Toward Random Walk Based Clustering of Variable-Order Networks

Julie Queiros, Célestin Coquidé, François Queyroi

Network Science (2022), 1–xx
doi:10.1017/xxxxx

CAMBRIDGE
UNIVERSITY PRESS

ARTICLE

Toward Random Walk Based Clustering of Variable-Order Networks

Julie Queiros, Célestin Coquidé and François Queyroi

Nantes Université, Ecole Centrale Nantes, CNRS, LS2N, UMR 6004
F-44000 Nantes, France
* Email: firstname.surname@univ-nantes.fr
Action editor: Prof. Ulrik Brandes

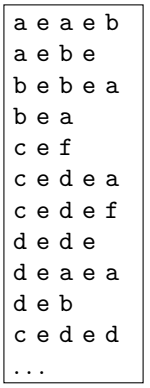
Abstract
Higher-order networks aim at improving the classical network representation of trajectories data as memory-less order 1 Markov models. To do so, locations are associated to different representations or "memory-nodes" representing indirect dependencies between visited places as direct relations. One promising area of investigation in this context are variable-order network models as it was suggested by Xu *et al.* (2016) that random walk based mining tools can be directly applied on such networks. In this paper, we focus on clustering algorithms and show that doing so leads to biases due to the number of nodes representing each location. To address them, we introduce a representation aggregation algorithm that produces smaller yet still accurate network models of the input sequences. We empirically compare the clustering found with multiple network representations of real world mobility datasets. As our model is limited to a maximum order of 2, we discuss further generalizations of our method to higher orders.

Keywords: Networks analysis, Higher-order networks, Clustering, Random walk, Sequential data.

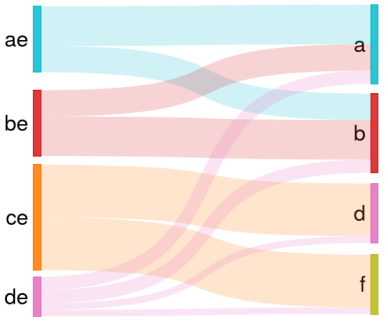
1. Introduction

Networks are powerful tools to represent relations between entities. They can, for example, be used in the context of trajectory and mobility analysis to encode the amount of movements between different locations. These movements can correspond to sequences of locations (Fig. 1a) tracking the order in which a traveler (ship, car, *etc.*) visited these places. Classic approaches only look at pairwise interactions obtained from input sequences, this results in a first-order network (Fig. 1c) where weighted edges represent transition probabilities between locations. The corresponding model is a memory-less first-order Markov model where the probability for a random walker to visit a given place depends only on its current location.

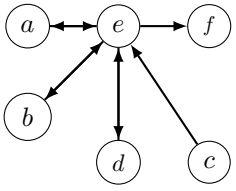
Recent works (Xu *et al.*, 2016; Scholtes, 2017) suggest that this representation cannot capture the spatial dependencies existing in the input sequences (Fig. 1b). Indeed, they show that random walks on first-order networks are often poor approximations of the behavior of travelers. "Higher-order" networks have been proposed in order to improve the network representation and go beyond the classic dyadic relation (Eliassi-Rad *et al.*, 2021). In this paper, we use this term to refer specifically to network models designed to represent indirect dependencies in sequential data. In such a model, different representations of locations (sometimes called "memory-nodes") are used as nodes to encode these dependencies. In this context, most of the studies focused on a first type of higher-order networks: the fixed-order networks of order k (FON_k), where the probability to visit a location depends on the last k visited places (Rosvall *et al.*, 2014; Scholtes, 2017). Other authors (Xu *et al.*, 2016; Saebi *et al.*, 2020a) proposed the use of a second type



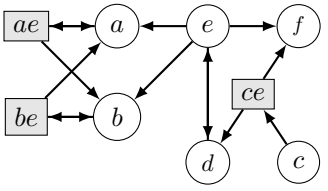
(a) Input



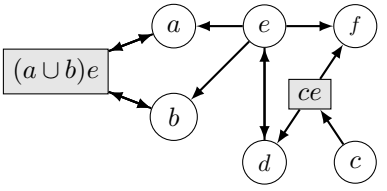
(b) Distributions P_{xe} with $x \in \{a, b, c, d\}$



(c) 1st-Order (FON_1) network



(d) VON_2 network



(e) Aggregated VON_2 network

Figure 1: Example of inputs and variable-order networks representing dependencies when visiting locations $\mathcal{A} = \{a, b, c, d, e, f\}$. When looking at successive pairwise interaction of set $\mathcal{S}$ in **(a)** a natural representation is the first-order network **(c)**. However, we can identify second-order dependencies in **(b)**. For instance, when coming from a the flow tends to either return to a or go to b after visiting e.
The VON_2 network in **(d)** only includes relevant dependencies. Sub sequence de is not relevant extension of the sequence e since knowing we visited d before e does not impact the prediction of next visited location. Second-order nodes are displayed as grey rectangles. The set $\mathcal{V}(e) = \{ae, be, ce, e\}$ of nodes are *representations* of the location e.
In our aggregated model **(e)**, multiple representations can be merged if they correspond to similar distributions. Here, the probability distribution P_{ae} is close enough to P_{be} to justify using this smaller model. Node $(a \cup b)e$ encodes the event "the traveler came either from a or b before visiting e".

of higher-order networks: variable-order networks (VON) that only include statistically relevant dependencies (Fig. 1d).

The motivation behind the design of these models is to improve the relevance of random-walk based mining methods such as PageRank ranking (Brin and Page, 1998) and Infomap clustering (Rosvall *et al.*, 2009). For example, the Infomap algorithm was applied to VON built from a maritime transportation dataset (Xu *et al.*, 2016). Since ports may have many representations as nodes in the network, the partition of nodes found actually corresponds to overlapping clustering of the ports. A possible application is the prediction of invasion of an ecosystem by non-indigenous aquatic species (Saebi *et al.*, 2020b).

One important argument (Xu *et al.*, 2016; Saebi *et al.*, 2020a) for the use of VON was that algorithms such as Infomap can be used directly on VON since these are defined as a weighted

graph. This paper aims to investigate this claim with a focus on clustering algorithms. Several higher-order networks can be built from the same input dataset, for instance, we can imagine smaller VON models that can also accurately predict travelers' movements (see Fig. 1e). Even though they contain similar features, we show that, even if trajectories are modeled using a second-order Markov process, the difference in the number of representations of locations has important effects on the results. Our results support the idea that, even if higher-order dynamics can be encoded in a classic weighted graph, mining tools designed for memory-less networks may not be adapted to capture these dynamics. Rather, future network algorithms should be specifically designed to take the multiplicity of higher-order network representations into account.

The contributions of this paper are:

- Analysis of the impact of multiple representations of locations on the Infomap algorithm (Rosvall *et al.*, 2009).
- New model of aggregated VON (limited to a maximum order of 2) that produces a more parsimonious representation of the input sequences.

The paper is organized as follows. We discuss the semantic of "higher-order networks" and works related to their clustering in Section 2. Notations and the construction of VON, FON and Agg-VON models are given in Section 3, then three potential effects of the existence of memory-nodes in the context of the Infomap clustering algorithm are presented. In Section 4, we show the effect of aggregation on synthetic benchmark networks, and in the context of real sequences datasets we show that our model is still accurate and that there are important differences in the clusterings found with it. As our model is limited to a maximum order of 2, the generalization of our aggregation procedure to any order as well as other possible adaptations of network analysis tools are discussed in Section 5.

2. Related Works

The terms "high-order" or "higher-order" might be confusing as they are used to describe different concepts in data mining and network analysis. Eliassi-Rad *et al.* (2021) broadly define "higher-order networks" as generalizations of graph representations designed to *"capture more that dyadic interactions"*. Examples include subset dependencies found in co-authorship data, spatial dependencies found in transportation's networks or indirect dependencies found in sequential data. The first example corresponds to "higher-order interactions" as discussed in Battiston *et al.* (2020). A good survey of these different representations can be found in Torres *et al.* (2021). Following the terminology used in this survey, we refer here to "higher-order" as models of indirect dependencies extracted from sequential data.

Indeed, this term directly refers to higher-order Markov models. They are used in sequence prediction for compression algorithms (Begleiter *et al.*, 2004) or sequences classification (Ching *et al.*, 2004; Chen *et al.*, 2021). For the latter, the distance between a sequence and a cluster of sequences is defined as the likelihood of this sequence in a model built from the sequences in the cluster. The number of previous steps used can be fixed to a given k or inferred statistically. We will refer to this subset of higher-order models as *fixed-order* models. An important drawback is that the number of parameters of the models grows exponentially with k. Chen *et al.* (2021) use a more general form of models called *variable-order* Markov models. In these models, a set of contexts of various lengths is used instead (Ron *et al.*, 1994).

