Statistical Models Coupling Allows for Complex Local Multivariate Time Series Analysis

Veronica Tozzo*†
vtozzo@mgh.harvard.edu
Center for System Biology and Department of Pathology, Massachusetts General Hospital
Department of Systems Biology, Harvard Medical School
Boston, Massachusetts, USA

Federico Ciech*
Davide Garbarino
Alessandro Verri
DIBRIS - Università degli Studi di Genova
Genova, Italy

ABSTRACT

The increased availability of multivariate time-series asks for the development of suitable methods able to holistically analyse them. To this aim, we propose a novel flexible method for data-mining, forecasting and causal patterns detection that leverages the coupling of Hidden Markov Models and Gaussian Graphical Models. Given a multivariate non-stationary time-series, the proposed method simultaneously clusters time points while understanding probabilistic relationships among variables. The clustering divides the time points into stationary sub-groups whose underlying distribution can be inferred through a graphical model. Such coupling can be further exploited to build a time-varying regression model which allows to both make predictions and obtain insights on the presence of causal patterns. We extensively validate the proposed approach on synthetic data showing that it has better performance than the state of the art on clustering, graphical models inference and prediction. Finally, to demonstrate the applicability of our approach in real-world scenarios, we exploit its characteristics to build a profitable investment portfolio. Results show that we are able to improve the state of the art, by going from a -%20 profit to a noticeable 80%.

CCS CONCEPTS

• **Theory of computation → Unsupervised learning and clustering**; *Structured prediction*; Nonconvex optimization; • **Computing methodologies → Learning latent representations**; • **Mathematics of computing → Markov networks**.

KEYWORDS

Multivariate time-series, Graphical Models, Non-stationary processes, Forecasting, Causality

*Both authors contributed equally to this research.
†Work partially done while at Università di Genova.

KDD '21, August 14–18, 2021, Virtual Event, Singapore.

ACM ISBN 978-1-4503-8332-5/21/08…$15.00
https://doi.org/10.1145/3447548.3467362

ACM Reference Format:
Veronica Tozzo, Federico Ciech, Davide Garbarino, and Alessandro Verri. 2021. Statistical Models Coupling Allows for Complex Local Multivariate Time Series Analysis. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '21), August 14–18, 2021, Virtual Event, Singapore.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3447548.3467362

1 INTRODUCTION

Local multivariate time series analysis involves a variety of tasks that aim at studying the characteristics of subsequent points in time, such as their short or long term similarities or effect on future behaviour.

Consider the example of stock trends in a financial system. Each stock has an associated value that changes in time. Such value could possibly depend on other stocks behaviour as well as external environmental causes. At each time point one may want to obtain various insights on such stock values. Here, we refer to the following scenarios: *(a)* understanding how stocks are related to each other (*multivariate correlation analysis*) [7] ; *(b)* detecting patterns of stock interactions repeated in time (*time points clustering*) [28, 41]; *(c)* predicting their value at the next time-point (*forecasting*) [8, 37, 52], and, lastly *(d)* understanding possible causal relationships among them (*non-stationary causality*) [40].

These tasks aim at locally characterizing every single observation in the time series under different perspectives. If applied on the same time series, these local characterizations can naturally enable a global understanding of the underlying system. Typically, these tasks are approached separately by restricting the problem to a constrained setting (*i.e.*, strong assumptions on data) with the intent of having the best possible performances on such domain [14, 18, 19, 25, 30, 36, 44, 47]. Nonetheless, we argue that these tasks are deeply connected to one another (Figure 1). Solving them globally may therefore result both in an increased accuracy per-task and a more in-depth insight on the system as a whole. On the latter, especially, the unified analysis makes the *a posteriori* global analysis easier to perform as the learned models are coherent with the initial assumptions and between each other by construction.

The main contribution of this paper is a framework that solves tasks *a-d* globally, filling a missing gap in the literature. Our method synergically solves the four tasks by leveraging their dependencies as depicted in Figure 1. It achieves this goal with a simple coupling of two statistical models which allows to retain a single set of

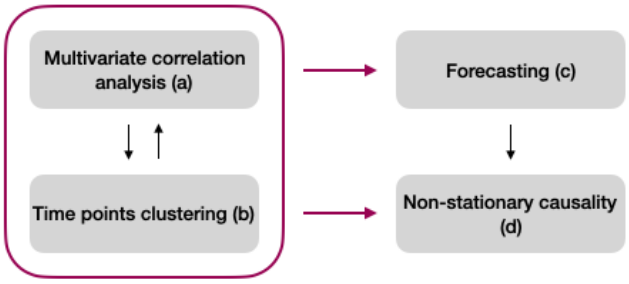


Figure 1: Schematic representation of the four tasks and how they impact each other in our framework. Multivariate correlation analysis (task *(a)*) and time point clustering (task *(b)*) are mutually supportive to one another, together (purple arrows) they allow for forecasting (task *(c)*) and the understanding of non-stationary causal patterns (task *(d)*). This last is also influenced by forecasting.

mild assumptions on the data. More formally, we present a statistical model, *Time Adaptive Gaussian Model* (TAGM), that combines Hidden Markov Models (HMMs) and Gaussian Graphical Models (GGMs). The main idea is presented in Figure 2. Given a time series $\{\mathbf{x}_1, \ldots, \mathbf{x}_6\} \subset \mathbb{R}^5$, we assign to each time point a hidden state $\{\mathbf{z}_1, \ldots, \mathbf{z}_6\}$ that, in the specific example, assumes $K = 2$ values k_1 or k_2. Suppose to observe the state k_1, all observations assigned to that state (*i.e.*, $\{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_5\}$) are then i.i.d. and assumed to be drawn from a multivariate Gaussian distribution $\mathcal{N}(\mu_1, \Theta_1^{-1})$. The associated distribution is inferred through a GGM from $\{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_5\}$. Specifically, a GGM univocally determines the underlying distribution through a graph that is encoded in the matrix Θ_1. Such matrix is called *precision matrix* and is the inverse of the covariance matrix. In the graph we can directly read pairwise conditional independencies among variables, *e.g.* $v_1 \perp\!\!\!\perp v_2 | v_3, v_4, v_5$ if and only if $\Theta_1(1, 2) = 0$, or, equivalently, there is no edge between the two nodes in the corresponding graph.

TAGM naturally solves tasks *(a)* and *(b)* (Section 2). Indeed, we can divide the time points into clusters by pairing each cluster to a hidden state. At the same time, multivariate correlation analysis is achieved through the inference of the corresponding underlying graph. This coupling is both advantageous and necessary. On the one hand, it is advantageous since knowledge on the variable dependencies can be leveraged to better assign each time point to a cluster. On the other hand, it is necessary as the inference of a GGM requires more than one observation. Thus, we can exploit the division into clusters to collect independent and equally distributed samples. TAGM can also be directly used for forecasting (task *(c)*, Section 3). At training time, we construct an augmented version of the input time series in which we concatenate each time point with the next one (Figure 6). We then apply TAGM on this augmented time series to infer the relative clusters and variable dependencies. At prediction time, we predict the next most probable state/cluster by exploiting the information on the previous inferred states in the HMM. The values of the unseen point are then estimated exploiting the GGM associated to that cluster. The same procedure can be used to understand causal patterns (task *(d)*, Section 4). Indeed, the learned statistical dependencies and sequentiality allow us to have

