Relevant States and Memory in Markov Chain Bootstrapping and Simulation

Cerqueti, Roy and Falbo, Paolo and Pelizzari, Cristian

University of Macerata, Italy, University of Brescia, Italy, University of Brescia, Italy

March 2010

Relevant States and Memory in Markov Chain Bootstrapping and Simulation

Roy Cerqueti

Università degli Studi di Macerata - Dipartimento di Istituzioni Economiche e Finanziarie

Via Crescimbeni, 20 - 62100 Macerata MC - Italy. E-mail: roy.cerqueti@unimc.it

Paolo Falbo

Università degli Studi di Brescia - Dipartimento Metodi Quantitativi

Contrada Santa Chiara, 50 - 25122 Brescia BS - Italy. E-mail: falbo@eco.unibs.it.

Cristian Pelizzari

Università degli Studi di Brescia - Dipartimento Metodi Quantitativi

Contrada Santa Chiara, 50 - 25122 Brescia BS - Italy

Corresponding author. Tel.: +39 030 2988516. Fax: +39 030 2400925. E-mail: pelizcri@eco.unibs.it.

Abstract

Markov chain theory is proving to be a powerful approach to bootstrap highly nonlinear time series. In this work we provide a method to estimate the memory of a Markov chain (i.e. its order) and to identify its relevant states. In particular the choice of memory lags and the aggregation of irrelevant states are obtained by looking for regularities in the transition probabilities. Our approach is based on an optimization model. More specifically we consider two competing objectives that a researcher will in general pursue when dealing with bootstrapping: preserving the "structural" similarity between the original and the simulated series and assuring a controlled diversification of the latter. A discussion based on information theory is developed to define the desirable properties for such optimal criteria. Two numerical tests are developed to verify the effectiveness of the method proposed here.

MSC classification: 60J10, 60J20, 60J22, 62B10, 62F40, 91G60.

Keywords: Bootstrapping; Information Theory; Markov chains; Optimization; Simulation.

1 Introduction

In the financial literature, starting from the tests on efficient market hypothesis and the technical analysis (e.g., Brock et al., 1992; Sullivan et al., 1999), bootstrap procedures have been applied intensively to solve a wide variety of problems. Following such a spread interest, several methodological contributions have appeared to improve the initial bootstrap method advanced by Efron (1979), even if the basic idea remains unchanged (e.g., see the methodological discussion on the *classical bootstrap* methods in Freedman, 1984; Freedman and Peters, 1984; Efron and Tibshirani, 1986, 1993). In particular, the heart of the bootstrap consists of resampling some given observations with the purpose of obtaining a good estimation of statistical properties of the original population. However, an important restriction to the classical bootstrap methods is that the original population must be composed of independent identically distributed observations. In the case of time series taken from the real life, this condition is hardly true. When such hypothesis is not true, a theoretical model for the data is required and the bootstrap is then applied to the model errors.

A new group of bootstrapping methods has been advanced to reduce the risk of mis-specifying the model. To this group belong the so called *block*, *sieve*, and the *local* methods of bootstrapping (see Bühlmann, 2002, for a comparison of these methods). The methods are nonparametric, and assume that observations can be (time) dependent.

This category of literature has increased in a relatively recent period of a new method of bootstrapping based on Markov chain theory. The major advantage of this approach is that it is entirely data driven, so that it can smoothly capture the dependence structure of a time series, releasing a researcher from the risk of wrongly specifying the model, and from the difficulties of estimating its parameters.

The limitation connected to Markov chains is of course that they are naturally unsuitable to model discrete-valued processes. This is an unfortunate situation, since several phenomena in many areas of research are often modeled through continuous-valued processes. In the economic and financial literature, there are plenty of cases of continuous-valued processes showing complex behaviors, where observations appear to depend nonlinearly from previous values. It is well known that in the financial markets, next to technological and organizational factors, psychology and emotional contagion introduce complex dynamics in driving the expectations on prices (e.g., think to the terms popular in the technical analysis such as "psychological thresholds", "price supports", "price resistances", etc.). In such cases, guessing the correct model for complex continuous-valued stochastic processes is highly risky.

To overcome this risk, a researcher in the need of bootstrapping or simulating a continuous-valued stochastic process could in principle resort to partitioning its support, obtaining a discretized version of it, and then apply Markov chain bootstrapping or simulation techniques to model brilliantly any arbitrary dependence structure. Such a powerful solution has however a major difficulty, that is how to organize an *efficient partition* of the process support. Indeed in the absence of some guide, fixing arbitrarily a partition excessively refined or raw involves different kinds of drawbacks, ranging from insufficient diversification of the simulated trajectories to unsatisfactory replication of the key features of the stochastic process.

In this paper we develop an original general approach to determine the relevant states and the memory (i.e. the order) of a Markov chain, keeping in mind the major problems connected to applying Markov chain bootstrapping and simulation to continuous-valued processes.

There is a wide literature which has dealt with the analysis of states and memory of a Markov chain for resampling purposes, which we review in the following section. It is quite important to notice that such literature has mainly focused on the estimation of the order of a Markov chain more than it has done to discriminate the relevant states, and this is due to the fact that Markov chains are discrete-valued processes, where the states are usually taken as equally important.

From our perspective, focusing on the relevant states is crucial if we want to consider the discretized versions of complex continuous-valued processes. As mentioned previously, it is frequent in economic and financial markets that some observed states, or combinations of them, are more relevant than others in determining the future evolution of the process. In other words, not all the partitions of the support of a continuous-valued process are suitable to capture the relevant information about its dependence structure. Finding the optimal ones is therefore crucial to apply correctly the methodology of Markov chain bootstrapping and simulation. However bootstrapping requires to take care of an aspect, which we deal with here explicitly and which is not as critical with simulation. In Markov chain bootstrapping the probability to re-generate large portions of the original series is a serious drawback, especially when the number of states and order of the Markov chain increase and transition probabilities get close to unity (the limiting case is the repetition of the entire original series). We deal with this diversification problem in our model.

The approach we propose in this paper is based on the joint estimation of the relevant states and of the order of a Markov chain and consists of an optimization problem. The solution identifies the partition which groups the states with the most similar transition probabilities. In this way the resulting groups emerge as the relevant states, that is the states which significantly influence the conditional distribution properties of the process. Furthermore, as we will show, our approach is information efficient in the sense of Kolmogorov (1965), that is it searches for the partition which minimizes the information loss. Our optimization problem includes also the “multiplicity” constraint which controls for a sufficient diversification of the resampled trajectories.

Our work contributes to the literature on Markov chain bootstrapping in various ways.

Firstly, we develop a method to estimate the parameters of a Markov chain dedicated to bootstrap via constrained optimization. When the threshold defining the multiplicity constraint is let to vary, an efficient frontier obtains, whose properties provide a complete description of the optimal solutions.