Accordingly, we shall refer in this paper to fixed-order networks of order k and variable-order networks denoted as FON_k and VON respectively. Both are special cases of higher-order networks

and include "memory nodes" in order to encode the transition probabilities of the corresponding higher-order Markov models. As the underlying model is a better predictor of sequences, a random walk performed on these networks should better represent the system that produces those sequences. In this context, most studies were dedicated to the class of FON$_k$. Finding the k that leads to the best representation was addressed in Scholtes (2017). In a recent perspective article, Lambiotte *et al.* (2019) highlight the importance of model selection and the need for alternatives such as variable-order networks.

A variable-order network model was introduced in Xu *et al.* (2016), although the authors use the broader term "higher-order networks" or HON to name their model. The main idea is to only keep memory nodes that actually add information about a random walker behavior. The authors implement this idea in an algorithm that recursively searches for contexts that correspond to significantly different transition probabilities when compared to shorter contexts already found (we precisely described the procedure in the following section). The implementation of this algorithm was further improved in Krieg *et al.* (2020). The concept of significance of "contexts" is also used in sequence classification although significance is here defined as the number of times a given context is found in the input data (Chen *et al.*, 2021). In this case, even a small difference in transition probabilities can be deemed important. Saebi *et al.* (2020a) then further developed the model of Xu *et al.* (2016) by introducing a threshold function to assess the significance of the difference between transition probabilities making the generation of VON parameter-free.

Clustering of FON$_2$ networks was discussed in Rosvall *et al.* (2014). The authors introduced a generalization of the Infomap algorithm (Rosvall *et al.*, 2009) in order to find overlapping clustering of the locations. The Infomap algorithm is also used by Xu *et al.* (2016) however the algorithm was, in this case, directly applied on the VON representations. In this paper, we investigate the choice of higher-order model selection on the clustering results of the Infomap algorithm. In particular, we investigate the effect of using sparser model achieved by merging memory nodes with similar transition probabilities. A similar idea was previously suggested in Jääskinen *et al.* (2014) resulting in a model called *Sparse Markov Chains*. The goal of the authors was to improve the rate of compression and the classification of DNA sequences and protein data. However, their contribution is based on mixed-order models (a combination of fixed-order model of order lesser or equal to k).

In the context of VON, the effect of multiple representations on PageRank values was investigated in Coquidé *et al.* (2022). In this context, the PageRank of locations was defined as the sum of the PageRank values of its representations. It was shown that the non-uniformity of the number of representations per locations leads to a bias.

3. Definitions and Methods

We describe here the notations used in this paper and the construction of variable-order networks. The application of the Infomap clustering algorithm to VON is then discussed. We highlight the influence of the number of location per location. A more parsimonious network (Agg-VON$_2$) is then introduced.

3.1 Variable-Order Network (VON) Representation

Let $\mathcal{A}$ be the set of *locations* (itemset). An input dataset corresponds to a set $\mathcal{S} = \{s_1, s_2, \ldots\}$ of sequences $s_i = \sigma_1\sigma_2\sigma_3\ldots$ where all $\sigma_j \in \mathcal{A}$. For a sequence s of symbols in $\mathcal{A}$, the order of s denoted $|s|$ is the length of s and the count (or flow) of s denoted $c(s)$ is the number of occurrences of s in dataset $\mathcal{S}$. We will also use $C_s = (c(s\sigma_1), c(s\sigma_1), \ldots, c(s\sigma_m))$ to denote the occurrences

of every possible location σ_i following the sequence s. Let $s = s_1 s_2$ be a sequence resulting in the concatenation of sequences s_1 and s_2. We say that s_2 is a *suffix* of s and s_1 is a prefix of s. The interactions and dependencies within the system that produced the sample $\mathcal{S}$ may loosely be called the *flow dynamic*. In this context, we are interested in inferring the possible locations visited by a traveler after a given set of locations.

Definition 1. ***Transition probability.*** *For a sequence s, the transition probability from s (context) to $\sigma \in \mathcal{A}$ (target) is defined as*

$$p(\sigma \mid s) = \frac{c(s\sigma)}{\sum_{\sigma' \in \mathcal{A}} c(s\sigma')} \tag{1}$$

where $s\sigma$ is the concatenation of symbols in s followed by σ. Also, the probability distribution of locations σ_i visited after context s is denoted as $P_s = (p(\sigma_1 \mid s), p(\sigma_2 \mid s), \ldots, p(\sigma_m \mid s))$.

The probability in Eq. 1 is the maximum likelihood estimate given a sample $\mathcal{S}$. However, we will not use special notation to distinguish it from an unknown true distribution. The main difference between the compared models here is going to depend on the set of contexts s for which transition probabilities are defined.
Fixed-order models usually rely on taking all possible contexts of order k. Obviously, the size of the model is $O(|\mathcal{A}|^k)$. The variable-order network model studied here aims at finding a subset of relevant contexts. They can be expressed as *extension* of each other.

Definition 2. ***Relevant extension (Saebi** et al. **(2020a)).*** *For a sequence s' and s one suffix of s', we say that s' is a relevant extension of s if*

$$D_{KL}\left(P_{s'} || P_s\right) > \frac{\alpha |s'|}{\log_2(1 + c(s'))} \tag{2}$$

where D_{KL} is the Kullback-Leibler (KL) divergence (in bits) and $\alpha \geq 1$ is the threshold multiplier.

In Fig. 1d, the knowledge that we came from a before visiting e is relevant as we can better predict the next visited location. Therefore, ae is a relevant extension of e. However, de is not since knowing that we came from d does not add significantly more information (as expressed by D_{KL}) that we already had. In general, it is possible to have a 3rd or higher-order extension $s = \sigma_1\sigma_2\sigma_3 \ldots$ identified as relevant while some of the suffixes of s are not. We can see from the right part of Eq. 2 that it is increasingly harder for high order and sparsely observed subsequences to be relevant. Saebi *et al.* (2020a) use a value of $\alpha = 1$. We can use a higher α value to construct more parsimonious networks. In the experiments, we use this parameter to evaluate the accuracy of the aggregated model described in section 5.1.

The variable-order network constructed with dataset $\mathcal{S}$ using the method of Saebi *et al.* (2020a) with threshold multiplier α is denoted $\textsc{Von}(\alpha) = (\mathcal{V}, \mathcal{E}, w)$. In the rest of the paper, we will just call it VON when we assume $\alpha = 1$. The general idea is to build a graph where each location $\sigma \in \mathcal{A}$ is represented by multiple nodes corresponding to its extensions. We say that these nodes are the *representations* of σ. We use the sequences as node labels and we refer in this case to the terms "nodes" and "sequences" interchangeably.

The construction of VON is done in two phases. First, we extract the set of relevant extensions. Starting from the set $\mathcal{A}$, the relevant extensions are found using a recursive procedure (order 1 sequences $\{\sigma \in \mathcal{A}\}$ are always considered as relevant). An upper-bound of D_{KL} can be used to stop the recursion and the algorithm does not require a maximum order parameter to stop the search. The set of nodes $\mathcal{V}$ will include all detected relevant extensions. Also for each extension

$s = \sigma_1\sigma_2\sigma_3 \ldots \in \mathcal{V}$, all of its prefixes are added to $\mathcal{V}$ even if they are not relevant extensions in order to guarantee that the node s is reachable (see Prop. 1 below).
Second, the edge set $\mathcal{E}$ and the weights w are defined as follows. Let $s \in \mathcal{V}$ and $\sigma \in \mathcal{A}$ such that $p(\sigma|s) > 0$, VON contains a link $s \to s^*\sigma$ of weight $w(s \to s^*\sigma) = p(\sigma|s)$ where

$$s^* = \arg \max_{s'\sigma \in \mathcal{V}} \{|s'|, s' \text{ is a suffix of } s\} \tag{3}$$

For example, let $s = abc$ and $s^*\sigma = bc\sigma$ be relevant extensions of c and σ respectively then there will be a link $s \to s^*\sigma$ if $abc\sigma$ is not a relevant extension of $bc\sigma$ and $p(\sigma|s) > 0$. In Fig. 1d, links $x \to e$ in the first-order network are replaced by links $x \to xe$ if $x \in \{a, b, c\}$. Note that this definition of the edge set is a shorter reformulation of the idea given by Saebi *et al.* (2020a) which only provides a procedure to construct the set $\mathcal{E}$ involving edge rewiring.