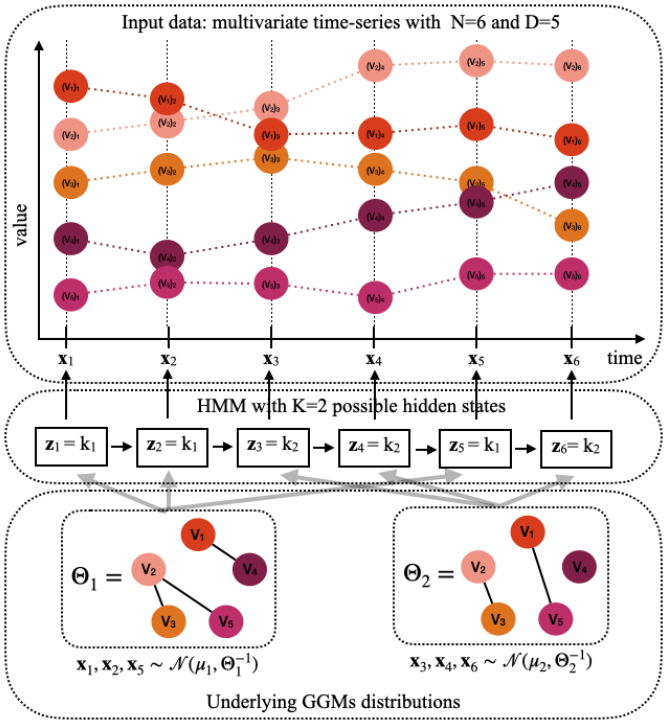


Figure 2: Schematic representation of the simultaneous instantiation of multivariate correlation analysis and time point clustering. Given a time series of multi-dimensional vectors, through an HMM, we associate each time point to a hidden state (*i.e.*, cluster). Given the K=2 hidden states, we infer two GGMs that model the underlying distributions. Note that, if we observe the Markov chain of the HMM, we can assume the time points associated to a specific state to be *i.i.d.*

insights on causal relationships. Note that, the ability of detecting more than one GGM allows for non-stationary causality patterns, which entails high flexibility.

We extensively evaluate TAGM on synthetic data sets and show how it leads to significant improvement compared to the state of the art in clustering, learning graph structures and prediction. Results are presented in Section 2.2 for tasks *(a)* and *(b)*, and Section 3.1 for task *(c)*. We also show the applicability of our method to real-world scenarios through a financial use case (Section 5). In particular, we exploit it to construct a profitable investment portfolio [13]. The standard approach used in finance is to fix a temporal window on which perform correlation analysis [39]. Such approach does not account for underlying changes and thus assumes all time points in that window to be stationary. Differently, our method removes such assumption while also exploiting the interpretability of GGMs to understand the dependencies among stocks. This is fundamental, as simple correlation analysis often carries spurious information that can lead to poor decision making. We show that TAGM overcomes the state-of-the-art approach for building a portfolio in terms of profit and loss variations, obtaining a 80% profit compared to the -%20 of standard methods. We conclude the paper showing how the proposed approach is interconnected with a variety of

state-of-the-art methods (Section 6) and suggesting possible future improvements or research directions (Section 7).

2 THE TIME ADAPTIVE GAUSSIAN MODEL

TAGM is based on the assumption that the system under analysis is non-stationary, *i.e.*, the underlying distribution of variables at each time point may change in time. One of the possible methods to analyse sequential non-stationary data is a Hidden Markov Model (HMM) [2]. HMMs assume that the series of observations is generated by a certain number of (hidden) internal states connected through a Markov chain of latent variables. If we consider Figure 2, the latent variables are modelled as $\mathbf{z}_1, \ldots, \mathbf{z}_6$. Such variables may assume possible $K = 2$ different states, each of these gets associated to a possibly different distribution. The family of such distributions depends on the type of data in analysis, in this paper, for simplicity, we consider continuous data and we assume underlying multivariate Gaussian distributions. Nonetheless, adopting the same idea for other distribution assumptions is straightforward from the model definition. Differently from standard approaches, instead of directly estimating the empirical means and covariances of such distributions from observations, we associate a graphical model where the lack of an edge explicitly encodes the conditional independencies between a pair of variables. We exploit Gaussian Graphical Models (GGMs) [29] where the precision matrices (Θ_1 and Θ_2) are the inverse of the covariance matrices and they can be interpreted as the adjacency matrices of a graph. This switch of perspective still allows us to estimate a multivariate Gaussian distribution, while allowing for directly imposing sparsity on the adjacency matrices [18]. Sparsity is fundamental as: it provides a higher stability to noise in presence of fewer samples; it grants higher interpretability as it allows for identifying the most relevant dependencies (*i.e.* the edges) while removing spurious correlations that would be captured by the empirical covariance matrix; and, it allows us to extend this method to perform prediction as well as understand causality patterns (more details about this will be provided in Section 3 and 4).

2.1 The model

Consider N sequential (temporal) observations, $\mathbf{x}_n \in \mathbb{R}^D$ for $n = 1, \ldots, N$ on D variables. We assume that such observations follow a non-stationary process, and thus are generated by possibly more than one underlying distribution. The number of such distributions could be ideally infinite [4], but, for the sake of simplicity, we here assume them to be fixed to K. In order to map each observation $\mathbf{x}_n$ to one distribution, HMM pairs them to a hidden (latent) variable $\mathbf{z}_n \in \{0, 1\}^K$, that has only one non-zero component at position k if the n-th observation is associated to the k-th distribution. In the rest of the paper, we will use the notation $z_{n,k}$ to indicate the k-th positional value of the vector $\mathbf{z}_n$.

The sequence of latent variables follows a Markov chain process, meaning that $\mathbf{z}_{n+1} \perp\!\!\!\perp \mathbf{z}_{n-1} | \mathbf{z}_n$. As a consequence, if we condition on the latent variables, the observations $\mathbf{x}_n$ and $\mathbf{x}_{n+1}$ become independent thus allowing to freely use them to infer the corresponding underlying distributions (see Figure 2).

The joint distribution on observed $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N)$ and latent variables $\mathbf{Z} = (\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_N)$ is given by

$$p(\mathbf{X}, \mathbf{Z}|\pi, A, \phi) = p(\mathbf{z}_1|\pi)\Big[\prod_{n=2}^{N} p(\mathbf{z}_n|\mathbf{z}_{n-1}, A)\Big]\prod_{n=1}^{N} p(\mathbf{x}_n|\mathbf{z}_n, \phi), \quad (1)$$

where [2]:

- the probability $p(\mathbf{z}_1|\pi) = \prod_{k=1}^{K} \pi_k^{z_{1,k}}$, with $\sum_k \pi_k = 1$, is the initial latent node $\mathbf{z}_1$ probability, which differs from the other states as there is no parent node;
- the probability $p(\mathbf{z}_n|\mathbf{z}_{n-1,A}) = \prod_{k=1}^{K}\prod_{j=1}^{K} A_{j,k}^{z_{n-1,j}z_{n,k}}$ is the *transition probability* of moving from one state to the other. Here, $A \in [0, 1]^{K\times K}$ is the *transition matrix* that we assume to be constant in time and it is defined as
$$A_{j,k} = p(z_{n,k} = 1|z_{n-1,j} = 1)$$
with $0 \leq A_{j,k} \leq 1$ and $\sum_k A_{j,k} = 1$.
- the probabilities $p(\mathbf{x}_n|\mathbf{z}_n, \phi) = \prod_{k=1}^{K} p(\mathbf{x}_n|\phi_k)^{z_{n,k}}$, are the *emission probabilities* where $\phi = \{\phi_1, \ldots, \phi_K\}$ is a set of K different parameters governing the distributions, one for each of the possible K states.