Secondly, we propose a non hierarchical approach, which means that a non sequential search of the order of the Markov chain is performed. More precisely, if some states are grouped at a given time lag w, then they are not forced to stay together at farther time lags $w+r$ (with $r>0$). This "freedom" adds flexibility in modeling the dependence structure of a Markov chain and, to our knowledge, our approach is the first in the literature on Markov chain bootstrapping and simulation to abandon hierarchical grouping. Such a feature is not of secondary importance, since it allows to model a Markov chain with non monotonically decreasing memory.

Thirdly, comparing to the bootstrap literature developed in econometrics and applied statistics, our proposal treats states as if they were of qualitative nature, and the search of efficient partitions is based only on transition probabilities. In other words, no distance between the values of the different states is used in the decision of merging them. Again, this approach allows a higher flexibility in the identification of the relevant states and an increased capacity to capture the dynamics of a Markov chain.

Fourthly, this paper provides the theoretical grounds for Markov chain bootstrapping and simulation of continuous-valued processes. To the best of our knowledge, this is the first attempt to extend Markov chain bootstrapping and simulation in this sense. Our search for the relevant states supplies the levels where the process modifies significantly its dynamics (i.e. its expected value, its variance, etc.). Hence, it is designed to minimize the information loss deriving from aggregating the states, so it helps maintaining highly complex nonlinearities of the original process.

Fifthly, we introduce two new non entropic measures of the disorder of a Markov chain process, and we study their main properties.

The paper is organized as follows. Section 2 reviews the relevant literature on Markov chain bootstrapping. Section 3 introduces the settings of the problem. Section 4 discusses some theoretical properties of the criteria used here to select the optimal dimension of a Markov chain transition probability matrix. Section 5 discusses some methodological issues. In Section 6 the criteria are applied to two examples. Section 7 concludes.

2 A Bibliography Review on Markov Chain Bootstrapping

It is possible to group different contributions on resampling procedures based on Markov chain theory.

A first major category is concerned with processes that are not necessarily Markov chains. A series of stationary data is divided into blocks of length l of consecutive observations; bootstrap samples are then generated joining randomly some blocks. The seminal idea appears first in Hall (1985) for spatial data, has been applied to time series by Carlstein (1986), but has been fully developed starting with Künsch (1989) and Liu and Singh (1992). In Hall et al. (1995), Bühlmann and Künsch (1999), Politis and White (2004), and Lahiri et al. (2007), the selection of the parameter l -a crucial point of this method- is driven by the observed data.

Many variants of the block bootstrap method exist by now; standard references include Politis and Romano (1992) for the *blocks-of-blocks bootstrap*, Politis and Romano (1994) for the *stationary bootstrap*, and Paparoditis and Politis (2001a, 2002a) for the *tapered block bootstrap*. For a survey, see Lahiri (2003). Despite the block based bootstrap methods have been developed to get over the problem of dependence disruption, they only partially succeed in their goal. Indeed they pass from the loss of dependency among data to that among blocks.

A second category relies to Markov chains (or processes) with finite states and faces explicitly the problem of maintaining the original data dependency. Earlier approaches to bootstrap Markov chains were advanced by Kulperger and Prakasa Rao (1989), Basawa et al. (1990), and Athreya and Fuh (1992), and have been further investigated in Datta and McCormick (1992). This second group is more closely related to our work, since it focuses on the transition probabilities of a stationary Markov chain (or process), as we also do here. It is useful to distinguish some different approaches. The *sieve (Markov) bootstrap* method was first advanced by Bühlmann (1997); it consists of fitting Markovian models (such as an AR) to a data series and resampling randomly from the residuals. This idea has been further developed in Bühlmann (2002), where the *variable length Markov chain sieve bootstrap* method is advanced. This is an intriguing approach since in nature it happens that only "some" sequences of states (i.e. paths) tend to reappear in an observed sequence more than others and to condition significantly the process evolution. However this method proceeds in a hierarchical way to search for the relevant paths, which can be a severe limitation when time dependence is not monotonically decreasing.

Still in the framework of Markov processes, Rajarshi (1990) and Horowitz (2003) estimate the transition density function of a Markov process using kernel probability estimates. The idea of using kernels is adopted also by Paparoditis and Politis (2001b, 2002b), which advance the so called *local bootstrap* method. This method rests on the assumption that similar trajectories will tend to show similar transition probabilities in the future. However it is not uncommon to observe empirical contradiction to such hypothesis. Besides, the number of time lags to be observed to compare trajectories has to be chosen arbitrarily.

Anatolyev and Vasnev (2002) propose a method (*Markov chain bootstrap*) based on a finite state discrete Markov chain. Similarly to what we do here, the authors partition the state space of the series into I sets (bins). While some interesting estimation properties of the bootstrap method are shown, the bins are formed simply distributing the ordered values evenly in each of them. Besides, an arbitrary number of time lags is also fixed to bound the relevant path length.

The approach called *regenerative (Markov chain) block bootstrap* has been initially developed by

Athreya and Fuh (1992) and Datta and McCormick (1993), and has been further analyzed by Bertail and Clémençon (2006, 2007). This method focuses on a chosen recurring state (atom) and the consecutive observations between departure from and return to the atom (cycle or block). Bootstrapping is then accomplished by sampling at random from the observed cycles. This method reconciles the gap between Markov chain bootstrapping procedures and block bootstrapping, with the important difference that the cutting points (used to form the blocks) in the Markov chain approach are not chosen at random, but are data driven. Besides, it does not need to explicitly estimate the transition probabilities of the observed process. However this relies heavily on the identification of the atom, which is unfortunately unknown.

The problem of estimating the relevant states and the order of a Markov chain process for bootstrapping purposes can also be related to the information theory literature, with particular reference to the data compression analysis. In general terms, data compression problems rely on flows of data generated by a process with a finite alphabet, like a finite state Markov chain. The criteria adopted for estimating the relevant parameters of a finite state process include, for example, the AIC (Akaike Information Criterion, Akaike, 1970), the BIC (Bayesian Information Criterion, Schwarz, 1978), and the MDL principle (Minimum Description Length principle, Rissanen, 1978). Each criterion consists of two parts: an entropy-based functional and a penalty term depending on the number of parameters, both to be minimized.

The link between bootstrapping and data compression analysis can be stated as follows. As already stressed above, a key point in bootstrap problems consists in generating simulated series keeping the relevant statistical properties of the original one and avoiding the risk of exactly replicating the original series. Under the data compression theory point of view, the former aspect can be translated into the minimization of the entropy-type distance, while the latter is formalized through the minimization of the penalty term.