Property 1. ***Random walk as variable-order Markov model simulation*** *Let* VON$= (\mathcal{V}, \mathcal{E}, w)$ *and* $s = \sigma_1\sigma_2 \ldots \sigma_m \in \mathcal{V}$ *a representation of* σ_m*, there exists a path* $\sigma_1 \to \sigma_1\sigma_2 \to \ldots \to s$ *followed by a random walker starting in* σ_1 *with probability* $\prod_{i=2}^{m} p(\sigma_i|\sigma_1 \ldots \sigma_{i-1}) > 0$.

Proof. By definition, each $(\sigma_1, \sigma_1\sigma_2, \ldots, s)$ is a labeled node of $\mathcal{V}$ as prefix of the relevant extension s. Let s'_i be the prefix of order $i < m$ of s. Since $s'_i\sigma_{i+1}$ is also a prefix of s, we have $c(s'_i\sigma_{i+1}) > 0$ so there is an edge $e = (s'_i \to s'_i\sigma_{i+1}) \in \mathcal{E}$ with $w(e) = p(\sigma_{i+1}|s'_i) > 0$ by definition since s'_i is the largest suffix of s'_i (Eq. 3). □

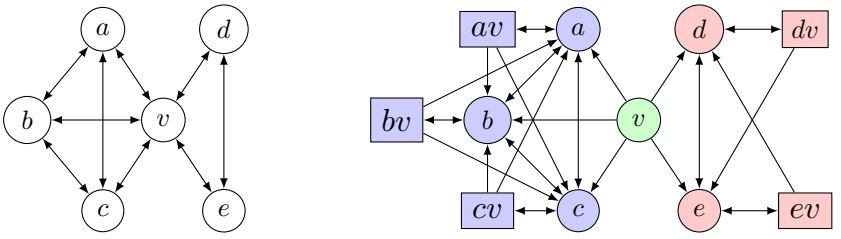
The above property shows that a random walk on VON corresponds to a simulation of the variable-order Markov model composed of the selected subsequences of $\mathcal{V}$. A random walker can follow the higher-order dependencies detected in the input dataset. When restricted to a maximal order of k (*i.e.* when stopping the recursive extraction of extension at order k), the returned network is called VON$_k$. For a higher-order network, the set of kth-order nodes is $\mathcal{V}_k \subseteq \mathcal{V}$. Moreover, the set of the k-order nodes corresponding to representations of a location $v \in \mathcal{A}$ is denoted $\mathcal{V}_k(v)$. Note that VON$_k$ can still be viewed (even with $k = 2$) as a variable-order model since not all extensions of order k are kept as relevant.

The fixed-order network FON$_k = (\mathcal{V}_F, \mathcal{E}_F, w_F)$ is built by taking all subsequences of length k in $\mathcal{S}$ as the set of nodes $\mathcal{V}_F$ while $\mathcal{E}_F$ and w_F definition is similar to VON. As such, FON$_k$ corresponds to a subgraph of the De Bruijn graph of length k over alphabet $\mathcal{A}$.

3.2 Clustering of VON using Infomap

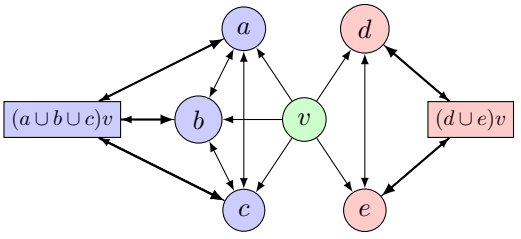
Node partitioning aims at identifying well connected subgraphs that are sparsely connected to the rest of the network. Several contributions in this domain gave different formal definitions to this idea in the form of quality measures to optimize. Dao *et al.* (2020) give a comprehensive review and comparison of the most commonly used strategies. The Map Equation (Rosvall *et al.*, 2009) denoted $L(\mathcal{C})$ can be viewed as a description length metric of the partition $\mathcal{C} = \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_m\}$. For a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$, the best partition of $\mathcal{V}$ corresponds to the best two-level encoding of random walks in $\mathcal{G}$.

Let π_v be the steady state probability that a random walk visits node $v \in \mathcal{V}$. In a one-level encoding, an optimal binary code for node v (called *codeword*) would be of length $\log_2(\pi_v)$ and the expected usage of the codeword per step of a random walk is π_v. The average number of bits encoding a random walk using a one-level encoding is therefore given by $H(\{\pi_v\}_{v\in\mathcal{V}}) = \sum_{v\in\mathcal{V}} \pi_v \log_2(\pi_v)$ *i.e.* the entropy of the visit probabilities for nodes in $\mathcal{G}$.
Using a two-level encoding, it is possible to achieve a lower code length. Indeed, the second level of encoding assigns enter and exit codewords to clusters. A random walk leaving cluster



(a) First-order network

(b) VON$_2$ Network



(c) Minimal VON$_2$ Network

Figure 2: 2 Cliques example. Assume a graph **(a)** made of two cliques $\{v, a, b, c\}$ and $\{v, d, e\}$. The sequences $\mathcal{S}$ are constructed as follows: a traveler picks any outgoing edge at random. However, if it reaches v, it will return to another node of the clique it came from. The network VON$_2(\mathcal{S})$ **(b)** then includes one representation of v for each other node and all edges leaving the same node have the same weight. In MIN-VON$_2(\mathcal{S})$ **(b)** the representations of v are merged according to the known flow dynamic and include only two representations of v. The color indicates the partition that should be found by the clustering algorithm. When projecting on the locations, v will be found in 3 clusters (1 is trivial).

$\mathcal{C}_1$ and entering cluster $\mathcal{C}_2$ would use the exit codeword of $\mathcal{C}_1$ and the enter codeword of $\mathcal{C}_2$. This allows for a codeword to be reused for nodes belonging to different clusters. If clusters are well separated, the increase in length due to enter/exit codewords will be compensated by the reuse of nodes codewords.

The algorithm called *Infomap* aims at minimizing the objective function L using a fast greedy multi-level procedure. Note that the algorithm does not actually find the best two-level encoding. Rather, the Map Equation $L(\mathcal{C})$ is defined as the theoretical minimum average number of bits *per* step of the random surfer.

The application of Infomap to higher-order networks is natural. Indeed, different representations can be clustered into different groups. This corresponds to the idea that representations of a same location encode different behaviours. Consider the example given in Fig. 2b, the different representations of location v perfectly capture the flow dynamic (*i.e.* a traveler can never leave a given clique). A clustering algorithm should return the node partition corresponding to the three colors. From this partition we can construct more general clustering by assigning to each location the set of clusters where at least one of its representations was found, resulting in an overlapping clustering of the locations (Lancichinetti *et al.*, 2008).

As said previously, it was suggested that we could directly apply Infomap algorithm on VON (Xu *et al.*, 2016) without any adaptation *i.e.* we can consider VON as a "simple" first-order network in this context. However, the multiplicity of representations may have important effects

on the algorithm. In the example above, another network can be built with fewer representations (see Fig. 2c). Notice it also splits vertices into two strongly connected components. We refer to this alternative higher-order model as *minimal* and call it MIN-VON$_2$. In the simple case of Fig. 2, one could expect the result of an algorithm to correspond to the same overlapping clustering. In the general case, we argue that the number of "additional" representations of VON when compared to an ideal *minimal* model has three potential effects discussed next.

Effect on the number of codewords per cluster.
When Infomap is applied to VON, two representations of the same location belonging to the same clusters are given different codewords. This may make the detection of clusters containing many representations of the same location harder. This issue has already been identified Rosvall *et al.* (2014) for FON$_2$ networks. They suggested a modification of Infomap in order to give the same codeword to representations of the same location. Indeed, the more a location is represented in a cluster the larger the contribution of this cluster to the Map Equation will be. For example, in the VON$_2$ network in Fig. 2b, we use three codewords for each of the nodes $\{av, bc, cv\}$. The code length achieved for the clustering would therefore be higher than the one achieved for the MIN-VON$_2$ network in Fig. 2c where there is only one representation of a location *per* cluster.