Combining HHMs with GGMs is achieved by setting the emission probabilities to

$$p(\mathbf{x}_n|\mathbf{z}_n, \phi) = \prod_{k=1}^{K} \mathcal{N}(\mathbf{x}_n|\mu_k, \Theta_k^{-1})^{z_{n,k}}$$

in such a way to have an explicit correspondence between the distribution of each state and a graph, modelled through the precision matrix Θ_k. Moreover, we impose sparsity by coupling the emission probabilities with a Laplacian prior on the precision matrices. The posterior of the final model is defined as

$$p(\pi, A, \mu, \mathbf{\Theta}|\mathbf{X}, \mathbf{Z}) \propto p(\mathbf{z}_1|\pi)\Big[\prod_{n=2}^{N} p(\mathbf{z}_n|\mathbf{z}_{n-1}, A)\Big] \prod_{n=1}^{N}\prod_{k=1}^{K} \mathcal{N}(\mathbf{x}_n|\mu_k, \Theta_k^{-1})^{z_{n,k}} e^{-\frac{\lambda}{2}||\Theta_k||_{1,od}}. \quad (2)$$

Optimization. The optimization of parameters $\theta = \{\pi, A, \mu, \mathbf{\Theta}\}$ in functional (2) is performed through a Maximum A Posteriori approach. In particular, we employ the Baum's version of the *expectation maximization* (EM) algorithm [3]. In short, it consists in an alternating minimization procedure, in which the parameters π, A, μ are updated as in a standard HMM, while the inference of the $\mathbf{\Theta}$ reduces to solving a Graphical Lasso [18]

$$\Theta_k = \underset{\Theta>0}{\operatorname{argmin}}\ \operatorname{tr}(\Theta \mathbf{S_k}) - \log\det(\Theta) + \lambda\|\Theta\|_{1,od} \quad (3)$$

where S_k is the empirical covariance matrix of the observations that are associated to the k-th hidden state, and if we denote such observations as $\mathbf{X}_k$, it is defined as

$$\mathbf{S}_k = \frac{1}{\sum_{n=1}^{N} z_{nk}} \sum_{n=1}^{N} z_{nk}(\mathbf{x}_n - \hat{\boldsymbol{\mu}}_k)(\mathbf{x}_n - \hat{\boldsymbol{\mu}}_k)^{\mathrm{T}}, \hat{\boldsymbol{\mu}}_k = \frac{1}{\sum_{n=1}^{N} z_{nk}} \sum_{n=1}^{N} z_{nk}\mathbf{x}_n.$$

The objective value $\operatorname{tr}(\Theta S_k) - \log\det(\Theta)$ is the negative log likelihood of the multivariate normal distribution and $\|\cdot\|_{1,od}$ is the off-diagonal ℓ_1-norm that imposes sparsity on the precision matrix Θ without considering the diagonal elements. More details on the optimization of this model can be found in [11].

Higher order and online extensions. TAGM could benefit from two extensions to better handle real-world scenarios. The first extension is an online learning variation, *Incremental TAGM* or *IncTAGM*, that could be useful in presence of high-frequency data. Indeed, in real-world contexts where new observations arrive at a high rate (*e.g.* every second) one may want to be able to fine tune the model online in order to consider such observations instantaneously instead of re-fitting on the complete time series. We derive such extension following the idea in [10]. In practice IncTAGM starts as a standard TAGM on the available data, as a new time point becomes available the set of parameters is updated accordingly and the new point gets assigned to a state. Such update can be achieved by a variation in the E step of the EM algorithm, more details in [11].

The second extension is a higher-order Markov chain version, *Memory TAGM* or *MemTAGM*. It is based on the idea that latent states could have higher order dependencies, which are not captured if we consider a Markov chain of order one. In short, the main idea is to allow the emission probability of $\mathbf{x}_n$ to depend not only on $\mathbf{z}_n$ but also from the previous $m \in \mathbb{Z}^+$ sequence of states $p(\mathbf{x}_n|\{\mathbf{x}_\ell\}_{\ell<n}, \{\mathbf{z}_\ell\}_{\ell\leq n}) = p(\mathbf{x}_n|\{\mathbf{z}_\ell\}_{\ell=n-(m-1)}^{n})$. Each observation is conditionally independent of the previous ones and of the state sequence history, given the current and the preceding $m-1$ states. For the derivation of the related optimization algorithm we followed [20]. The main drawback is a much higher computational time as the transition matrix and the corresponding initial state dimensions increase, more details in [11].

2.2 Experimental assessment

In order to evaluate the performance of TAGM we devised three sets of synthetic experiments. We generated data fixing the number of states K; for each state $k = 1, \dots, K$, the mean $\mu_k \in \mathbb{R}^D$ is assumed to be drawn from a multivariate standard normal distribution $\mathcal{N}(\mathbf{0}, \mathcal{I})$, while the precision matrix $\Theta_k \in \mathcal{S}^{D\times D}$ is generated as a sparse semi-positive definite matrix. The sequence of hidden states is generated by sampling from a transition matrix $A \in [0,1]^{K\times K}$ where each row is sampled from a Dirichlet distribution (see Appendix A for more details). Given the sequence of states and the related μs and Θs, we draw a sequence of N samples in D dimensions. TAGM has two hyper-parameters, the number of hidden states K and the sparsity penalty λ in the Graphical Lasso (see Equation (3)). We cross-validated such parameters using the Bayesian Information Criterion (BIC) [43] (see Appendix B). Note that the problem is non-convex, thus different initialization may lead to different local minima. To handle this issue, we performed multiple initializations and we select the result with highest likelihood (possible ways for initializing the model are described in Appendix C). Results are presented in terms of V-measure for clustering performance [42], and Matthews correlation coefficient (MCC) for network inference performance [33] where we binarise the inferred precision matrix in such a way that 0 corresponds to a missing edge and 1 indicates an identified edge, see Appendix D for details. The optimization algorithm and data generation pipelines for the experiments are implemented within an open-source Python framework that contains the real-world dataset as well [1]. ß

[1] https://github.com/veronicatozzo/regain/tree/HMM

Studying asymptotic behaviour. As a first assessment we characterized the model in terms of number of observations needed to learn the data structure and its sensitiveness to external noise.On both experiments we fixed $D = 10$ variables and $K = 5$ states, for the first experiment we incremented the number of observations N until perfect inference is reached, while in the second experiment we fixed $N = 2000$ and we let the noise to signal ratio increments until the performances are equal to chance. The results are shown in Figure 3. As can be seen from the left panel of Figure 3, our model is able to converge to the real cluster labels after 400 observations while to infer the real graphs is necessary to have 10000 observations. On the right panel of Figure 3 we observe, as expected, that TAGM performance decreases as the noise standard deviation increases. We can also note that model performance remains good up to the point where noise to signal ratio is equal to one. Beyond that point the model is not able to distinguish the signal from the noise and therefore it reaches the performance of a random model. Given these results, for the following experiments we set the number of observations to $N = 2000$ and the noise to signal ratio to 1.

Clustering and network inference performance. Here, we wanted to assess the ability of TAGM to infer the correct states of the system and the related GGMs. We generated synthetic datasets allowing for both the number of states K and the number of dimensions D to vary in the sets $\{2, 5, 10, 15\}$ and $\{10, 15, 20, 30\}$ respectively and we fixed $N = 2000$. We compared TAGM with state-of-the-art methods, in particular HMM [2], Gaussian Mixture Models (GMM) [14], spectral clustering [36] and K-Means [32]. Of these methods the only one that directly provides an estimate of the underlying distribution is HMM, note that it provides the empirical covariance matrix that we need to invert to be able to compare it with the precision matrix. For the other methods, we first infer the clusters and then on the samples belonging to each cluster we perform Graphical Lasso separately. Results are shown in Figure 4. On the left we show the point per point behaviour of the methods as both the number of states and the number of dimensions vary, while on the right we show the mean behaviour across different dimensions for the different number of states. It is evident that TAGM is the one that performs best in both V-measure and MCC and that HMM and GMM have close performance in clustering but have less ability in detecting the true graph.

Higher order and on-line extension. We finally performed two experiments to compare TAGM to the online and the higher order extensions. To compare TAGM and IncTAGM, we generated a synthetic dataset with $K = 5$ states, $D = 10$ dimensions and $N = 2000$ observations. We wanted to assess the behaviour of IncTAGM with respect to TAGM as the percentage of initial data given in input $\hat{N} = \%N$ to IncTAGM increases. The results are shown on the left panel of Figure 5, where we can see that IncTAGM has reasonable performances when the percentage is low and it asymptotically tends to the performance of TAGM as the percentages of input data reaches 100%. To compare TAGM and MemTAGM we generated a synthetic dataset with $K = 3$ states, $d = 10$ dimensions and $N = 2000$ observations, while letting the memory of the hidden Markov process vary in the set $\{1, \dots, 5\}$. In this way we are able to evaluate the behaviour of MemTAGM with respect to TAGM as the memory of the hidden Markov process increases. The results

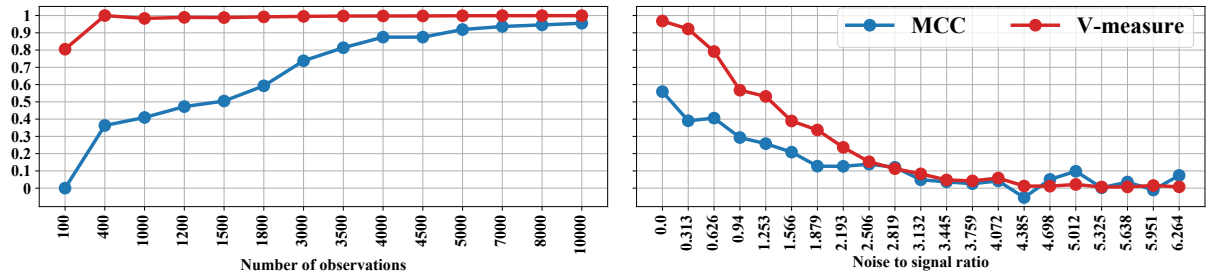


Figure 3: Asymptotic behaviour of TAGM on data. In the left panel we study the performance of the method as the number of observations increases, while on the right panel we fix the number of observations and we study the behaviour as the noise to signal ratio increases.