In this respect, we estimate the relevant parameters of a Markov chain process for bootstrapping purposes via a constrained optimization problem. Rather than entropy, two specific distances based on the transition probabilities are introduced and minimized. The introduction of non entropic measures is based on three reasons: first of all, there is no consensus on a preferable entropy measure among the several available (Ullah, 1996; Cha, 2007); secondly, as we will see, our distance indicators are close to usual dispersion measures, analytically simple, and we could prove easily for them the minimal properties required to disorder measures discussed in Kolmogorov (1965); lastly, the introduction of two new measures is an extension of the literature on information theory. A constraint is also introduced which corresponds to minimizing the penalty term.

Starting from Rissanen (1978), Rissanen (1983), Rissanen and Langdon Jr. (1981), and Barron et al. (1998), which first showed the strict link between coding and estimation, literature on data compression has indeed developed in the direction of model estimation.

Of particular interest for us are Rissanen (1986), Ziv and Merhav (1992), Weinberger et al. (1992), Feder et al. (1992), Liu and Narayan (1994), and Weinberger et al. (1995). These works study the class of finite-state sources and, among other results, develop methods for estimating their states; an important example of a finite-state source is a Markov chain with variable memory, also called *variable length Markov chain* (*VLMC*) (see Bühlmann and Wyner, 1999; Bühlmann, 2002). As its name suggests, a *VLMC* is characterized by a variable order depending on which state verifies at past time lags. Starting from time lag 1, states are differentiated only if they contribute to differentiate future evolution, otherwise they are lumped together. Farther time lags are considered only for those states showing additional prediction power. In the end, such approach identifies a Markov model whose memory changes depending on the trajectory followed by the process. This approach proves to be computationally efficient, as it allows a strong synthesis of the state space. As a further application, the method can be used to develop a bootstrap engine (*VLMC bootstrap*), which is more user-friendly and attractive than the block bootstrap (Künsch, 1989). Bühlmann and Wyner (1999) and Bühlmann (2002) are strongly related to our work, as the reduction to a minimal state space is also an objective of the present study. The main difference in our proposal consists of a non hierarchical selection of the relevant time lags, in the sense that we do not condition the relevance of farther time lags to depend on that of the closer ones.

Merhav et al. (1989), Finesso (1992), Kieffer (1993), Liu and Narayan (1994), Csiszár and Shields (2000), Csiszár (2002), Morvai and Weiss (2005), Peres and Shields (2008), and Chambaz et al. (2009) consider the problem of the estimation of the order of a Markov chain, assuming that the states are all relevant at all the time lags up to the estimated order. However, in some applications a satisfactory estimation of the relevant states is even more important than a precise estimation of the "memory" of the process. We refer, for example, to the bootstrapping of series with regimes characterizing the dynamics of different processes in economics and finance.

3 The model

Let us consider an evolutive observable phenomenon, either continuous or discrete. We suppose that we observe N realizations homogeneously spaced in time and we introduce the set of the time-ordered observations of the phenomenon, $E = \{y_1, \dots, y_N\}$. The $y_1, \dots, y_N$ are understood as the values of a discrete process or as the labels of a discretized continuous process. There exist $J_N \geq 1$ distinct states $a_1, \dots, a_{J_N} \in E$. The corresponding subsets of E, denoted as $E_1, \dots, E_{J_N}$, and defined as:

$$E_z = \{y_i \in E \,|\, y_i = a_z\}, \qquad z = 1, \dots, J_N, \, i = 1, ..., N$$

constitute a partition of E. Moreover, fixed $z = 1, \dots, J_N$, then the frequency of state a_z in the observed series E is the cardinality of E_z. Let $A = \{a_1, \dots, a_{J_N}\}$ be the range of the observed series.

We now consider a time-homogeneous Markov chain of order $k \geq 1$, denoted as $\{X(t), t \geq 0\}$, with state space A. To ease the notation, in the following we will simply write Markov chain instead of time-homogeneous Markov chain. The k-lag memory of the Markov chain implies that the transition probability matrix should account for conditioning to trajectories of length k. Therefore, we refer hereafter to a *k-path transition probability matrix*.

We deal in our paper with a couple of questions related to finding the Markov chain which best describes the observed series E:

- Which is the optimal k?
- Which is the optimal clustering of A for each time lag w, with $w = 1, ..., k$?

It is important to notice that, though the second question focuses primarily on the search of the relevant states, it actually also addresses the analysis of the memory of a Markov chain. Indeed if the optimal clustering at time lag w returns many or just a few classes, we obtain an information about the relevance of that time lag. Few or no classes will in general signal low or no conditioning power. On the contrary the presence of many classes will signal higher relevance. Since the clustering is operated independently for each time lag, this approach can return a distribution of the relevance of the memory of a Markov chain over all the time lags, which need not to be in decreasing order from 1 to k. We introduce a measure of relevance, or "activity", for a time lag later in Section 5 (Methodological issues).

Let us consider $a_z \in A$ and $\boldsymbol{a}_h = (a_{h,k}, ..., a_{h,1}) \in A^k$. The row vector $\boldsymbol{a}_h$ is the ordered set of k states $a_{h,w} \in A$, $w = 1, ..., k$, listed, in a natural way, from the furthest to the closest realization of the chain. This ordering of the realizations will be maintained throughout the paper. The Markov chain has stationary probabilities:

$$P(\boldsymbol{a}_h) = P(X(t) = a_{h,1}, \ldots, X(t-k+1) = a_{h,k}), \tag{1}$$

and transition probability from $\boldsymbol{a}_h$ to state a_z:

$$P(a_z|\boldsymbol{a}_h) = P(X(t) = a_z|X(t-1) = a_{h,1}, \ldots, X(t-k) = a_{h,k}). \tag{2}$$

According to Ching et al. (2008), we estimate the transition probability $P(a_z|\boldsymbol{a}_h)$ by using the empirical frequencies $f(a_z|\boldsymbol{a}_h)$ related to the phenomenon. For the sake of simplicity, we avoid introducing throughout the paper a specific notation for the estimates of the probabilities, therefore we estimate $P(a_z|\boldsymbol{a}_h)$ by

$$P(a_z|\boldsymbol{a}_h) = \begin{cases} \frac{f(a_z|\boldsymbol{a}_h)}{\sum_{j:a_j \in A} f(a_j|\boldsymbol{a}_h)}, & \text{if } \sum_{j:a_j \in A} f(a_j|\boldsymbol{a}_h) \neq 0 \\ 0, & \text{otherwise} \end{cases}. \tag{3}$$

Analogously, $P(\boldsymbol{a}_h)$ is estimated by

$$P(\boldsymbol{a}_h) = \frac{\sum_{j:a_j \in A} f(a_j|\boldsymbol{a}_h)}{\sum_{b:\boldsymbol{a}_b \in A^k} \sum_{j:a_j \in A} f(a_j|\boldsymbol{a}_b)}.$$

The k-path transition probability matrix of $\{X(t), t \geq 0\}$, which is defined by the quantities in (2), is estimated by the quantities in (3).