Effect on rate of codewords usage
Remember that steady state probabilities of a random walker is used to compute the rate at which codewords are used in the Map Equation. In order to guarantee the uniqueness of π's values, a random walk must be turned into a random surfer with a positive probability τ to teleport to any given node at each step. Infomap uses the usual $\tau = 0.15$. The π values therefore correspond to the PageRank metric (Brin and Page, 1998). The values associated to each node are computed at the beginning of the algorithm.
The teleportation mechanism creates discrepancies between the VON$_2$ and MIN-VON$_2$ models (Coquidé *et al.*, 2022) even assuming the first effect is corrected *i.e.* we use a similar codeword for representations of a location that belongs to the same clusters. In the example of Fig. 2, assuming we are correcting for the first effect, the probability for a random surfer to use the codeword associated to nodes (av, bc, cv) after a teleportation is $\frac{3}{10}$ for VON$_2$ and $\frac{1}{8}$ (less than half) for MIN-VON$_2$.
This second effect can therefore also make the identification of larger clusters difficult. Indeed, the more a cluster contains representations the more its enter, exit and nodes codewords are used. In the minimal model, the π rates are also biased due to the different number of representations but to a lesser degree. Even tough a random surfer is more likely to visit a representation of location belonging to different clusters, there is, by definition, only one representation per cluster.

Effect on the solution space exploration
The Infomap algorithm follows a hierarchical greedy procedure, starting with a partition where each representation is assigned to a single cluster. As such, regrouping the representations of a location in the same clusters requires as many modifications as the number of representations in each cluster. The chance of running into local minima is therefore higher.

We conclude that there are several reasons why we should try to compare clustering achieved by VON and those achieved on a more parsimonious network model. The fact that the Map Equation L is affected by the number of representations used does not means that the results of Infomap will be worse. However, the experiments detailed in Section 4.1 show that it is the case even when considering only second order dynamics.

3.3 Aggregated VON$_2$ Model

In the previous section, the minimal VON model is built from an underlying known clustering. In order to access the relevance of parsimonious models in real case studies, we define here our aggregated model of VON$_2$ called AGG-VON$_2$. The underlying hypothesis we use to get closer to a minimal model is that representations having similar output transition probabilities will belong to the same clusters. This is clear when looking at the example in Fig. 2. As for the previous case study, we assume that the flow dynamic is captured at a maximum order of 2. We will clarify this assumption in Section 5.1. We first define the concept of merging representations and then introduce the criteria that we use to obtain an aggregated model. This leads to the definition of the Algorithm 1.

Definition 3. ***Merged representation.*** *For $v \in \mathcal{A}$ a location, we call merged representation a subset $X \subseteq \mathcal{V}_2(v)$ and for $\sigma \in \mathcal{A}$ we define $c(X\sigma) = \sum_{x_1x_2 \in X} c(x\sigma)$ as the merged flows of X and*

$$p(\sigma|X) = \frac{c(X\sigma)}{\sum_{\sigma' \in \mathcal{A}} c(X\sigma')} \tag{4}$$

as the transition probabilities of X to σ. As with Def. 1, P_X (C_X) shall denote the probability (flow) distribution associated to symbols following any sequence in X.

This generalized form of transition probability can be interpreted as the probability of arriving at location σ given the fact that we are in location v having visited *one of* the locations x such that $xv \in X$.

Definition 4. ***Possible Merge.*** *Let X, Y be two disjoint merged representations of v. We say that (X, Y) can be merged or $X \boxplus Y$ if the following inequalities hold*

$$D_{KL}\left(P_X||P_{X\cup Y}\right) < \frac{2}{\log_2(c(X)+1)} \tag{5}$$

$$D_{KL}\left(P_Y||P_{X\cup Y}\right) < \frac{2}{\log_2(c(Y)+1)} \tag{6}$$

$$D_{KL}\left(P_{X\cup Y}||P_v\right) > \frac{2}{\log_2(c(X\cup Y)+1)} \tag{7}$$

An example of possible merge is given in Fig. 1e with the merging of representations ae and be into a single representation $(a \cup b)e$. The idea behind the above conditions is to reuse the criterion for relevant extension in Def. 2 with a threshold multiplier $\alpha = 1$. Indeed, we have $ae \boxplus be$ when knowing that the traveler came from a *or* b is relevant (Cond. 7) however the additional information "it actually came from a (or b)" is not (Cond. 5 and 6).

Property 2. ***Non-Transitivity.*** *Let X, Y, Z be disjoint merged representations of v then*

$$(X \boxplus Y) \wedge (Y \boxplus Z) \nRightarrow (X \boxplus (Y \cup Z))$$

Proof. Let $C_X = (n, 0, 0, \ldots, 0)$, $C_Y = (n, n, 0, \ldots, 0)$, $C_Z = (0, n, 0, \ldots, 0)$ and $C_v = (2n, 2n, N, 0, \ldots, 0)$ where $n \ll N$. Due to the last assumption, we can assume the condition Eq. 7 always holds. We have $D_{KL}(P_X, P_{X\cup Y\cup Z}) = 1$ so for any $n > 2$ we have $\neg\left(X \boxplus (Y \cup Z)\right)$. However, $D_{KL}(P_X, P_{X\cup Y}) = D_{KL}(P_Z, P_{Y\cup Z}) = \log_2(3) - 1$. Therefore, for $2 < n \leq 9$, $(X \boxplus Y) \wedge (Y \boxplus Z)$ is true while $(X \boxplus (Y \cup Z))$ is false. □

A consequence of Property 2 is that a minimum set of merged representations can not be found by iteratively performing all possible merges since the results may be arbitrary. We therefore use a hierarchical aggregation procedure that will prioritize the merges of representations that are the most similar (in terms of distance to their union distribution). However, since not all representations can be merged, the aggregation procedure does not need a stopping condition and the number of merged representations returned depends on the thresholds of Def. 4. This operation is therefore parameter-free. The drawback for it is similar to the one related to the building of VON networks: it relies on the definition of relevance according to Saebi *et al.* (2020a).

Require: $\mathcal{V}_2(v)$ (2nd-order representations of v)
Ensure: $\mathcal{R}$ (partition of $\mathcal{V}_2(v)$)
1: $\mathcal{R} \leftarrow \{X \in \mathcal{V}_2(v)\}$
2: $M \leftarrow \{(X, Y) \in \mathcal{R} \times \mathcal{R} : X \boxplus Y\}$
3: **while** $M \neq \emptyset$ **do**
4: $\quad (X, Y) \leftarrow argmin_{(X',Y') \in M} D_{KL}\left(P_{X'} || P_{Y' \cup Y'}\right) + D_{KL}\left(P_{Y'} || P_{X' \cup Y'}\right)$
5: $\quad \mathcal{R} \leftarrow \mathcal{R} \setminus X \setminus Y \cup (X \cup Y)$
6: $\quad M \leftarrow \{(X, Y) \in \mathcal{R} \times \mathcal{R}, X \boxplus Y\}$
7: **end while**
8: **return** $\mathcal{R}$

Algorithm 1: Merge second-order representations of v

The outline of the procedure can be found in Algorithm 1. For simplicity sake, it corresponds to a direct approach of the problem. This problem is similar to a hierarchical clustering (Manning *et al.*, 2008) with two main differences. First, testing $x \boxplus y$ and computing the similarity between x and y is done in $O(|\mathcal{A}|)$. Assuming $N = |\mathcal{V}_2(v)|$ and given that $N \leq |\mathcal{A}|$, the time complexity of Alg. 1 is therefore $O(N^2|\mathcal{A}|)$. Second, as not all merges are possible, the set M may be sparse and requires $O(N^2)$ space. The procedure space complexity therefore corresponds to the $O(N|\mathcal{A}|)$ required to store values of c. Computation times for real datasets are discussed in Section 4.

The third condition in Eq. 7 guarantees that merged representations are still relevant extensions of first-order sequences. Having identified all merged representations of each location using Alg. 1, we construct AGG-VON$_2$ by merging the second-order nodes of VON$_2$ that belong to the same group as shown in Fig. 1e. Merged nodes transition probabilities are defined using Eq. 4. The node fusion preserves Prop. 1 albeit a random walker will use the latter estimation of transition probabilities. Indeed, for a merged representation X of v and $\sigma \in \mathcal{A}$, the longest suffix s^* in Eq. 3 is similar for every $xv \in X$ as $s^*\sigma$ is either σ or $v\sigma$. Moreover, for $xv \in X$ and $yv \in X$ there is no $s \in \mathcal{V}$ such that $(s \rightarrow xv) \in \mathcal{E}$ and $(s \rightarrow yv) \in \mathcal{E}$ since it would mean that s is a representation of both x and y.

There are several tests we have to perform in other to assess the relevance of AGG-VON$_2$. First, the produced aggregated network should be significantly smaller in terms of number of nodes. Second, it should represent flow dynamics almost as well as the VON$_2$ network. This condition is crucial since the point of using random walk based clustering on higher-order networks was to take advantage of their capacity to reproduce observed sequences. Third, if the first hypothesis is verified, we expect the clusterings found on the aggregated network to be different that the one obtained on the VON$_2$ network for the reasons described in the previous section.