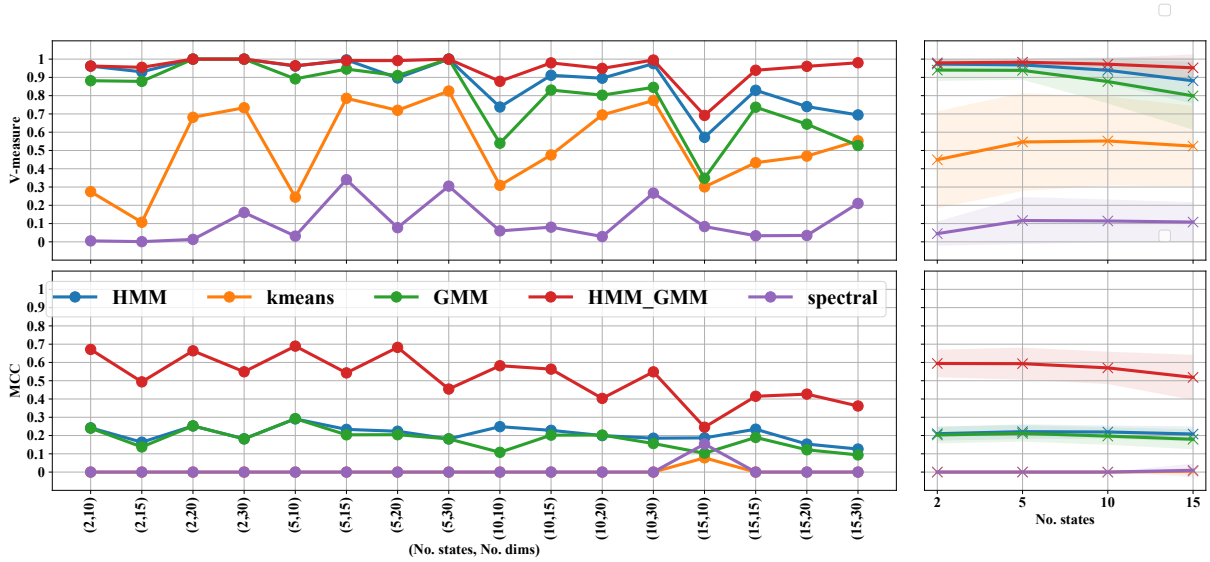


Figure 4: Comparison of TAGM with state-of-the-art methods in terms of V-measure and MCC. On the left panel we have the behaviour for different states and increasing dimensions, on the right panel the mean behaviour for the number of states.

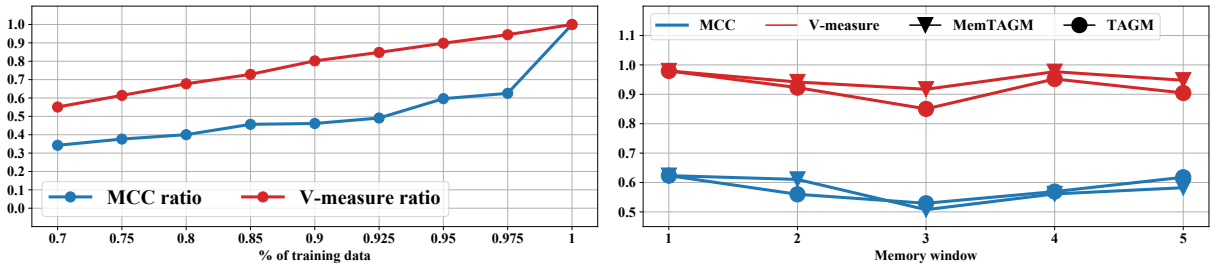


Figure 5: Comparison of TAGM with its extensions in terms of V-measure and MCC. On the left hand panel we drew the ratio V-measure(TAGM)/V-measure(IncTAGM) and the ratio MCC(TAGM)/MCC(IncTAGM) as the percentage of training data increases. On the right hand panel we drew the V-measure and MCC of MemTAGM and TAGM as the memory of the hidden Markov process increases.

are shown on the right panel of Figure 5, where we can see that MemTAGM performance in terms of V-measure is slightly better than TAGM for every considered memory window. On the other hand, the MCC results are comparable. We want to point out that the time complexity of the step of the optimization algorithm that assigns each point to a state is $O(K^2N)$, and MemTAGM number of states has a number of possible states $\hat{K} = K^\nu$ (with ν being the memory window), thus having complexity $O(K^{\nu 2}N)$. Therefore, the slight improvement in performances that we see in Figure 5 does

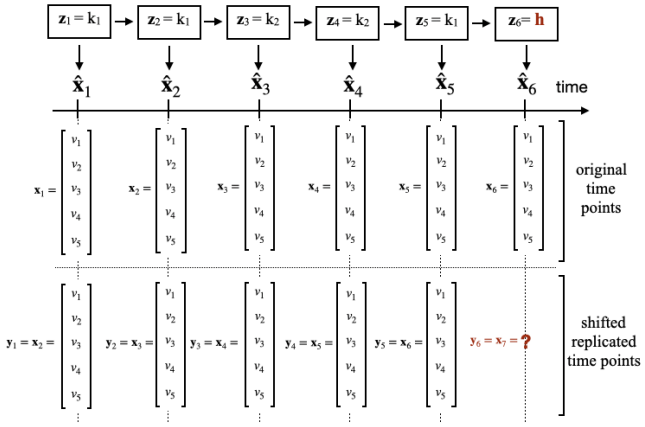


Figure 6: Schematic example of the construction of the augmented time series given in input to TAGM to perform predictions.

not justify the need of a such increased complexity and therefore higher computational time.

3 MAKING PREDICTIONS

Being able to predict future time points may be useful in applied contexts in which, for example, we seek to make decision based on unseen data. Here, we aim at performing a multi-output regression where given the values at time point n, $\mathbf{x}_n \in \mathbb{R}^D$, we want to predict the values at time point $n+1$, denoted as $\mathbf{y}_n \in \mathbb{R}^D$.

If we are provided N observations in D variables, in order to exploit TAGM to predict the observation at $N+1$, we first need to augment the time series. In practice for each time point $n = 1, \dots, N-1$ we stack $\mathbf{x}_n$ and $\mathbf{y}_n = \mathbf{x}_{n+1}$ in a new vector $\hat{\mathbf{x}}_n = [\mathbf{x}_n, \mathbf{y}_n] \in \mathbb{R}^{2D}$ (see Figure 6).

If we now apply TAGM on the newly built time series, we obtain, for each time point its state and related underlying GGMs.

Suppose now that, at time n, we inferred a hidden state k with the related precision matrix $\Theta^k \in \mathbb{R}^{2D \times 2D}$. Such matrix can be divided in blocks as

$$\Theta^k = \begin{pmatrix} \Theta^k_{\mathbf{xx}} & \Theta^k_{\mathbf{xy}} \\ \Theta^{k\top}_{\mathbf{xy}} & \Theta^k_{\mathbf{yy}} \end{pmatrix}$$

where $\Theta^k_{\mathbf{xx}}$ indicates the sub-matrix that encodes the conditional independencies of the vector $\mathbf{x}_n$, $\Theta^k_{\mathbf{yy}}$ denotes the sub-matrix of conditional independencies of the vector $\mathbf{y}_n$ and the block $\Theta^k_{\mathbf{xy}}$ encodes the conditional independencies among $\mathbf{x}_n$ and $\mathbf{y}_n$. Similarly, we can divide the inferred means vector as $\mu^k = [\mu^k_{\mathbf{x}}, \mu^k_{\mathbf{y}}]$.