Let us now introduce the set Λ of the partitions of A. A generic element $\lambda \in \Lambda$ can be written as $\lambda = \{A_1, \ldots, A_{|\lambda|}\}$, where $|\lambda|$ is the cardinality of λ, with $1 \leq |\lambda| \leq J_N$, and $\{A_q\}_{q=1,\ldots,|\lambda|}$ is a partition of nonempty subsets of A. The cardinality of Λ is $B(J_N)$, i.e. the Bell number[1] of the J_N elements in set A.

Extending our notation to a multidimensional context, we consider the set $\boldsymbol{\Lambda}_k$ of k-dimensional partitions. The set $\boldsymbol{\Lambda}_k$ contains the partitions we will focus on in the present paper. A k-dimensional partition of $\boldsymbol{\Lambda}_k$ is denoted as $\boldsymbol{\lambda}$ and is defined as

$$\boldsymbol{\lambda} = \{A_{q_k,k} \times \cdots \times A_{q_w,w} \times \cdots \times A_{q_1,1} |\, q_w \in \{1, \ldots, |\lambda_w|\},\, w = 1, \ldots, k\}\,, \tag{4}$$

where $A_{q_w,w}$ is a generic class of partition λ_w and λ_w is a partition of A at time lag w.

A k-dimensional partition of $\boldsymbol{\Lambda}_k$ can also be (more easily) represented by the k-tuple of partitions λ_w, $w = 1, ..., k$, which the classes $A_{q_w,w}$ belong to. So partition $\boldsymbol{\lambda}$ can also be identified with the following notation:

$$\boldsymbol{\lambda} = (\lambda_k, \ldots, \lambda_w, \ldots, \lambda_1).$$

Such notation describes the fact that $\boldsymbol{\lambda}$ is a time-dependent partition of A, i.e. A is partitioned in different ways for each time lag w, $w = 1, ..., k$.

The cardinality of $\boldsymbol{\Lambda}_k$ is $[B(J_N)]^k$.

The cardinality of partition $\boldsymbol{\lambda}$ is:

$$|\boldsymbol{\lambda}| = \prod_{w=1}^{k} |\lambda_w|\,.$$

We refer to the probability law P introduced in (2) and define

$$P(a_z|\boldsymbol{A}_q) = P(X(t) = a_z|X(t-1) \in A_{q_1,1}, \ldots, X(t-k) \in A_{q_k,k}), \tag{5}$$

where

$$\boldsymbol{A}_q = A_{q_k,k} \times \cdots \times A_{q_w,w} \times \cdots \times A_{q_1,1} \subseteq A^k, \tag{6}$$

[1] The following holds:

$$B(J_N) = \sum_{z=1}^{J_N} S(J_N, z),$$

where $S(J_N, z)$, $z = 1, ..., J_N$, denote the Stirling numbers of the second kind. $S(J_N, z)$ indicates the number of ways a set of J_N elements can be partitioned into z nonempty sets. It holds:

$$S(J_N, z) = \sum_{j=1}^{z} (-1)^{z-j} \cdot \frac{j^{J_N - 1}}{(j-1)!(z-j)!}.$$

and $a_z \in A$. The quantity in (5) is the transition probability to reach state a_z at time t after the process has been in the classes $A_{q_k,k}, \dots, A_{q_1,1}$ in the previous k times.

The transition probabilities $P(a_z|\boldsymbol{A}_q)$ in (5) are estimated, as usual, through the empirical frequencies:

$$P(a_z|\boldsymbol{A}_q) = \begin{cases} \frac{\sum_{i:\boldsymbol{a}_i \in \boldsymbol{A}_q} f(a_z|\boldsymbol{a}_i)}{\sum_{i:\boldsymbol{a}_i \in \boldsymbol{A}_q} \sum_{j:a_j \in A} f(a_j|\boldsymbol{a}_i)}, & \text{if } \sum_{i:\boldsymbol{a}_i \in \boldsymbol{A}_q} \sum_{j:a_j \in A} f(a_j|\boldsymbol{a}_i) \neq 0 \\ 0, & \text{otherwise} \end{cases} .$$

The quantities $P(a_z|\boldsymbol{A}_q)$ estimate a new transition probability matrix. To keep the notation as simple as possible, we continue to refer to this matrix as to the k-path transition probability matrix.

3.1 Partition $\boldsymbol{\lambda}$ and k-path transition probability matrices

It is worth to explore how the k-path transition probability matrix of $\{X(t), t \geq 0\}$ modifies with the lag k and the particular time-dependent clustering of the state space. If we consider a partition $\boldsymbol{\lambda}$, then we will associate to $\boldsymbol{\lambda}$ a k-path transition probability matrix of dimension $|\boldsymbol{\lambda}| \times J_N$. Each row of this matrix corresponds to a class $\boldsymbol{A}_q \in \boldsymbol{\lambda}$ of process paths of length k.

For a sufficiently high k, we can find a partition $\boldsymbol{\lambda}$ removing the randomness of transitions between paths and single states. Indeed, the longer the paths are the more the empirical observation of the phenomenon drives transition probabilities to be trivially equal to 0 or 1. More precisely, each row of the k-path transition probability matrix would consist of 0's, with the (possible) exception of one cell (equal to 1) corresponding to the value that is historically observed after the path (provided that such a value exists). We explain our concern with an example.

Example 1. *Consider a Markov chain $\{X(t), t \geq 0\}$ of order $k \geq 1$, with state space $A = \{1, 2\}$. The process is represented through different k-path transition probability matrices depending on the number of time lags. The transition probabilities are driven empirically by the observation of an evolutive phenomenon. In particular, we assume the following set of time-ordered observations of the phenomenon:*

$$E = \{1, 2, 1, 1, 2, 2, 1\}.$$

To avoid confusing notation, we will denote the k-paths $\boldsymbol{a}_{h,k}$, the partitions $\boldsymbol{\lambda}_k$ and partition classes $\boldsymbol{A}_{q,k}$ of these k-paths and their corresponding transition probability matrices $\mathcal{M}_k$ with a subscript k to distinguish the different values of k used in the present example.