Table 2. : Difference in number of representations in the LFR benchmark for networks VON$_2$ and MIN-VON$_2$

Overlap / Mixing	Clust Size Range	Median Nb. Nodes (VON$_2$ / MIN-VON$_2$ / Diff)
15% / 15%	20 - 50	2248 / 1295 / - 953
	50 - 100	2113 / 1297 / - 816
15% / 30%	20 - 50	1944 / 1293 / - 651
	50 - 100	1916 / 1292 / - 624
30% / 15%	20 - 50	2928 / 1575 / - 1353
	50 - 100	2886 / 1574 / - 1312
30% / 30%	20 - 50	2579 / 1592 / - 1017
	50 - 100	2501 / 1555 / - 946

Reading: (third line) There is a median of 1944 nodes in the VON$_2$ network for parameters Overlap, Mixing and Clust Range of $15\%, 30\%$ and $20-50$ respectively. There is however 1293 nodes in the MIN-VON$_2$ network (a difference of -651). This line correspond to the first column in the top-right panel of Fig. 3.

4. Experiments and Results

We report in this section the experiments made to test the various hypothesis made in previous section. First, in Section 4.1, we demonstrate on synthetic datasets that a VON with a minimal number of representation is more efficient for the identification for known clusters. Next, in Section 4.2, we show that the AGG-VON$_2$ model produces a smaller and accurate model. We also discuss the differences obtained with the Infomap clustering on three real-world sequences dataset[1].

4.1 Effect of aggregation on synthetic benchmarks

We focus on the impact of the number of codewords per cluster on clusters extracted with Infomap. Considering synthetic networks, we measure the efficiency of reducing the number of representations in the clustering.

These test experiments are made using graphs with more complex clusters and node degree distributions than the ones in Fig. 2a. We use the LFR benchmark (Lancichinetti *et al.*, 2008) to generate directed graphs along with a ground-truth overlapping clustering of the nodes. In this benchmark, the "Mixing" parameter corresponds to the percentage of outgoing edges that are inter-clusters edges. The "Overlap" is the percentage of nodes in more than one cluster. Clusters size range gives the minimum and maximum sizes of clusters. The rest of the parameters take the following values found in the literature (Xie *et al.*, 2013): $N = 1000$ (number of nodes), $om = 2$ number of different clusters that overlapping nodes are members of, $\bar{k} = 10$ (average degree), $\max(k) = 50$ (maximum degree), $\tau_1 = 2$ (exponent of degree distribution), $\tau_2 = 1$ (exponent for clusters size distribution).

We use a flow dynamic similar to the example of Fig. 2: if a traveler coming from a cluster $\mathcal{C}_i$ arrives at an overlapping node v in $\mathcal{C}_i$, it will return to a random non-overlapping neighbor of v in $\mathcal{C}_i$. Otherwise, it will follow an outgoing edge at random. In this situation, the traveler is able to move from a cluster to another by following inter-cluster edges. Knowing the graph, the ground truth clustering and the flow dynamic we can directly construct the networks and try to recover

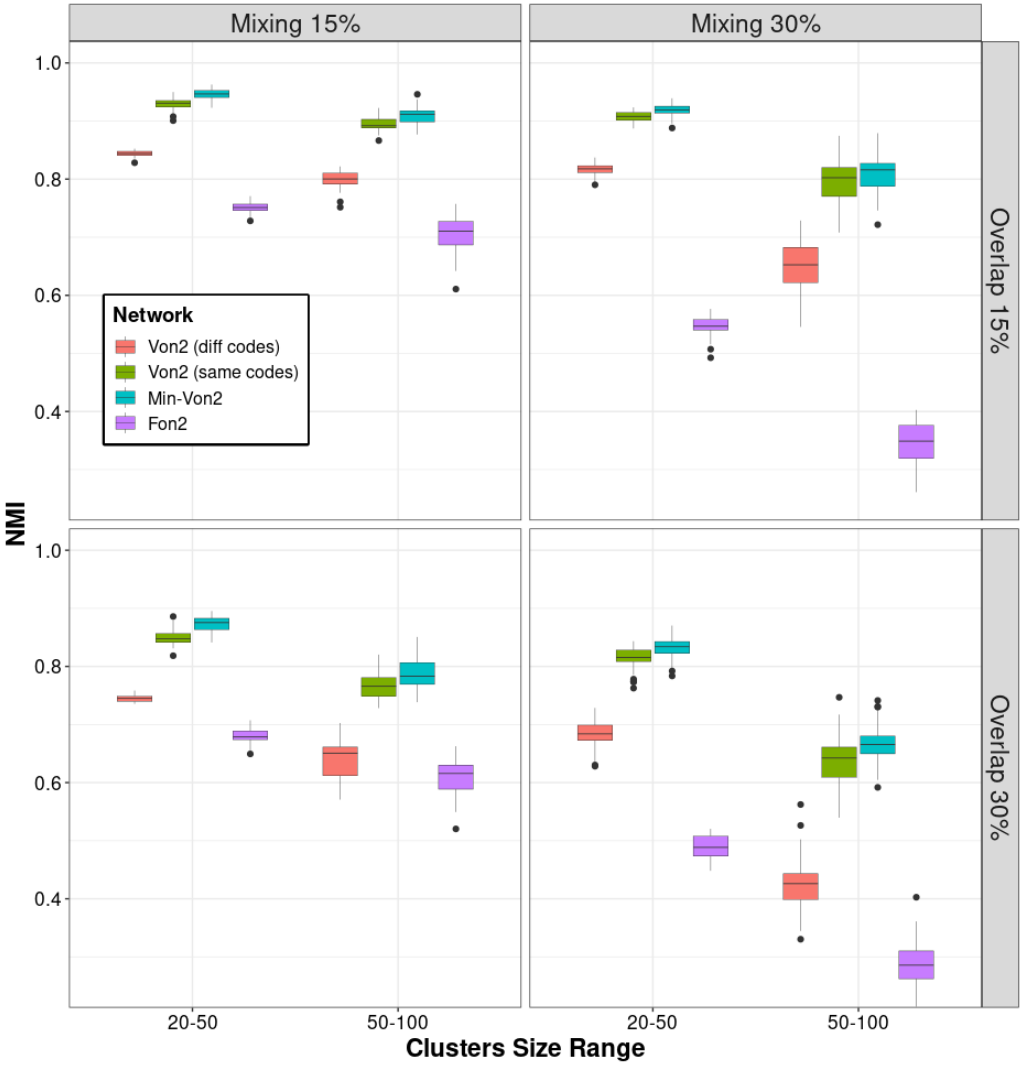


Figure 3: NMI between Infomap clustering found for each test case and the ground truth clustering as generated by the LFR benchmark. A value of 1 indicates a perfect identification of the real clusters (McDaid *et al.*, 2013). Each boxplot corresponds to the distribution over 50 tests.

the original clustering using Infomap. To test the impact of the effects described at the end of Section 3.3, we compare four different Infomap inputs:

- Network VON$_2$ when not correcting for the first effect described in Section 3.2 *i.e.* using different code words for all representations of locations
- Network VON$_2$ when correcting for the first effect *i.e.* using the same code word for representations of a location that belong to the same clusters
- Network MIN-VON$_2$ where representations are merged according to the clusters they should belong to (as in Fig. 2c). We also correct for the first effect
- Network FON$_2$ (Rosvall *et al.*, 2014) correcting for the first effect

The number of nodes found for VON_2 and MIN-VON_2 networks are reported in Tab. 2 for the various parameters used. Since the number of communities of overlapping locations is 2, each overlapping location should have 3 representations in MIN-VON_2 *e.g.* taking an overlap of 15% and 1000 locations (first four lines in Tab. 2), MIN-VON_2 should contain 1300 nodes. The discrepancies with the reported results come from the LFR algorithm that may need to relax some constraints *i.e.* the number of overlapping nodes may vary. Notice that the number of nodes in VON_2 is, however, lower with higher mixing values. Indeed, the inter-cluster neighbors of an overlapping location do not generate additional memory nodes.

The difference between clusterings found and the ground-truth are reported in Fig. 3. The distributions of NMI (McDaid *et al.*, 2013) values suggest that correcting for the first effect is important as it greatly improves the detection of the real clustering in all situations. The improvement when using the MIN-VON_2 is not as large. We can however notice an important gap when the difference in number of representations between VON_2 and MIN-VON_2 is the largest (15% mixing ratio and 30% of overlapping nodes). In this situation, it seems that too many clusters are identified overall. It also appears that the variable-order networks always perform better than the FON_2 networks. This shows that correcting for the first effect as suggested in Rosvall *et al.* (2014) is not enough.