We can now observe that each $\mathbf{y}_n$ is normally distributed, indeed

$$p(\mathbf{y}_n | \mathbf{x}_n, z_{n,k} = 1) = \mathcal{N}(\bar{\mu}, \bar{\Theta}^{-1}) \tag{4}$$

where

$$\bar{\mu} = \mu^k_{\mathbf{y}} + (\Theta^k_{\mathbf{xy}})^\top (\Theta^k_{\mathbf{xx}})^{-1} (\mathbf{x}_n - \mu^k_{\mathbf{x}}), \tag{5}$$

and,

$$\bar{\Theta} = \Theta^k_{\mathbf{yy}} - (\Theta^k_{\mathbf{xy}})^\top (\Theta^k_{\mathbf{xx}})^{-1} \Theta^k_{\mathbf{xy}}. \tag{6}$$

Then, given a time-series of length N, and an inferred TAGM model on the augmented time series on the first $1, \dots, N-1$ time points, we aim at estimating the unknown values of $\mathbf{y}_N$, given the observed $\mathbf{x}_N$ as $\mathbf{y}_N = f(\mathbf{x}_N)$. It is trivial to observe that the minimization of the expected squared prediction error $f(\mathbf{x}_N) = \mathbb{E}[\mathbf{y}_N | \mathbf{x} = \mathbf{x}_N]$ corresponds to Equation (5). Thus, we predict the value of $\mathbf{y}_N$ as

$$\mathbf{y}_N = \mu^h_{\mathbf{y}} + (\Theta^h_{\mathbf{xy}})^\top (\Theta^h_{\mathbf{xx}})^{-1} (\mathbf{x}_N - \mu^h_{\mathbf{x}}) \tag{7}$$

which corresponds to a time-varying lasso linear regression [17]. This regression model assumes the knowledge of h, *i.e.*, the hidden state assigned at time point N. Such state cannot directly be inferred from data because we do not have the complete values for $\hat{x}_N$, but it can be estimated propagating the information from the Markov chain of the HMM. To this end we exploit the Viterbi method [15].

This approach not only allows us to estimate the values of $\mathbf{y}_N$, it also provides information on the predicted underlying GGM whose precision matrix is obtained as in Equation (6).

Moreover, this approach is flexible to consider more than one previous time point for the prediction of $\mathbf{y}_N$. Indeed, if we want to exploit information on a window of length w, it is sufficient to build an augmented time series where, for each $n = 1, \dots, N-1$, we define $\hat{\mathbf{x}}_n = [\mathbf{x}_{n-w}, \dots, \mathbf{x}_n, \mathbf{y}_n] \in \mathbb{R}^{Dw}$.

Method	K=2	K=3	K=4	K=5
Emp Cov last 25 days	0.23 ± 0.03	0.14 ± 0.07	0.00 ± 0.09	0.04 ± 0.12
Emp Cov last 50 days	0.29 ± 0.02	0.17 ± 0.13	0.02 ± 0.06	0.09 ± 0.15
Emp Cov last 100 days	0.31 ± 0.05	0.12 ± 0.11	0.04 ± 0.11	0.05 ± 0.09
TAGM	**0.65 ± 0.03**	**0.56 ± 0.09**	**0.62 ± 0.11**	**0.67 ± 0.12**

Table 1: Performance in the prediction of the next precision matrix in terms of MCC.

3.1 Experimental assessment

We evaluated the performance of TAGM for prediction on one synthetic experiment. Data are fixing $K = 2, 3, 4, 5$ and $N = 2000$. The generation method is as described in Section 2.2 and Appendix A. The hyper-parameters are cross-validated with BIC (Appendix B) and results are presented in terms of Mean Absolute Error (MAE) (Appendix D). For the estimate of the next time point precision matrix we compared the performances of TAGM with the inverse of the empirical covariance matrix of the last 25, 50 and 100 days on a time series of dimension $D = 10$. The results are in Table 1 where we observe that TAGM greatly outperforms the prediction compared to the estimate of the covariance matrix. For the evaluation of the prediction of the specific values we compared our model with Gradient Boosting (LGB) [30], Long-Short Term Memory Neural Network (LSTM) [25], Kernel regression with Gaussian assumption (Kernel RBF) [47] and vector autoregression (VAR) [44] on a time-series of $D = 5$ variables. The results are in Table 2 where we observe that TAGM has always a lower prediction error compared to all the other considered methods.

4 UNDERSTANDING CAUSALITY

The ability of making predictions allows us to connect the inferred GGMs to Granger causality test [19]. Multivariate Granger causality analysis aims at detecting those variables that across all time series are causal for other. Typically, this is achieved by fitting an autoregressive model on the time series. The main drawback of

Method	K=2	K=3	K=4	K=5
Lgb	1.17 ± 0.23	1.43 ± 0.46	1.95 ± 0.45	1.55 ± 0.63
LSTM	1.16 ± 0.26	1.42 ± 0.45	2.06 ± 0.51	1.50 ± 0.38
VAR	1.14 ± 0.24	1.43 ± 0.45	1.97 ± 0.45	1.37 ± 0.36
Kernel RBF	1.15 ± 0.24	1.43 ± 0.45	1.97 ± 0.49	1.47 ± 0.46
TAGM	**1.09 ± 0.23**	**1.40 ± 0.46**	**1.93 ± 0.41**	**1.35 ± 0.34**

Table 2: Performance in the prediction of the next time point values in terms of MAE (below table).

this approach is that it assumes that all the observations are *i.i.d.* and, therefore, that causal relations do not change in time. TAGM, on the other hand, provides more flexibility and interpretability in this matter as to each of the K state is associated a different causal pattern given by the inferred precision matrix Θ^k. By looking at Equation (7), we can observe that the regression coefficients are given by $W = (\Theta^h_{\mathbf{xy}})^\top (\Theta^h_{\mathbf{xx}})^{-1}$. Thus, for each variable $\mathbf{y}[j]$ for $j = 1, \ldots, D$ the features that are causal for it are given by the coefficients in the j-th column of W. The causality of this is simply implied by the sequentiality of the data. Under a different perspective, our predictive model (Equation (7)) can be seen as a solution of an ordinary differential equation that models mass-action kinetics as specified in [40], Equation (2). Thus, TAGM allows us to detect possibly K multivariate non-stationary causality patterns in any input time-series.

5 USE CASE: STOCK PRICES

TAGM can be exploited to analyse stock market prices. In particular we consider the tasks of building an investment portfolio as well as forecasting of tomorrow stock values.

Building an investment portfolio. Ideally, a portfolio consists in set of stocks on which one invest. The best portfolio is one that provides the highest possible profit while maintaining a low fixed risk level. *Stock picking* (*i.e.*, the selection of the best stocks to put in the portfolio) is a hard task, indeed, even if you restrict to a given industrial sector, there are many factors that can cause underlying variations in the market. Moreover, stocks may be dependent on each other in a way that is often difficult to disentangle. The ability to detect and understand stock dependencies as well as changes in the market would allow to perform the best *hedging* strategy. Hedging is the process of investing in contrary or opposite sectors in order to balance against a possible loss.

Nowadays, the study of stock dependencies is performed by fixing a temporal window and inferring the related empirical covariance matrix. Such method does not account for possible underlying changes in the distribution in that window, and, moreover, it requires to fix an arbitrary cut-off in the length of the analysed time series. TAGM, on the other hand, solves both problems as it could be applied as an exploratory step on the complete time series, while automatically detecting when the underlying distribution changes possibly due to environmental or political conditions. Simultaneously, it provides a cleaner view on the dependencies than the empirical covariance matrix as the GGMs graphs remove spurious dependencies among stocks.