We initially consider two time lags ($k = 2$). The possible process paths $\boldsymbol{a}_{h,2} = (a_{h,2}, a_{h,1}) \in A^2$, $h = 1, ..., 4$, are

$$\boldsymbol{a}_{1,2} = (1, 1),\ \boldsymbol{a}_{2,2} = (1, 2),\ \boldsymbol{a}_{3,2} = (2, 1),\ \boldsymbol{a}_{4,2} = (2, 2).$$

We denote with $\mathcal{M}_2^s$ the 2-path transition probability matrix of the Markov chain related to the observed phenomenon. $\mathcal{M}_2^s$ is associated to the partition of singletons, i.e. each class of the partition *collects exactly one* 2*-path:*

$$\boldsymbol{\lambda}_2^s = \{\{\boldsymbol{a}_{1,2}\}, \{\boldsymbol{a}_{2,2}\}, \{\boldsymbol{a}_{3,2}\}, \{\boldsymbol{a}_{4,2}\}\}.$$

The estimation in (3) gives

$\mathcal{M}_2^s =$

	states a_z	
partition classes $\boldsymbol{A}_{q,2}^s$ *of* $\boldsymbol{\lambda}_2^s$	1	2
$\{(1,1)\}$	0	1
$\{(1,2)\}$	0.5	0.5
$\{(2,1)\}$	1	0
$\{(2,2)\}$	1	0

On the contrary, the all-comprehensive set partition $\boldsymbol{\lambda}_2^a$ *is*

$$\boldsymbol{\lambda}_2^a = \{\{\boldsymbol{a}_{1,2}, \boldsymbol{a}_{2,2}, \boldsymbol{a}_{3,2}, \boldsymbol{a}_{4,2}\}\}$$

and the corresponding 2*-path transition probability matrix is*

$\mathcal{M}_2^a =$

	states a_z	
partition classes $\boldsymbol{A}_{q,2}^a$ *of* $\boldsymbol{\lambda}_2^a$	1	2
$\{(1,1),(1,2),(2,1),(2,2)\}$	0.6	0.4

We admit that the all-comprehensive set partition is the one providing less information on the future evolution of the chain. Nevertheless we stress that, since the second row of $\mathcal{M}_2^s$ *does not contain solely* 0*'s, with the possible exception of one* 1*, there is not a partition* $\boldsymbol{\lambda} = (\lambda_2, \lambda_1)$ *of the set* $A^2 = \{1,2\}^2$ *such that the randomness of the transitions is completely removed. The number of time lags* ($k = 2$) *adopted is not large enough.*

To get to "deterministic paths", we therefore extend k *from* 2 *to* 3*: we have* $\boldsymbol{a}_{h,3} = (a_{h,3}, a_{h,2}, a_{h,1}) \in A^3$*,* $h = 1, ..., 8$*. We construct the matrix* $\mathcal{M}_3^s$ *associated to the partition of singletons*

$$\boldsymbol{\lambda}_3^s = \{\{\boldsymbol{a}_{1,3}\}, ..., \{\boldsymbol{a}_{8,3}\}\}$$

as

$\mathcal{M}_3^s =$

	states a_z	
partition classes $\boldsymbol{A}_{q,3}^s$ *of* $\boldsymbol{\lambda}_3^s$	1	2
$\{(1,1,1)\}$	0	0
$\{(1,1,2)\}$	0	1
$\{(1,2,1)\}$	1	0
$\{(1,2,2)\}$	1	0
$\{(2,1,1)\}$	0	1
$\{(2,1,2)\}$	0	0
$\{(2,2,1)\}$	0	0
$\{(2,2,2)\}$	0	0

It is totally evident that the partition of singletons $\boldsymbol{\lambda}_3^s$ removes the randomness of transitions to states 1 *and* 2. *Consider also partition $\boldsymbol{\lambda}^x = (\lambda_3^x, \lambda_2^x, \lambda_1^x)$, with $\lambda_3^x = \{\{1,2\}\}$, $\lambda_2^x = \{\{1\},\{2\}\}$, and $\lambda_1^x = \{\{1,2\}\}$; the partition includes the following multidimensional classes:*

- $\boldsymbol{A}_1^x = \{1,2\} \times \{1\} \times \{1,2\} = \{(1,1,1),(1,1,2),(2,1,1),(2,1,2)\}$,
- $\boldsymbol{A}_2^x = \{1,2\} \times \{2\} \times \{1,2\} = \{(1,2,1),(1,2,2),(2,2,1),(2,2,2)\}$.

Such a partition removes randomness and the corresponding 3*-path transition probability matrix is*

$\mathcal{M}^x =$

	states a_z	
partition classes $\boldsymbol{A}_q^x$ of $\boldsymbol{\lambda}^x$	1	2
$\{(1,1,1),(1,1,2),(2,1,1),(2,1,2)\}$	0	1
$\{(1,2,1),(1,2,2),(2,2,1),(2,2,2)\}$	1	0

Observe that, by extending k from 2 *to* 3, *we find partitions with deterministic evolution. In these cases, starting from an initial path, the evolution of the process continues in a deterministic way. Despite such "deterministic evolutions", the all-comprehensive set partition $\boldsymbol{\lambda}_3^a = \{\{\boldsymbol{a}_{1,3}, \ldots, \boldsymbol{a}_{8,3}\}\}$ is still associated to non deterministic transitions of the chain; indeed, the* 3*-path transition probability matrix associated to $\boldsymbol{\lambda}_3^a$ is*

$\mathcal{M}_3^a =$

	states a_z	
partition classes $\boldsymbol{A}_{q,3}^a$ of $\boldsymbol{\lambda}_3^a$	1	2
$\{(1,1,1),...,(2,2,2)\}$	0.5	0.5

Generally speaking, for a given k and A, the all-comprehensive set partition loses all the information about the conditional distribution of $X(t)$, for each $t \geq 0$, while the partition of singletons preserves all the information available about that distribution.

4 Optimal Criteria

The aim of this section is to present some optimal criteria for choosing the order k of the Markov chain and the clustering of A^k. As already mentioned in the Introduction, our optimization problems are based on two competing guidelines: statistical similarity and multiplicity.

4.1 Information-type criteria

Consider a Markov chain $\{X(t), t \geq 0\}$ of order $k \geq 1$, where A is its state space, and Ω is the event space of all its trajectories. Let $\mathcal{G}$ be a functional space, and $g \in \mathcal{G}$ be a transformation of the process $\{X(t), t \geq 0\}$ classifying all its trajectories into the classes of a partition $\boldsymbol{\lambda}$. In particular, class q of partition $\boldsymbol{\lambda}$, namely $\boldsymbol{A}_q$, contains the trajectories of $\{X(t), t \geq 0\}$ having k-path $\boldsymbol{a}_h$ as their last k realizations ($\boldsymbol{a}_h$ is used here to name any k-path included in class q).

Clearly there is a bijection between the g's and the $\boldsymbol{\lambda}$'s. Consequently, letting $\mathcal{I}_g$ be the σ-algebra generated by g, it can be viewed as the information generated by $\boldsymbol{\lambda}$. We denote hereafter $\{X(t), t \geq 0\} \,|\mathcal{I}_g$ as the stochastic process $\{X(t), t \geq 0\}$ conditioned on the information provided through $\mathcal{I}_g$.

In the spirit of Kolmogorov (1965), we define a disorder measure for $\{X(t), t \geq 0\}$ given the information $\mathcal{I}_g$, and denote it as

$$\eta(\{X(t), t \geq 0\} \,|\mathcal{I}_g) = \{\eta(X(t)|\mathcal{I}_g), t \geq 0\},$$

where η is a function transforming random variables in nonnegative real numbers. This measure should not be understood as the conditional probability of the random variables $X(t)$, as t varies, rather as the "ignorance" that we have about their conditional distributions. Achieving a value of $\eta = 0$ will therefore tell us that we have perfect knowledge about the (conditional) distribution of $\{X(t), t \geq 0\}$, not that we have eliminated its randomness.