4.2 Comparison on spatial trajectories datasets

Even if the last case study is revealing, the flow dynamic used is rather simple and the LFR benchmark may not reflect real world systems where the observed sequences occur. Moreover, the networks constructed here correspond to ideal *scenarii* where the transition probabilities are not estimated and the relevant extensions are not extracted from a set of sequences. It is indeed possible that the choice of model does not impact the clusterings of location found for real datasets. It is therefore important to compare results of clustering on real datasets. In this context, we use AGG-VON_2 model instead of MIN-VON_2.

We first show that our aggregated model is more parsimonious and maintains a good representative power. Moreover, such accuracy is not achieved when building smaller $\text{VON}_2(\alpha)$ with different threshold values. We then show that spatial trajectories can lead to relatively similar Infomap clusterings in one case (US flight itineraries). For the two other cases (Taxi itineraries and Shipping records), the clusterings found contain noticeable variations. When using the model AGG-VON_2, clustering tends to contain less clusters and overlapping locations.

4.2.1 Experimental settings

The three datasets correspond to trajectories or movements observation. Spatial sequences are indeed a major source of applications for higher-order networks analysis. The datasets studied are however different in terms of nature, number of locations, number of the sequences, *etc.*. Moreover, they were all used previously as applications of higher-order network mining. For each one, we removed consecutive repetitions of locations in each sequence.

- *Airports dataset* (Rosvall *et al.*, 2014; Scholtes, 2017): US flight itineraries extracted from the *RITA TransStat* 2014 database[2]. A sequence corresponds to passenger itineraries (as sequences of airports stops) that took place during the first trimester of 2011. We have $|\mathcal{A}| = 446$ and $|\mathcal{S}| = 2751K$.

- *Maritime dataset* (Xu *et al.*, 2016): Sequences of ports visited by shipping vessels extracted from the Lloyd's Maritime Intelligence Unit. The sample corresponds to observations that took place between April 1st and July 31th 2009. We have $|\mathcal{A}| = 909$ and $|\mathcal{S}| = 4K$.
- *Taxis dataset* (Saebi *et al.*, 2020a): Sequences of neighborhoods (represented by the closest police station) visited by taxis in Porto city between Jul. 1, 2013 and Jun. 30, 2014. The original dataset, consisting in GPS trajectories, was part of the ECML/PKDD challenge 2015[3]. The trajectories are mapped to neighborhoods as described by Saebi *et al.* (2020a). We have $|\mathcal{A}| = 41$ and $|\mathcal{S}| = 1514K$.

For each constructed network, we report the number of nodes (total number of representations) and the number of representations by location $N_{\mathcal{V}} : \mathcal{A} \to \mathbb{N}^+$.
To evaluate the networks ability to model the flow dynamics, relevant extensions are extracted using a training set of 90% of the sequences. For each constructed network, an accuracy score Acc (Eq. 8) is computed on the remaining sequences $\mathcal{S}^{test}$. The score corresponds to the average probability to correctly identify the next location starting with the third entry in each sequence given the two previous entries.

$$Acc(p) = \frac{1}{|\mathcal{S}^{test}|} \sum_{S \in \mathcal{S}^{test}} \frac{1}{|S| - 2} \sum_{i=3}^{|S|} p(s_i | s_{i-2} s_{i-1}) \tag{8}$$

Since, VON networks are variable order models, if a context $s = s_0 s_1 \in \mathcal{A} \times \mathcal{A}$ is not a relevant extension, then the probability $p(\sigma|s_0 s_1)$ will be estimated using $p(\sigma|s_1)$. Moreover, for the aggregated VON_2 model, $p(\sigma|s_0 s_1) = p(\sigma|X)$ where X is the merged representation $s_0 s_1$ belongs to. As explained in Section 5.1, we can use a value of $\alpha > 1$ in Eq. 2 in order to construct more parsimonious VON. By doing so, we simply make harder for contexts to be qualified as relevant. For each dataset we find the parameter α^* such that the total number of representations in $\text{VON}_2(\alpha^*)$ is as close as possible to AGG-VON_2's. We therefore compare four networks, for each dataset: $\text{VON}_2(1)$, $\text{VON}_2(\alpha^*)$, AGG-VON_2 and FON_2. $\text{VON}_2(\alpha^*)$ networks are used to compare the changes in representative power of AGG-VON_2. For each dataset and each model, the test was done 50 times and we report the mean Acc value and the standard deviation $sd(Acc)$.

Since Infomap is a non-deterministic algorithm, we applied it 50 times on each network and keep the clustering $\mathcal{C}$ associated to the smallest code length. We report the number of clusters $N_{\mathcal{C}} : \mathcal{A} \to \mathbb{N}^+$ for each location. We also report the decrease in code length ΔL when compared with the absence of clustering. A ΔL value close to 0 would suggest that the clustering is not a good summary of the flow dynamics. Note this measure (as well as the absolute values of L) cannot be used to directly compare the network models between them since the code length will mechanically be higher with the number of nodes, which can vary. We however report the NMI (McDaid *et al.*, 2013) between the clusterings found.

We used the Infomap python library developed by the authors[4]. Experiments were done using Intel i7-8650U processors at 1.90GHz with 30Gio of RAM, running Ubuntu 18.04 and Python 3.6.9.

4.2.2 Results discussion

We start by addressing the first two questions at the end of Section 5.1: is the aggregated network more parsimonious and, if so, does it represent flow dynamics well enough? Datasets and networks statistics are given in the top part of Tab. 3. Cumulative distributions of $N_{\mathcal{V}}$ values are shown in the left panels of Fig. 4.
We can first notice that the total number of nodes in the aggregated VON_2 is significantly smaller

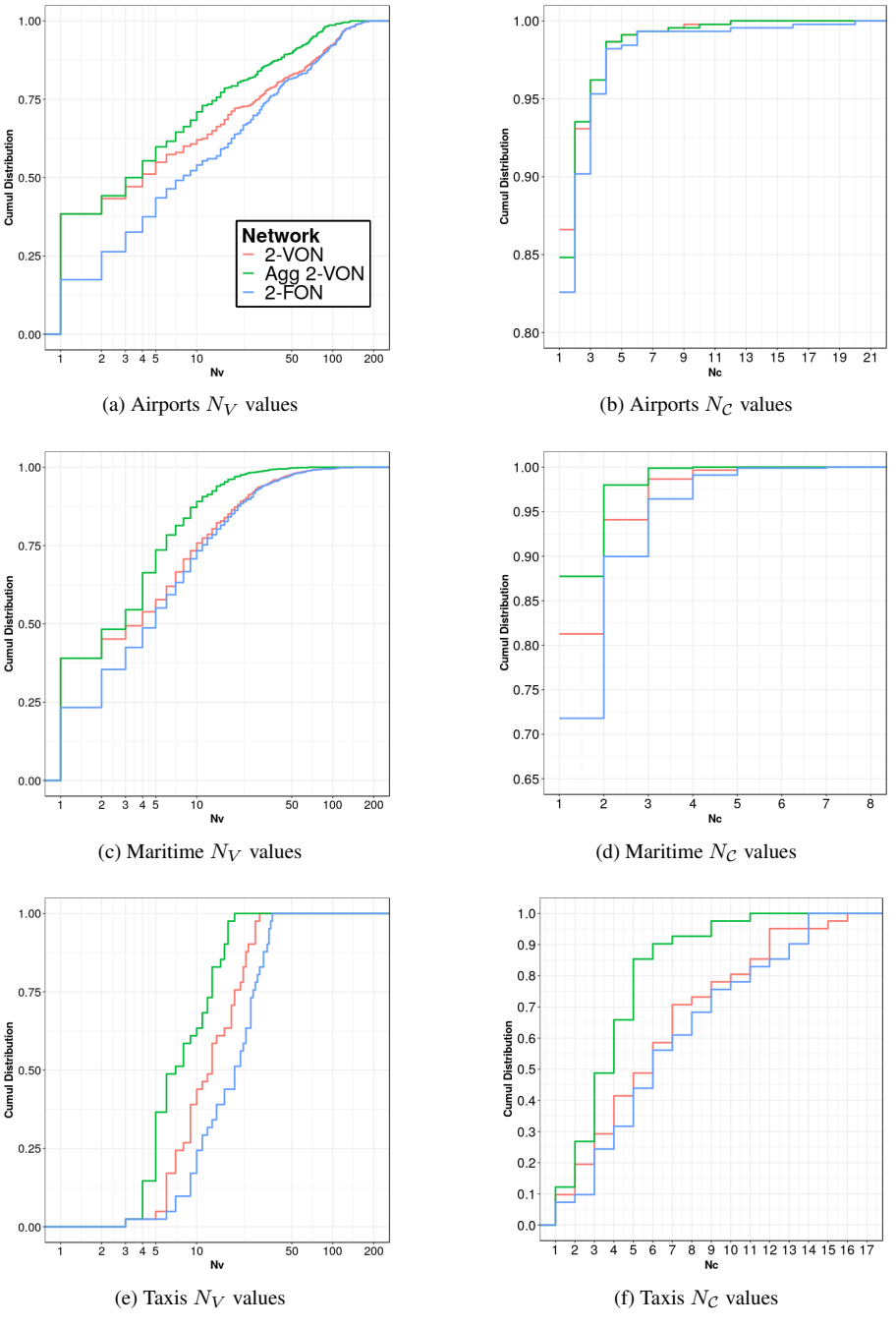
Table 3. : Network's model accuracy comparison