We performed a small experiment by considering three securities (*i.e.* tradable financial assets): Petrobras (PETR4), WTI crude Oil front futures, and exchange from US dollar to Brazilian Real (USD/BRL). We considered the period from 12/01/2010 to 15/09/2016, corresponding to 1635 trading days which have many price swings (up and down) (see Figure 7 leftmost panel). We trained our model on the first 1470 days and we tested on the last 165 (from 20/01/2016 to 15/09/2016). Such securities have deep investment connections and the goal is to find a combination of weights (*i.e.*, amount of invested money) for each security which allows to earn a positive return in the long run and being backed from big losses in case of price oscillations (*i.e.* keeping a fixed risk). According to the Markowitz mean-variance portfolio optimization theory the two quantities of interests are: the expected value of returns and their covariance [5]. As these two quantities vary, the portfolio weights should vary accordingly.

To this end, each day starting from 20/01/2016, we fit TAGM on all the previous observations of the time series. The inferred GGM at the current date is used as weights for the securities. Note that, we adjust such weights if and only if there has been a change in the underlying distribution (*i.e.* the hidden state of the HMM) otherwise we keep the weights fixed to the previous day values. We compared the performance of our approach against the common state-of-the-art method of taking the last 50 days covariances. The performances are given in terms of profits and losses (P&L) and in the evaluation we suppose for simplicity that there are no trading fees. Results are shown in Figure 7 central panel where it is possible to see how TAGM greatly outperforms the empirical covariance approach going from a -20% to a 80% profit. This is due to the changes in trend of the WTI Crude Oil security, that are captured by TAGM but not by the mean empirical covariance strategy.

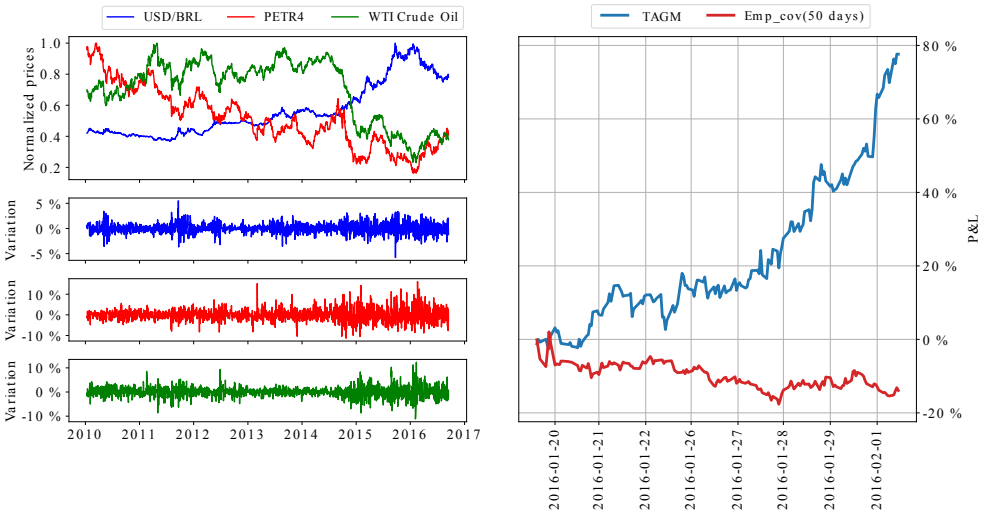
Stock values forecasting. To test TAGM regression performance on real data we considered a period of 30 trading days from 4/08/2016 to 15/09/2016. We compare our results with the same state-of-the-art regression methods used in synthetic experiments. Figure 7 right panel shows that the performance of all the considered methods are very closed to each other, with TAGM slightly improving overall. This is due to the fact that past price values are not very informative to predict the next values since their short-run movements depend on the real time news and market sentiment. Therefore, as long as we do not introduce this information in the prediction it would be difficult to evaluate the prediction performance as all methods catch the same information (typically just noise).

6 RELATED WORK

We position TAGM against various state-of-the-art approaches that perform clustering, temporal network inference, forecasting and causality analysis. All these approaches perform these tasks in isolation whereas TAGM can be used to simultaneously perform all four.

The combination of inferring a latent representation as clustering coupled with GGMs was presented in [14, 31], where the authors combined GGMs with Gaussian Mixture Models. Note that, such approach does not explicitly account for sequential data.

In the context of graph inference, TAGM can be seen as a generalisation of current state-of-the-art methods for temporal graphical



	MAE
TAGM	**0.459± 0.371**
Kernel RBF	0.462 ± 0.380
LSTM	0.461 ± 0.412
VAR	0.460 ± 0.386
Lgb	0.459 ± 0.396

Figure 7: TAGM enables significantly improvement in the construction of a financial portfolio (central panel) due to its ability to promptly detect changes in underlying dependencies among stocks (data showed on the left). TAGM shows better performance also in the prediction of next day stock prices (right panel).

models inference. Such methods typically rely on the assumption that the time points within a chunk, which size is arbitrarily chosen, are *i.i.d.* [16, 21]. Our approach relaxes such assumption making it more reliable for real-world analysis. We can find methods that perform simultaneous graph inference and clustering [22, 46]. Nonetheless, being based on the assumption of chunks they are impossible to compare directly with the presented model. All the previous methods have to rely on the imposition of norms to consider sequentiality. We avoid such imposition by relying on the Markov chain of the hidden states. In literature, we also find papers that look at single time points assuming local topological changes [7, 23, 35, 50] but they do not provide a way to directly perform clustering of the inferred networks.

For prediction tasks, we found in literature two examples that explicitly consider non-stationarity and sequentiality in a setting similar to ours. In [6] the authors use a vector autoregressive model (VAR) on time-dependent splines while in [35] they infer a dynamic graphical model and they predict the topology of the next time point. These last two methods, while allowing for next time point prediction, do not directly allow us to estimate the underlying precision matrix. Another interesting relationship of TAGM is with multivariate gaussian process regression [9], which makes explicit use of the conditional dependencies to estimate future time points. We want to point out that many methods that perform prediction on time series exist. We do not explicitly report them as their integration with GGMs is not obvious.

Lastly, differently from our setting, causality is often studied assuming stationarity of the time series, thus causal relationships are inferred as constant in time [12, 25, 30, 40, 44, 47]. In literature, we can find research directions that consider non-stationarity [24, 38, 45], but, to the best of our knowledge, the explicit use of dynamic graphical models to this aim is not present in literature.

7 CONCLUSIONS AND FUTURE DIRECTIONS

We present a novel methodology to perform data-mining, forecasting and understanding causality patterns on multi-variate time-series. Our method combines HMMs and GGMs, providing a way to simultaneously cluster non-stationary time-series into stationary sub-groups and for each cluster detecting probability relationships among variables through graphical model inference. This simultaneous inference is suitable to be transformed into a time-varying regression model that allows to make predictions on non-stationary time-series. Moreover, the regression coefficients can be interpreted as causal patterns. Our method generalizes many state-of-the-art methods and provides a wide range of analysis type to be performed on time series. Both synthetic and real experiments show that it does indeed outperform state-of-the-art method for clustering, network inference and prediction.

There are many improvements that could be performed. One could add flexibility in the detection of each observation state by making the transition probabilities (the matrix A) time-dependent [26]. Also, using a non-parametric Bayesian approach would allow us to transform TAGM into an infinite-state model [4] thus removing the problem of identifying the most suitable hyper-parameter K (the number of states). Moreover, TAGM could benefit from convergence analysis and faster optimization techniques as it requires a high computational time when dealing with long time-series as the inference of the Markov chain cannot be easily parallelized. Two future interesting directions could be to relax the assumption of Gaussian distributed data and thus, by changing the emission probabilities to graphical models that allow for other type of distributions (*e.g.*, Poisson, Binomial, or others) [1, 23, 27, 49]. Moreover, if one is interested just in exploiting graphical models to study causality, we want to point out an interesting resemblance between Equation (2) in [40] that models system kinetics and Equation (8) in [48] that define a pairwise graphical model on a general exponential family distribution. To conclude, the urge to dissect the underlying system

observed through time series has led current research to deeply rely on graphical models. The approach we presented in this paper reinforces the general understanding that, indeed, graphical models are a powerful tool to study time series under a variety of different perspectives.