A definition concerning the equivalence of the informative contents of transformations is needed.

Definition 2. *Consider $g_1, g_2 \in \mathcal{G}$, and suppose that they are associated to a pair of σ-algebras $\mathcal{I}_{g_1}, \mathcal{I}_{g_2}$, respectively. We say that g_1 and g_2 generate the same information with respect to the process $\{X(t), t \geq 0\}$ when $\eta(\{X(t), t \geq 0\} \,|\mathcal{I}_{g_1}) = \eta(\{X(t), t \geq 0\} \,|\mathcal{I}_{g_2})$. We denote in this case $g_1 \sim g_2$ or, equivalently, $\mathcal{I}_{g_1} \sim \mathcal{I}_{g_2}$.*

We denote as $g_a \in \mathcal{G}$ the less informative transformation. It is associated to the all-comprehensive set partition $\boldsymbol{\lambda}^a$ (the partition making no distinction among all k-paths) and generates the σ-algebra $\mathcal{I}_a = \{\emptyset, \Omega\}$.

Following an information-type argument (see Kolmogorov, 1965), we can define the *gain* in applying g at $\{X(t), t \geq 0\}$

$$I(g) = \eta(\{X(t), t \geq 0\} \,|\mathcal{I}_a) - \eta(\{X(t), t \geq 0\} \,|\mathcal{I}_g).$$

Among all the g's in $\mathcal{G}$, we call g_s the most information conservative transformation. It distinguishes any k-path $\boldsymbol{a}_h$, in the sense that, under such transformation, different k-paths will be assigned to different classes of the related partition $\boldsymbol{\lambda}^s$. Hence, $\boldsymbol{\lambda}^s$ is a partition of singletons and $\mathcal{I}_s$ indicates the corresponding σ-algebra. It is easy to show that the functionals g_a and g_s are opposite in the following sense:

$$g_a \in \operatorname*{arg\,max}_{g \in \mathcal{G}} \; \eta\left(\{X(t), t \geq 0\} \,|\mathcal{I}_g\right); \tag{7}$$

$$g_s \in \operatorname*{arg\,min}_{g \in \mathcal{G}} \; \eta\left(\{X(t), t \geq 0\} \,|\mathcal{I}_g\right). \tag{8}$$

The following result states immediately:

Theorem 3. *It holds*

$$0 \leq I(g) \leq \eta(\{X(t), t \geq 0\} \,|\mathcal{I}_a), \quad \forall g \in \mathcal{G},$$

with $I(g_a) = 0$ and $I(g_s) = \eta(\{X(t), t \geq 0\} | \mathcal{I}_a)$.

Remark 4. *The result stated above has an intuitive interpretation: if the σ-algebra associated to g is the most informative (i.e. $g \sim g_s$), then the gain in applying g to $\{X(t), t \geq 0\}$ is maximum, in that g reduces the disorder by an amount equal to $\eta(\{X(t), t \geq 0\} | \mathcal{I}_a)$. Conversely, there is no gain in applying the less informative g, i.e. if $g \sim g_a$.*

To link our work to this information-type framework, we specify in the following sections two distance indicators, which we call $d_{\boldsymbol{\lambda}}$ and $v_{\boldsymbol{\lambda}}$, as disorder measures for the conditional distribution of $\{X(t), t \geq 0\}$. As it will be apparent after the analysis of these two distances, $d_{\boldsymbol{\lambda}}$ and $v_{\boldsymbol{\lambda}}$ fulfill the defining properties stated in Kolmogorov (1965), as for both of them we have:

$$\begin{aligned} \boldsymbol{\lambda}^a &\in \underset{\boldsymbol{\lambda} \in \boldsymbol{\Lambda}_k}{\arg\max} \quad d_{\boldsymbol{\lambda}} \\ \boldsymbol{\lambda}^a &\in \underset{\boldsymbol{\lambda} \in \boldsymbol{\Lambda}_k}{\arg\max} \quad v_{\boldsymbol{\lambda}}, \end{aligned} \tag{9}$$

and

$$\begin{aligned} \boldsymbol{\lambda}^s &\in \underset{\boldsymbol{\lambda} \in \boldsymbol{\Lambda}_k}{\arg\min} \quad d_{\boldsymbol{\lambda}} \\ \boldsymbol{\lambda}^s &\in \underset{\boldsymbol{\lambda} \in \boldsymbol{\Lambda}_k}{\arg\min} \quad v_{\boldsymbol{\lambda}}. \end{aligned} \tag{10}$$

Observe that (9) is equivalent to (7) and (10) is equivalent to (8), because of the bijection between g and $\boldsymbol{\lambda}$. As already discussed several other disorder measures can of course be devised instead of the ones we advance. We remark here that respecting the Kolmogorov properties requires careful inspection. It can happen in some cases that the partition giving the lowest disorder is not the partition of singletons ($\boldsymbol{\lambda}^s$), or that the maximum disorder is not achieved through $\boldsymbol{\lambda}^a$. For example, a slight variation of the distance indicator $d_{\boldsymbol{\lambda}}$ (as shown in Remark 8) turns out to violate the arg max requirement in (9).

4.1.1 Bootstrapping

So far we have dealt with the reduction of a disorder measure η about the conditional distribution of $\{X(t), t \geq 0\}$. In the absence of any type of constraints, it should be obvious for a researcher to take the partition of singletons $\boldsymbol{\lambda}^s$ as the best choice in replicating the original series. However dealing with Markov chain bootstrapping such choice is not trivial at all. Indeed it can happen that for η approaching 0 the following outcome also results:

$$P\left(a_z | \boldsymbol{a}_h\right) = 1 \text{ or } 0,$$

for all $z = 1, ..., J_N$ and all $h = 1, ..., (J_N)^k$, that is the model forecasts with certainty if a time t realization of the process is $X(t) = a_z$ or not, whatever its previous k-path. In such cases the bootstrapped series will be exact replications of the original series, starting from the initial k observations.

In practice such a situation will usually verify when the number of observations are insufficient with respect to the initial number of states J_N and the number of time lags k (i.e. insufficient sampling to estimate the transition probability matrix).

In such cases joining some states through a partition $\boldsymbol{\lambda}$ coarser than $\boldsymbol{\lambda}^s$, amounts to reintroducing some randomness in the bootstrapped series. Indeed joining the rows of the transition probability matrix in classes, recovers a non-degenerate conditional distribution of $\{X(t), t \geq 0\}$. However notice that, in the lack of knowledge about the true conditional distribution of the process $\{X(t), t \geq 0\}$, a partition $\boldsymbol{\lambda}$ coarser than $\boldsymbol{\lambda}^s$ re-introduces also disorder next to randomness, and we will not be able to distinguish neatly between the two effects. This key remark justifies the need of a method to reintroduce randomness in a controlled way.