Dataset	**Network**	**Time Const.**	$\lvert\mathcal{V}\rvert$	**avg** $N_{\mathcal{V}}$	**max** $N_{\mathcal{V}}$	**Acc** $\pm 2sd$
Airports	VON$_2(1)$	24.07s	10456	23.34	183	$19.8\% \pm 0.07$
	VON$_2(3.6)$	15.52s	6440	13.4	149	$19.2\% \pm 0.68$
	AGG-VON$_2$	81.02s	6404	14.30	135	$19.5\% \pm 0.07$
	FON$_2$	16.98s	12036	26.87	183	$19.8\% \pm 0.07$
Maritime	VON$_2(1)$	0.58s	8005	8.85	136	$33.1\% \pm 1.20$
	VON$_2(2.8)$	0.29 s	4534	4.22	115	$29.8\% \pm 1.40$
	AGG-VON$_2$	3.33s	4397	4.86	66	$32.0\% \pm 1.12$
	FON$_2$	0.41s	8755	9.68	141	$33.0\% \pm 1.33$
Taxis	VON$_2(1)$	12.63s	574	14.00	29	$39.4\% \pm 0.12$
	VON$_2(4.4)$	7.65s	364	7.65	22	$33.0\% \pm 0.84$
	AGG-VON$_2$	12.76s	363	8.85	19	$39.3\% \pm 0.12$
	FON$_2$	5.95s	798	19.46	36	$39.8\% \pm 0.10$

than with the VON$_2$ or FON$_2$ networks. In the top row of Fig. 4, we can see that, not surprisingly, the drop in $N_{\mathcal{V}}$ values mostly impacts highly represented locations. For the taxis dataset (Fig. 4e), $N_{\mathcal{V}}$ values are more uniformly distributed. Next we can see that VON$_2$ and FON$_2$ models have close accuracy scores. These results are consistent with those of Xu *et al.* (2016) albeit the authors used a different definition of accuracy. Lower accuracy values are observed with the AGG-VON$_2$ model. The loss seems however negligible when compared to the differences in N_V values. Moreover, accuracy results are significantly worse for VON$_2(\alpha^*)$, even if the difference is not necessarily large, as it is with the Airport dataset for instance. We can conclude that our aggregation strategy is efficient in this regard and that our first two hypothesis are verified on these datasets.

We now discuss the clusterings obtained with the Infomap algorithm using the different models. The relevant statistics are reported in the bottom part of Tab. 4, the cumulative distributions for $N_{\mathcal{C}}$ are given in the right panels of Fig. 4. The similarity (according to the NMI) between the clusterings can be found in Tab. 5.
The clusterings found for AGG-VON$_2$ and VON$_2$ networks are almost identical for the Airports dataset. The reported NMI between these networks is smaller in the case of the Maritime dataset. In this case, the $N_{\mathcal{C}}$ are lower for most of the locations with the aggregated VON$_2$ model (see. Fig. 4d). This means there is less overlaps between the clusters. The biggest difference in clustering results between AGG-VON$_2$ and the rest appears with the Taxi dataset where the VON$_2$ clustering is closer to the FON$_2$ clustering. In this case, the number of cluster *per* location is significantly lower with AGG-VON$_2$. Also, the relevance of the clusterings quantified with ΔL values is lower overall. This seems consistent with the results of the case study of Section 4.1 since it means that the clusterings are less well defined.

Even if those real-world datasets do not include a ground-truth clustering, we can conclude that using AGG-VON$_2$ model can lead to significantly different clusterings. This suggests that the impact of the number of representations on random-walk-based algorithms is not marginal.



(a) Airports N_V values

(b) Airports N_C values

(c) Maritime N_V values

(d) Maritime N_C values

(e) Taxis N_V values

(f) Taxis N_C values

Figure 4: Cumulative distribution of the number of representations $N_{\mathcal{V}}$ (left column) and clusters $N_{\mathcal{C}}$ (right column) for locations. For each panel, $y(x)$ gives the ratio of location having at most x representations/found in at most x clusters. Note that the y-axis range varies among the panels.

We report in Tables 3 and 4 the computation times for the construction of each network and the average time taken by the Infomap algorithm to produce one clustering. We can see that the

Table 4. : Clustering results comparison.

Dataset	**Network**	**Time Clust.**(avg)	$\lvert\mathcal{C}\rvert$	**avg** $N_{\mathcal{C}}$	**max** $N_{\mathcal{C}}$	ΔL
Airports	$\text{VON}_2(1)$	5.59s	22	1.29	12	51.7%
	AGG-VON_2	2.93s	24	1.30	12	49.3%
	FON_2	5.98s	56	1.42	20	51.8%
Maritime	$\text{VON}_2(1)$	1.41s	8	1.28	5	55.8%
	AGG-VON_2	0.56s	7	1.15	4	52.4%
	FON_2	1.25s	43	1.43	7	57.0%
Taxis	$\text{VON}_2(1)$	0.08s	64	6.22	16	46.7%
	AGG-VON_2	0.02s	33	3.90	11	44.0%
	FON_2	0.05s	82	6.85	14	48.4%

Table 5. : NMI between clusterings found with VON_2, Agg VON_2 and FON_2.

Dataset	**Networks**	AGG-VON_2	FON_2
Airports	$\text{VON}_2(1)$	0.88	0.48
	AGG-VON_2	-	0.51
Maritime	$\text{VON}_2(1)$	0.66	0.42
	AGG-VON_2	-	0.35
Taxis	$\text{VON}_2(1)$	0.45	0.69
	AGG-VON_2	-	0.39

aggregation of representations significantly slows down the network construction. The additional time required is obviously dependent on the number of nodes in the VON_2 network. However, an important speed up is achieved when using Infomap. Remember that, since the algorithm is non-deterministic, it should be run multiple times and only the average time for a single run is reported here.

5. Discussion and Future Works

5.1 Extension of the aggregated model to any order

The results presented in this paper are valid in the limited case where the maximum order of representation is 2. The AGG-VON_2 model shows that we can achieve very different clustering results by using a smaller network model that captures the flow dynamics almost as well. However, for the different datasets considered here, previous studies suggest that strong dependencies exist at higher-order. Indeed, general VON networks contain significantly more representations of each location than VON_2 networks. We can expect the effects on random-walk-based clustering described in Section 3.2 to be, not only still relevant, but strengthened. For instance, the VON network for the Airports, Maritime and Taxis datasets contain around 443K, 18K and 4K nodes respectively. These observations illustrate the need to generalize our aggregated model.

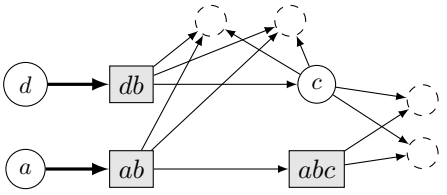


Figure 5: Example of ambiguous case when trying to build the aggregated network when representations of order greater than 2 exist. We assume distributions P_{db} and P_{ab} are similar. In this context, merging the corresponding nodes together will break the relation $ab \rightarrow abc$ that is required to encode this 3rd-order representation of c.

One important issue is the fact that merging nodes in a VON$_k$ with $k > 2$ may introduce ambiguities in the encoded sequential dependencies. Indeed, relations between higher-order nodes are a way to force a random walker into specifics destinations as expressed in Prop. 1. In the example given in Fig. 5, two representations of order 2 are similar in terms of the next visited location but they should not be merged, since doing so would break those constraints. This situation never occurs with a maximum order of 2, as shown in Section 3.3.

This issue seriously limits the gain obtained by merging groups of representations into a single node. This transformation is useful for VON$_2$ as it addresses the second and third effects discussed in Section 3.2. Indeed, for the network in Fig. 2b, the probability for a random surfer to teleport to any of $\{av, bc, cv\}$ or $\{dv, ev\}$ is $\frac{3}{10}$ or $\frac{2}{10}$ respectively, while it is $\frac{1}{8}$ for both in Fig. 2c. This mitigates the second effect as the probability to use the code word for a representation of v is reduced. Moreover, merging those nodes in Fig. 2c corresponds to the constraint of always considering them as being part of the same clusters. This cancels the third effect.