REFERENCES

[1] Genevera I Allen and Zhandong Liu. A local poisson graphical model for inferring networks from sequencing data. *IEEE transactions on nanobioscience*, 12(3):189–198, 2013.

[2] Leonard E Baum and Ted Petrie. Statistical inference for probabilistic functions of finite state Markov chains. *The Annals of Mathematical Statistics*, 37(6):1554–1563, 1966.

[3] Leonard E Baum, Ted Petrie, George Soules, and Norman Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *The Annals of Mathematical Statistics*, 41(1):164–171, 1970.

[4] Matthew J Beal, Zoubin Ghahramani, and Carl E Rasmussen. The infinite hidden Markov model. In *Advances in neural information processing systems*, pages 577–584, 2002.

[5] John L. G. Board, Charles M. S. Sutcliffe, and William T. Ziemba. *Portfolio Selection: Markowitz Mean-Variance Model*, pages 1992–1998. Springer US, Boston, MA, 2001.

[6] Laura F Bringmann, Ellen L Hamaker, Daniel E Vigo, André Aubert, Denny Borsboom, and Francis Tuerlinckx. Changing dynamics: Time-varying autoregressive models using generalized additive modeling. *Psychological methods*, 22(3):409, 2017.

[7] Andersen Chang, Tianyi Yao, and Genevera I Allen. Graphical models and dynamic latent factors for modeling functional brain connectivity. In *2019 IEEE Data Science Workshop (DSW)*, pages 57–63. IEEE, 2019.

[8] Kai Chen, Yi Zhou, and Fangyan Dai. A LSTM-based method for stock returns prediction: A case study of China stock market. In *2015 IEEE international conference on big data (big data)*, pages 2823–2824. IEEE, 2015.

[9] Zexun Chen, Bo Wang, and Alexander N Gorban. Multivariate gaussian and student-t process regression for multi-output prediction. *Neural Computing and Applications*, 32(8):3005–3028, 2020.

[10] Tiberiu Chis and Peter G Harrison. Adapting hidden Markov models for online learning. *Electronic Notes in Theoretical Computer Science*, 318:109–127, 2015.

[11] Federico Ciech and Veronica Tozzo. Time adaptive gaussian model, 2021.

[12] Rainer Dahlhaus and Michael Eichler. Causality and graphical models in time series analysis. *Oxford Statistical Science Series*, pages 115–137, 2003.

[13] Alexandre d'Aspremont. Identifying small mean-reverting portfolios. *Quantitative Finance*, 11(3):351–364, 2011.

[14] Brian S Everitt. Finite mixture distributions. *Wiley StatsRef: Statistics Reference Online*, 2014.

[15] G David Forney. The Viterbi algorithm. *Proceedings of the IEEE*, 61(3):268–278, 1973.

[16] Nicholas J Foti, Rahul Nadkarni, AK Lee, and Emily B Fox. Sparse plus low-rank graphical models of time series for functional connectivity in meg. In *2nd KDD Workshop on Mining and Learning from Time Series*, 2016.

[17] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics New York, 2001.

[18] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, 2008.

[19] Clive WJ Granger. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica: journal of the Econometric Society*, pages 424–438, 1969.

[20] Uri Hadar et al. High-order hidden Markov models-estimation and implementation. In *2009 IEEE/SP 15th Workshop on Statistical Signal Processing*, pages 249–252. IEEE, 2009.

[21] David Hallac, Youngsuk Park, Stephen Boyd, and Jure Leskovec. Network inference via the time-varying graphical lasso. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 205–213. ACM, 2017.

[22] David Hallac, Sagar Vare, Stephen Boyd, and Jure Leskovec. Toeplitz inverse covariance-based clustering of multivariate time series data. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 215–223. ACM, 2017.

[23] Jonas MB Haslbeck and Lourens J Waldorp. mgm: Structure estimation for time-varying mixed graphical models in high-dimensional data. *arXiv preprint arXiv:1510.06871*, 2015.

[24] Zonglu He and Koichi Maekawa. On spurious granger causality. *Economics Letters*, 73(3):307–313, 2001.

[25] Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural computation*, 9(8):1735–1780, 1997.

[26] Mian Huang, Yue Huang, and Kang He. Estimation and testing nonhomogeneity of Hidden Markov model with application in financial time series. *Statistics and Its Interface*, 12:215–225, 01 2019.

[27] Ali Jalali, Pradeep Ravikumar, Vishvas Vasuki, and Sujay Sanghavi. On learning discrete graphical models using group-sparse regularization. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 378–387, 2011.

[28] Tom Dupré La Tour, Thomas Moreau, Mainak Jas, and Alexandre Gramfort. Multivariate convolutional sparse coding for electromagnetic brain signals. In *Advances in Neural Information Processing Systems*, pages 3292–3302, 2018.

[29] Steffen L Lauritzen. *Graphical models*, volume 17. Clarendon Press, 1996.

[30] F. Li, L. Zhang, B. Chen, D. Gao, Y. Cheng, X. Zhang, Y. Yang, K. Gao, Z. Huang, and J. Peng. A Light Gradient Boosting Machine for Remainning Useful Life Estimation of Aircraft Engines. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 3562–3567, 2018.

[31] Anani Lotsi and Ernst Wit. High dimensional sparse gaussian graphical mixture model. *arXiv preprint arXiv:1308.3381*, 2013.

[32] James MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA, 1967.

[33] Brian W Matthews. Comparison of the predicted and observed secondary structure of T4 phage lysozyme. *Biochimica et Biophysica Acta (BBA)-Protein Structure*, 405(2):442–451, 1975.

[34] Nicolai Meinshausen, Peter Bühlmann, et al. High-dimensional graphs and variable selection with the lasso. *The annals of statistics*, 34(3):1436–1462, 2006.

[35] Alberto Natali, Mario Coutino, Elvin Isufi, and Geert Leus. Online time-varying topology identification via prediction-correction algorithms, 2020.

[36] Andrew Y Ng, Michael I Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in neural information processing systems*, pages 849–856, 2002.

[37] Maria Rosa Nieto and Esther Ruiz. Frontiers in VAR forecasting and backtesting. *International Journal of Forecasting*, 32(2):475–501, 2016.

[38] Angeliki Papana, Catherine Kyrtsou, Dimitris Kugiumtzis, and Cees Diks. Detecting causality in non-stationary time series using partial symbolic transfer entropy: Evidence in financial data. *Computational economics*, 47(3):341–365, 2016.

[39] Steven Peterson. *Investment Theory and Risk Management,+ Website*, volume 711. John Wiley & Sons, 2012.

[40] Niklas Pfister, Stefan Bauer, and Jonas Peters. Learning stable and predictive structures in kinetic systems. *Proceedings of the National Academy of Sciences*, 116(51):25405–25411, 2019.

[41] Sara Rebagliati and Emanuela Sasso. Pattern recognition using hidden Markov models in financial time series. *Acta et Commentationes Universitatis Tartuensis de Mathematica*, 21(1):25–41, 2017.

[42] Andrew Rosenberg and Julia Hirschberg. V-measure: A conditional entropy-based external cluster evaluation measure. In *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)*, pages 410–420, 2007.

[43] Gideon Schwarz. Estimating the dimension of a model. *Ann. Statist.*, 6(2):461–464, 03 1978.

[44] Christopher A Sims. Macroeconomics and reality. *Econometrica: journal of the Econometric Society*, pages 1–48, 1980.

[45] Linda Sommerlade, Marco Thiel, Bettina Platt, Andrea Plano, Gernot Riedel, Celso Grebogi, Jens Timmer, and Björn Schelter. Inference of granger causal time-dependent influences in noisy multivariate time series. *Journal of neuroscience methods*, 203(1):173–185, 2012.

[46] Federico Tomasi, Veronica Tozzo, and Annalisa Barla. Temporal pattern detection in time-varying graphical models. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 4481–4488. IEEE, 2021.

[47] Jean-Philippe Vert, Koji Tsuda, and Bernhard Schölkopf. A primer on kernel methods. *Kernel methods in computational biology*, 47:35–70, 2004.

