right) \Leftrightarrow \text{equation (1)} \tag{3}$$

$$\left(\sum_{v \in X} d_{in}(v) - \sum_{v \in X_I} d_{in}(v) \neq 0 \Rightarrow \sum_{v \in X} d_{out}(v) - \sum_{v \in X_I} d_{out}(v) = 0 \right) \Leftrightarrow \text{equation (2)} \tag{4}$$

which allows us to quickly and efficiently test whether the strong connected component is also an ergodic set.

2.2 Coarse-graining

If X is an output, just collapse all vertices in one and keep all the in-edges. If a single vertex from $v \in V \setminus X$ points to more than one node of a single ergodic set, only add one edge. If a single vertex from $v \in V \setminus X$ points to multiple vertices across different outputs, $O_1, O_2, \ldots, O_n$ add one edge per ergodic set with weight proportional to the number of vertices in the ergodic set to which $v$is connected to.

If X is an input, proceed as follows. Substitute X with a single node v_X and connected with all the nodes w for which $\exists v \in X : (v, w) \in E$. all the out-and in-degrees from the other vertices of the graph an in- or out-edge with weight

$$\omega_w = \frac{1}{\sum_{v \in X} d_{in}(v)} \sum_{v \in X} \frac{d_{in}(v)}{d_{out}(v)} \mathbb{I}_E(v, w) \tag{5}$$

in which

$$\mathbb{I}_E(v, w) = \begin{cases} 1 & \text{if}(v, w) \in E \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

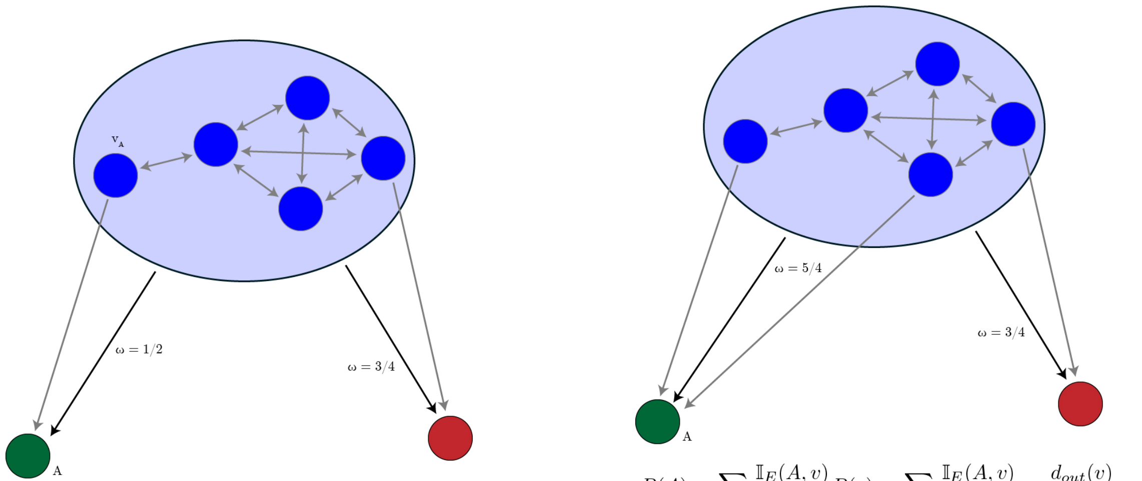
It is clear by construction that the reachable vertices w, as well as the probability of reaching them from the ergodic set X does not change when substituting X with its coarse-grained vertex v_X.

In order to have an equivalent dynamics, however, we need to set the hitting time for the outgoing nodes such that is the same before and after the substitution. We have one more degree of freedom to fix this extra requirement, that is, the weight ω_s of a self loop for v_X. Using the fact that the exit time of a random walk is an exponential variable, we see that we need to set ω_s such that:

$$\frac{\omega_s}{\omega_s + \sum_i \omega_i} = \sum_{v \in X} \frac{d_{in}(v)}{\sum_{w \in X} d_{in}(w)} \frac{d^{\dagger}_{out}(v)}{d_{out}(v)} \tag{7}$$

in which we have indicated the (out-)degrees of the vertices in X as a proper subset of V with $d(v)$, while $d^{\dagger}(v)$ indicates the (out-)degree of v as a vertex of X_I, the subgraph induced by X.

Finally, a quick note on the quantities involved. While the notation is not the most user-friendly, all the quantities involved can be easily calculated in linear time using the appropriate data structure for the graph (as they are just degrees) and so there is no major barrier to apply the method for large graphs.



$$P(A) = \frac{1}{d_{out}(v_A)} P(v_A) = \frac{1}{d_{out}(v_A)} \frac{d_{in}(v_A)}{\sum_{v \in X} d_{in}(v)} \propto \omega_A$$

$$P(A) = \sum_{v \in X} \frac{\mathbb{I}_E(A, v)}{d_{out}(v)} P(v) = \sum_{v \in X} \frac{\mathbb{I}_E(A, v)}{d_{out}(v)} \frac{d_{out}(v)}{\sum_{w \in X} d_{in}(w)}$$
$$= \frac{1}{\sum_{w \in X} d_{in}(w)} \sum_{v \in X} \frac{d_{in}(v)}{d_{out}(v)} \mathbb{I}(A, v)$$

Figure 1: Graphical representation of the weight assignment rules for the output sets

3 Perspectives

Ergodic set are a simple yet powerful tool that allows to simplify the structure of directed networks using only the underlying structure of the network. They could be deployed in the analysis of real world network, and particularly food chains, which naturally presents clusters of species that are either apex predator or at the bottom of the food chain.

They also can further used to measure the directedness of networks, which can be defined from a random walker $(X_n)_{n \geq n}$. We can use the distribution of the hitting times of the outputs, conditioned on the walker starting in an input, to infer how "directed" the network is. This is the same idea as the hodge decomposition, which we plan to use as a reference tool to test the measure developed as described.

Finally, we can also use the construction in this work to form an insight on how much the network is mixing, by defining the *ergodic mixing matrix*. Take a random walker $(X_n)_{n \geq n}$ and define:

$$C_{ij} = \mathbb{P}\left(\lim_{t \to +\infty} X_t \in \mathcal{O}_j | X_0 \in \mathcal{I}_i \right) \tag{8}$$

in which $\{\mathcal{O}_1, \ldots, \mathcal{O}_n\}$ is the set of all output, and $\{\mathcal{I}_1, \ldots, \mathcal{I}_m\}$ the set of all outputs. This can be used to reduce the entire network core and obtain an extreme coarse-grained representation of the network, thus showing a remarkable flexibility of the construction, which can coarse-grain network at different levels.

References

[1] Newman M., *Networks*,Oxford university press (2018).

[2] Brin S., Page L. *The anatomy of a large-scale hypertextual web search engine* Computer networks and ISDN systems, (1998)

[3] Tarjan R., *Depth-first search and linear graph algorithms* SIAM Journal of Computing, (1972)

[4] Nuutila E., Soisalon-Soinen E., *On finding the strongly connected components in a directed graph* Information Processing Letters, (1994)