Our proposal consists in measuring the degree of the potential diversification of the bootstrapped series linked to a given partition. In particular, we introduce a multiplicity measure and denote it as $m(\{X(t), t \geq 0\} \,|\mathcal{I}_g)$. Among all the partitions sharing the same measure of multiplicity, we will select the one with the lowest level of disorder. Such method corresponds to the following optimization problem:

$$\begin{aligned} &\min_{g \in \mathcal{G}} \eta(\{X(t), t \geq 0\} \,|\mathcal{I}_g) \\ &\text{s.t.} \quad m(\{X(t), t \geq 0\} \,|\mathcal{I}_g) \geq \gamma, \end{aligned} \tag{11}$$

where $\gamma \geq 0$. Letting γ vary, a set of optimal solutions of problem (11) obtains.

Two multiplicity measures $m(\{X(t), t \geq 0\} \,|\mathcal{I}_g)$ will be defined, and denoted as $l_{\boldsymbol{\lambda}}$ and $m_{\boldsymbol{\lambda}}$.

4.2 First distance indicator: Absolute difference of k-path transition probabilities

The first distance indicator focuses on the absolute difference between the elements of the k-path transition probability matrix. Fixed a value for k, we can define a distance $d_{i,j}$ between two paths $\boldsymbol{a}_i$ and $\boldsymbol{a}_j$ as follows:

$$d_{i,j} := \sum_{z=1}^{J_N} |P(a_z|\boldsymbol{a}_i) - P(a_z|\boldsymbol{a}_j)| \,. \tag{12}$$

In order to preserve similarity, we notice that $\boldsymbol{a}_i$ and $\boldsymbol{a}_j$ should be grouped together when their distance $d_{i,j}$ is close to zero: in this case, we have no reason to distinguish the paths $\boldsymbol{a}_i$ and $\boldsymbol{a}_j$. By extending this argument, we stress that it is desirable that the elements composing the classes of a suitable partition are close enough to each other, at least on average. We formalize this point. Let us consider a partition $\boldsymbol{\lambda} \in \boldsymbol{\Lambda}_k$ such that $\boldsymbol{\lambda} = (\lambda_k, \ldots, \lambda_1)$ and $\boldsymbol{A}_q$ as in (6). The distance in $\boldsymbol{A}_q$ is defined as

$$d_{\boldsymbol{A}_q} := \max_{i,j:\boldsymbol{a}_i, \boldsymbol{a}_j \in \boldsymbol{A}_q} d_{i,j}. \tag{13}$$

We can finally characterize the distance $d_{\boldsymbol{\lambda}}$ of partition $\boldsymbol{\lambda}$ with the average value of its classes distances. More precisely, we have

$$d_{\boldsymbol{\lambda}} := \frac{1}{C} \cdot \sum_{q=1}^{|\boldsymbol{\lambda}|} d_{\boldsymbol{A}_q} \cdot |\boldsymbol{A}_q|, \tag{14}$$

where $|\boldsymbol{A}_q|$ is the cardinality of partition class $\boldsymbol{A}_q$ and $C = \sum_{q=1}^{|\boldsymbol{\lambda}|} |\boldsymbol{A}_q|$.

Remark 5. *The cardinalities of the classes $\boldsymbol{A}_q$ are calculated discarding the k-paths having null rows in (3).*

Proposition 6. $d_{\boldsymbol{\lambda}} \in [0, 2]$.

Proof. See Appendix A. □

Remark 7. *The all-comprehensive set partition takes the maximum value of $d_{\boldsymbol{\lambda}}$ (not necessarily 2). The opposite case, represented by the partition of singletons, is associated (with certainty) to $d_{\boldsymbol{\lambda}} = 0$, since any singleton has zero distance from itself.*

Remark 8. *Observe that if we defined the distance indicator by interchanging the calculations of (13) and (14), we would obtain a contradiction. Indeed, define*

$$\tilde{d}_{\boldsymbol{A}_q} := \frac{1}{|\boldsymbol{A}_q|^2} \sum_{i,j:\boldsymbol{a}_i,\boldsymbol{a}_j \in \boldsymbol{A}_q} d_{i,j}$$

as the (simple) average distance of partition class $\boldsymbol{A}_q$. Define then

$$\tilde{d}_{\boldsymbol{\lambda}} := \max_{\boldsymbol{A}_q \in \boldsymbol{\lambda}} d_{\boldsymbol{A}_q}$$

as the distance indicator of partition $\boldsymbol{\lambda}$.

It is easy to show that such a defined distance indicator causes the all-comprehensive set partition to take a value strictly less than other partitions; such indicator contradicts the request of a similarity (distance) criterion to exhibit its minimum (maximum) value if all the elements are grouped together (see Theorem 3).

4.3 Second distance indicator: Variance-type measure of k-path transition probabilities

The second distance indicator is constructed by taking into account the average error made within the classes of a partition. Let us consider a partition $\boldsymbol{\lambda} \in \boldsymbol{\Lambda}_k$ such that $\boldsymbol{\lambda} = (\lambda_k, \ldots, \lambda_1)$ and $\boldsymbol{A}_q$ as in (6).

We then proceed by defining a variance-type measure of the multidimensional class $\boldsymbol{A}_q$ as follows:

$$v_{\boldsymbol{A}_q} := \frac{1}{J_N} \cdot \sum_{z=1}^{J_N} \left\{ \sum_{i:\boldsymbol{a}_i \in \boldsymbol{A}_q} W_i \cdot [P(a_z|\boldsymbol{a}_i) - P(a_z|\boldsymbol{A}_q)]^2 \right\}, \tag{15}$$

with weights

$$W_i = \frac{P(\boldsymbol{a}_i)}{\sum_{c:\boldsymbol{a}_c \in \boldsymbol{A}_q} P(\boldsymbol{a}_c)}.^{2}$$

In this case, we preserve the similarity by imposing that the classes of a suitable partition have a low value of the indicator defined in (15). More generally, the entire partition should have a low value of the variance-type measure. To this end, we introduce a weighted average of variance-type measures of partition classes: given $\boldsymbol{\lambda}$, we define its associated variance-type measure as the weighted average of the $v_{\boldsymbol{A}_q}$'s:

$$v_{\boldsymbol{\lambda}} := \frac{1}{C} \cdot \sum_{q=1}^{|\boldsymbol{\lambda}|} v_{\boldsymbol{A}_q} \cdot |\boldsymbol{A}_q|, \tag{16}$$

with $C = \sum_{q=1}^{|\boldsymbol{\lambda}|} |\boldsymbol{A}_q|$.

We state the following:

Proposition 9. $v_{\boldsymbol{\lambda}} \in [0, 0.25]$.