A possible solution for Infomap clustering of VON networks (at any order) is to still use *a priori* computed PageRank values but now using non-uniform teleportation rates corresponding to the merged representations. For example, the probability to teleport to any of the $\{av, bc, cv\}$ would be $\frac{1}{3}\frac{1}{8}$. Additionally, groups of representations should be moved together from a cluster to another while searching for the best partition. This last constraint would require important changes to the Infomap algorithm. Notice that in this study we only changed the input of the algorithm and used already available options.

5.2 Others Network mining tools

The results presented in this paper are in agreement with the idea that Infomap can not be directly applied to higher-order networks. As locations' clusters are built from a partition of its representations, the number and distribution of the representations have an important impact. These arguments point toward the importance of adapting classic network mining tools to VON. A similar conclusion was reached when looking at the PageRank metric Coquidé *et al.* (2022) as the more a location is represented in the higher-order network the higher its PageRank is. It is our opinion that multiple network representations of the same datasets are possible and that variable-order networks can successfully capture flow dynamics of the input sequences. However, efficient network mining tools should take the variability of these possible network models into account. This represents a challenge for researchers working in this domain.

This paper focused on random-walk-based clustering methods where we demonstrated that important divergence can be achieved with the Infomap algorithm using our aggregated VON model. There are different clustering algorithms that also rely on random walks. We can mention

the *Walktrap* algorithm (Pons and Latapy, 2006). It is based on the comparison of the set of reachable nodes using short random walks and therefore does not use a teleportation mechanism. Thus, it will be interesting to adapt it to VON networks.

Notes

1 The source code and datasets used for the experiments are available at`https://github.com/fqueyroi/von2network-clust`
2 `https://www.transtats.bts.gov/`
3 `https://www.kaggle.com/crailtap/taxi-trajectory`
4 `www.mapequation.org/infomap/` (version 1.3.0)

References

Xu, J., Wickramarathne, T. L., & Chawla, N. V. (2016). Representing higher-order dependencies in networks. *Science advances*, 2(5), e1600028.

Saebi, M., Xu, J., Kaplan, L. M., Ribeiro, B., & Chawla, N. V. (2020). Efficient modeling of higher-order dependencies in networks: from algorithm to application for anomaly detection. *EPJ Data Science*, 9(1), 15.

Saebi, M., Xu, J., Grey, E., Lodge, D., Corbett, J., & Chawla, N. V. (2020). Higher-order patterns of aquatic species spread through the global shipping network. *PLOS ONE*, 15.

Ron D., Singer y., & Tishby N. (1994). Learning probabilistic automata with variable memory length. In *Proceedings of the seventh annual conference on Computational learning theory* (COLT '94). Association for Computing Machinery, New York, NY, USA, 35–46.

Rosvall, M., Axelsson, D., & Bergstrom, C. T. (2009). The map equation. *The European Physical Journal Special Topics*, 178(1), 13-23. Springer.

Pons, P., & Latapy, M. (2006). Computing communities in large networks using random walks. In *J. Graph Algorithms Appl.* 191-218.

McDaid, A. F., Greene, D., & Hurley, N. (2011). Normalized mutual information to evaluate overlapping community finding algorithms. *arXiv preprint* arXiv:1110.2515.

Rosvall, M., Esquivel, A. V., Lancichinetti, A., West, J. D., & Lambiotte, R. (2014). Memory in network flows and its effects on spreading dynamics and community detection. *Nature communications*, 5(1), 1-13.

Lancichinetti, A., Fortunato, S., & Radicchi, F. (2008). Benchmark graphs for testing community detection algorithms. *Physical review E*, 78(4), 046110.

Scholtes, I. (2017). When is a network a network? Multi-order graphical model selection in pathways and temporal networks. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*. 1037-1046.

Xie, J., Kelley, S., & Szymanski, B. K. (2013). Overlapping community detection in networks: The state-of-the-art and comparative study. Acm computing surveys (csur), 45(4), 1-35.

Brin, S., & Page, L. (1998). The anatomy of a large-scale hypertextual web search engine. *Computer networks and ISDN systems*, 30(1-7), 107-117.

Begleiter, R., El-Yaniv, R., & Yona, G. (2004). On prediction using variable order Markov models. *Journal of Artificial Intelligence Research*, 22, 385-421.

Manning, C. D., Raghavan, P. & Schütze, H. (2008). Hierarchical clustering. Cambridge University Press. pages 346–368

Dao, V. L., Bothorel, C., & Lenca, P. (2020). Community structure: A comparative evaluation of community detection methods. *Network Science*, 8(1), 1-41. Cambridge University Press.

Chen, R., Sun, H., Chen, L., Zhang, J., & Wang, S. (2021). Dynamic order Markov model for categorical sequence clustering. *Journal of Big Data*, 8(1), 1-25.

Ching, W. K., Fung, E. S., & Ng, M. K. (2004). Higher-order Markov chain models for categorical data sequences. *Naval Research Logistics* (NRL), 51(4), 557-574.

Eliassi-Rad, T., Latora, V., Rosvall, M., Scholtes, I., & Dokumente, G. (2021). Higher-Order Graph Models: From Theoretical Foundations to Machine Learning. *Dagstuhl Reports*. Dagstuhl Seminar 21352.

Torres, L., Blevins, A. S., Bassett, D., & Eliassi-Rad, T. (2021). The why, how, and when of representations for complex systems. *SIAM Review*, 63(3), 435-485.

Lambiotte, R., Rosvall, M., & Scholtes, I. (2019). From networks to optimal higher-order models of complex systems. *Nature physics*, 15(4), 313-320.

Jääskinen, V., Xiong, J., Corander, J., & Koski, T. (2014). Sparse Markov chains for sequence data. *Scandinavian Journal of Statistics*, 41(3), 639-655.

Krieg, S. J., Kogge, P. M., & Chawla, N. V. (2020). GrowHON: A Scalable Algorithm for Growing Higher-order Networks of Sequences. In *International Conference on Complex Networks and Their Applications* (pp. 485-496). Springer, Cham.

Battiston, F., Cencetti, G., Iacopini, I., Latora, V., Lucas, M., Patania, A., & Petri, G. (2020). Networks beyond pairwise interactions: structure and dynamics. *Physics Reports*, 874, 1-92.

Coquidé, C., Queiros, J., & Queyroi, F. (2021). PageRank computation for Higher-Order Networks. In *International Conference on Complex Networks and Their Applications* (pp. 183-193). Springer, Cham.

Notations

Table 6. : Summary table for notations

	Related to sequences
$\mathcal{A}$	Set of locations (itemset)
σ	Generic element of $\mathcal{A}$
$\mathcal{S}$	Set of sequences (dataset)
$s = \sigma_1\sigma_2\sigma_3\ldots$	A sequence of locations
$c(s\sigma) : \mathcal{A} \to \mathbb{N}^+$	Occurrences of sequence $s\sigma$ in $\mathcal{S}$
$p(\sigma\|s) : \mathcal{A} \to [0,1]$	Transition probability from context s to σ
$C_s = \big(c(s\sigma_1), \ldots, c(s\sigma_{\|\mathcal{A}\|})\big)$	Occurrences of each σ_i following context s
$P_s = \big(p(\sigma_1\|s), \ldots, p(\sigma_{\|\mathcal{A}\|}\|s)\big)$	Distribution of each σ_i following context s
	Related to networks
$\textsc{Fon}_k$	Fixed order k network
$\textsc{Von}_k(\alpha)$	Variable-order network with max order k using threshold multiplier $\alpha \in \mathbb{R}^+$ (Eq. 2)
$\textsc{Min-Von}_k$	Minimal Von with max order k
$\textsc{Agg-Von}_k$	Aggregated Von with max order k
$\mathcal{V}$	Set of nodes (location representations)
$\mathcal{V}_k$	Set of representations of order k
$\mathcal{V}(\sigma)$	Set of representations of location σ
$N_{\mathcal{V}} = \big(\|\mathcal{V}(\sigma_1)\|, \ldots, \|\mathcal{V}(\sigma_{\|\mathcal{A}\|})\|\big)$	Number of representations of each location
$\mathcal{C} = \{\mathcal{C}_1, \ldots, \mathcal{C}_m\}$	Partition of $\mathcal{V}$
π_v	PageRank of node v
$L(\mathcal{C})$	Map Equation (Infomap computation)