[48] Eunho Yang, Genevera Allen, Zhandong Liu, and Pradeep K. Ravikumar. Graphical models via generalized linear models. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1358–1366. Curran Associates, Inc., 2012.

[49] Eunho Yang, Pradeep Ravikumar, Genevera I Allen, and Zhandong Liu. Graphical models via univariate exponential family distributions. *The Journal of Machine Learning Research*, 16(1):3813–3847, 2015.

[50] Jilei Yang and Jie Peng. Estimating time-varying graphical models, 2018.

[51] Ming Yuan. Discussion: Latent variable graphical model selection via convex optimization. *The Annals of Statistics*, 40(4):1968–1972, 2012.

[52] Yingjian Zhang. *Prediction of financial time series with Hidden Markov Models*. PhD thesis, Applied Sciences: School of Computing Science, 2004.

A SYNTHETIC DATASET GENERATION

We generated data through a Markov process which controls the probability to remain in the same state or to go from one state to another one.

The synthetic data generation comprehend the following steps:

(1) we fix suitable values for the number of observations N, the number of states K and the number of multivariate time series D.
(2) for every state $k = 1, \dots, K$, we allow for several combinations of distributions to generate the observations.
   (a) The mean can be drawn in two ways: from a multivariate normal distribution$\mu_k \sim \mathcal{N}(\mathbf{0}, \mathcal{I})$, where $\mathcal{I}$ is the identity matrix, or from an *uniform distribution* with $\mu_k \sim \mathcal{U}(a, b)$ and $a, b \in \mathbb{R}$ with $a < b$. If $a \ll b$ then the generated cluster are more likely to be separated between each other.
   (b) The covariance matrix Σ_k can be set in three ways: fixing a certain maximum degree for each node d, we randomly selected its neighbours and put deterministically the weights of the edges to $0.98/d$ to ensure positive definiteness of the resulting precision matrix [34, 51]; from the tool **scikit-learn.datasets** which generates a random symmetric, positive-definite matrix; from the precision matrix stressing the links between nodes, starting from the identity matrix and putting randomly ones in the off-diagonal places respecting the symmetric matrix constraint. In this way we are generating precision matrix with either strong links between nodes or no links at all. This case is interesting because in this way the networks corresponding to each state k are very different between each others like the case with means very far away.
(3) each row of the transition matrix A is generate from a Dirichlet distribution $\mathrm{Dir}(\alpha)$ where $\alpha \in \mathbb{R}_+^K$. In particular, to not have too quick transitions from one state to another we impose $\alpha_i = \kappa \cdot \alpha_j$ with $i \neq j$ where i is the index of the row transition we are drawing and α_j all the other element of α different than α_i. κ is also known as the *force* constant, in the sense that the bigger κ is the more likely the state i is respect to the others;
(4) finally, for each time point n, the state k is drawn from the transition matrix A then the data are drawn from the related normal distribution $\mathbf{X}_n \sim \mathcal{N}(\mu_k, \Sigma_k)$.

B MODEL SELECTION

Our model has two hyper-parameters to cross-validate:

(1) the number of finite states K;
(2) the regularization parameter λ which regulates the sparsity of the precision matrix Θ_k;

To estimate these two hyper-parameters we employ cross validation (CV) with a Bayesian Information Criterion (BIC) score. To determine the number of hidden states we use the BIC approach [43] which has the form

$$\mathrm{BIC}(m) = \ln \mathbf{p}(\mathbf{X}|m, \theta) - \frac{\nu}{2} \ln(n).$$

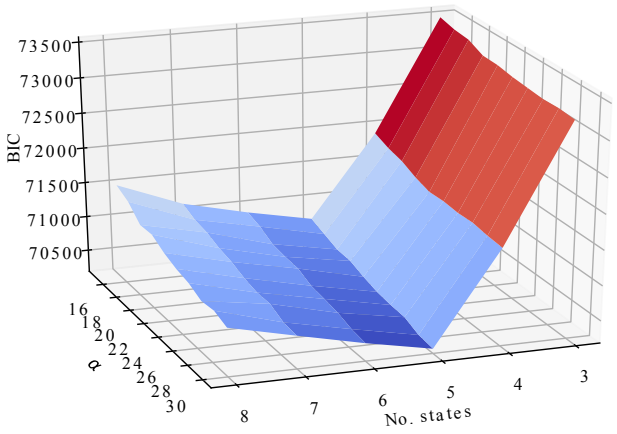


Figure 8: Cross validation

ν represents the number of free parameters and m the considered model. In our case the number of free parameters can be computed in the following:

- the probabilities π have dimension K with one constraint, so $\nu_\pi = K - 1$;
- the transition matrix A has dimension $K \times K$ but each row has a constraint, so $\nu_A = K(K-1)$;
- the means μ are K with dimension d without any constraint, so $\nu_\mu = KD$;
- the precision matrices Θ are K, one for each state, with dimension $d \times d$ but they have the constraint given by graphical lasso therefore $\nu_\Theta = \sum_{i \geq j} e_{i,j}$ where $e_{i,j} = 0$ if $\hat{\Theta}_{i,j} = 0$ and $e_{i,j} = 1$ otherwise. $\hat{\Theta}$ is the estimated precision matrix.

Putting all together the total number of free parameter ν is

$$\nu = \nu_\pi + \nu_A + \nu_\mu + \nu_\Theta = (K-1)(K+1) + KD + \sum_{k=1}^{K} \nu_{\Theta_k}.$$

To see that this CV combination of methods is suitable for the estimation of the hyper-parameters of our model we generated a multivariate time series with $D = 10$ and $K = 5$ and we cross-validate K and λ from the sets $K \in \{3, \dots, 8\}$ and $\lambda \in [18, 25]$. We show in Figure 8 the results and as we can see it found the K from which we have generated the data.

C INITIALIZATION CHOICES

The optimization algorithm requires an initialization of the initial parameters θ. Since the likelihood function we are considering is non-convex the parameters initialization is crucial to find the optimal solution. In particular,given the cluster number K we have to initialize four parameters: the transition matrix A and the initial probabilities π and the Gaussian distribution parameters $\boldsymbol{\Theta}$ and μ. In our implementation we adopt the following initializations choices:

- **the transition matrix A and the initial probabilities π** can either be initialised with equal probabilities for each state $\frac{1}{K}$ or by randomly sampling from a uniform distribution

$\mathcal{U}(0,1)$ or symmetric Dirichlet distribution $\mathrm{Dir}(1)$, with the constraints that each row has to sum to one;

- **the Gaussian distribution parameters Θ and** μ are initialized by computing an initial subdivision of the time points into clusters. To this aim with used K-means or Gaussian mixture model (GMM). Both GMM and K-means are non-convex, thus, depending on initialisation lead to different solutions as well. Given the dataset initial subdivision, we compute respectively the empirical covariances and means. Finally we run the graphical lasso to compute the corresponding precision matrix and we use that as initial parameters.

D EVALUATION METRICS

We use a metric score for each of the following aspects:

(1) **clustering performance**: we compare the clustering results in terms of V-measure [42] which returns a value $v \in [0,1]$ where $v = 0$ means that the cluster labels are assigned completely randomly while $v = 1$ means that there is a perfect match between the true labels and the one found by the models.

(2) **network inference performance**: in order to evaluate the performances of the methods we need to identify a map between the true clusters and the identified ones in order to compare the underlying graphs. Such map is obtained by taking the maximum per row of the contingency table of the true and predicted labels. We then consider the true and inferred graphs as binary classes (0 no edge identified, 1 edge identified) and we compute the Matthews correlation coefficient (MCC) [33]

$$\mathrm{MCC} = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$$

which return a value in the interval $[-1,1]$ where 0 corresponds to chance.

(3) **forecasting performance**: we used as score the Mean Absolute Error (MAE) which measures the error between the true next point value and the predicted one. Since we are predicting d values for each future time point we compute the mean MAE across entries of the vector:

$$MAE = \frac{1}{n}\sum_{i=1}^{N}\left(\frac{1}{D}\sum_{d=1}^{D}|x_{nd} - \hat{x}_{nd}|\right)$$

.