Proof. See Appendix A. □

Remark 10. *The all-comprehensive set partition identifies the minimum level of similarity, i.e. the maximum value of* $v_{\boldsymbol{\lambda}}$ *(not necessarily* 0.25*).*

It is easily observed that $v_{\boldsymbol{\lambda}} = 0$ *if the k-path transition probability matrix shows uniformly distributed columns within each class* $\boldsymbol{A}_q$*. The partition of singletons clearly verifies such condition.*

4.4 Multiplicity measure

The multiplicity measures we propose are based on the size of the partition classes.

Let us define $l_{\boldsymbol{\lambda}}$ an *absolute multiplicity measure* of the partition $\boldsymbol{\lambda}$:

$$l_{\boldsymbol{\lambda}} := \sum_{q=1}^{|\boldsymbol{\lambda}|} |\boldsymbol{A}_q|^2. \tag{17}$$

The following result holds:

Proposition 11. *It results*

$$C \leq l_{\boldsymbol{\lambda}} \leq C^2,$$

with $C = \sum_{q=1}^{|\boldsymbol{\lambda}|} |\boldsymbol{A}_q|$.

Proof. See Appendix A. □

We can also define a *relative multiplicity measure* $m_{\boldsymbol{\lambda}}$, related to a partition $\boldsymbol{\lambda}$, by normalizing $l_{\boldsymbol{\lambda}}$ as follows:

$$m_{\boldsymbol{\lambda}} := \frac{\sqrt{l_{\boldsymbol{\lambda}}} - \sqrt{C}}{C - \sqrt{C}}. \tag{18}$$

[2] It is easy to see that

$$P(a_z|\boldsymbol{A}_q) = \sum_{i:\boldsymbol{a}_i \in \boldsymbol{A}_q} W_i \cdot P(a_z|\boldsymbol{a}_i).$$

By Proposition 11 and arguments above, we have $m_{\boldsymbol{\lambda}} \in [0,1]$, being

$$\begin{cases} m_{\boldsymbol{\lambda}} = 0, & \text{for } |\lambda_w| = J_N, \ \forall\, w = 1, \dots, k; \\ m_{\boldsymbol{\lambda}} = 1, & \text{for } |\lambda_w| = 1, \ \forall\, w = 1, \dots, k. \end{cases}.$$

In the statement of the optimization problems, as we shall see, $m_{\boldsymbol{\lambda}}$ will be the adopted multiplicity measure.

4.5 Two optimization problems

We now present two optimization problems based on the similarity and multiplicity criteria developed so far. Solving them will provide a way to answer the questions addressed in this paper.
The first one is based on the distance defined in (14).

Definition 12. *Let us consider $\gamma \in [0,1]$, $k^* \in \{1, \dots, N\}$, and $\boldsymbol{\lambda}^* = (\lambda^*_{k^*}, \dots, \lambda^*_1) \in \boldsymbol{\Lambda}_{k^*}$.*
We say that the couple $(k^*, \boldsymbol{\lambda}^*)$ is d-γ-optimal when it is the solution of the following minimization problem:

$$\begin{aligned} &\min_{(k,\boldsymbol{\lambda}) \in \{1,\dots,N\} \times \boldsymbol{\Lambda}_k} d_{\boldsymbol{\lambda}} \\ &\text{s.t.} \quad m_{\boldsymbol{\lambda}} \geq \gamma. \end{aligned} \tag{19}$$

The second optimization problem involves the variance-type measure defined in (16).

Definition 13. *Let us consider $\gamma \in [0,1]$, $k^* \in \{1, \dots, N\}$, and $\boldsymbol{\lambda}^* = (\lambda^*_{k^*}, \dots, \lambda^*_1) \in \boldsymbol{\Lambda}_{k^*}$.*
The couple $(k^*, \boldsymbol{\lambda}^*)$ is said to be v-γ-optimal when it is the solution of the following minimization problem:

$$\begin{aligned} &\min_{(k,\boldsymbol{\lambda}) \in \{1,\dots,N\} \times \boldsymbol{\Lambda}_k} v_{\boldsymbol{\lambda}} \\ &\text{s.t.} \quad m_{\boldsymbol{\lambda}} \geq \gamma. \end{aligned} \tag{20}$$

In both Definition 12 and 13, we have that k^* is the optimal order of a Markov chain describing the evolutive phenomenon. Moreover, $\boldsymbol{\lambda}^*$ provides the optimal time-dependent clustering of the state space, in order to have an approximation of the k^*-path transition probability matrix.
According to the definitions of $d_{\boldsymbol{\lambda}}$, $v_{\boldsymbol{\lambda}}$, and $m_{\boldsymbol{\lambda}}$, we can briefly discuss the two optimization problems. Letting the multiplicity measure reach its minimum ($\gamma = 0$) is equivalent to allow for the partition of singletons, which ensures the minimum distance ($d_{\boldsymbol{\lambda}}, v_{\boldsymbol{\lambda}} = 0$). Letting $\gamma = 1$ corresponds to forcing the maximum level of multiplicity. This boundary in our case is satisfied only by the all-comprehensive set partition, in which case the two distance indicators take their maximum value.
It is important to point out how this approach selects jointly the relevant states and the time lags. Consider a time lag $w \leq k$ and suppose that a couple of paths $\boldsymbol{a}_i$ and $\boldsymbol{a}_j$ are both in state a_u at time lag w, while another couple $\boldsymbol{a}_m$ and $\boldsymbol{a}_n$ are in state a_x at the same time lag. For ease of

notation, let us call the first as the u-couple and the second as the x-couple. In addition suppose that coincidentally the paths of the u-couple have very similar transition probabilities; the paths of the x-couple also have very similar transition probabilities but very different from those of the u-couple. Keeping all other things equal, both minimization problems (19) and (20) will favor those partitions combining the u-couple and the x-couple in two separate classes. Distinguishing states a_u and a_x at time lag w would be relevant to our minimization problems.

If, on the contrary, the four paths were all very similar with respect to their transition probabilities, the partitions joining all of them will be preferred, as they would increase the multiplicity criterion. As a consequence states a_u and a_x at time lag w would result jointly of no relevance.

5 Methodological Issues

To perform the optimization procedures, a researcher faces several technical problems; an important computational problem is the restriction of the set of admissible solutions. In particular, we present in the following two methods/concepts that could help identifying which time lags "count" to determine the evolution of a process at time t.

A technical definition is firstly needed.

Definition 14. *Let us consider a k-dimensional partition $\boldsymbol{\lambda} = (\lambda_k, \ldots, \lambda_1)$ of set A^k. Time lag $w \in \{1, ..., k\}$ is a partition time for $\boldsymbol{\lambda}$ when $\lambda_w \neq \{A\}$, or, equivalently, $|\lambda_w| > 1$.*

We introduce the concept of *longest-memory k* in the following:

Definition 15. *Let us consider a k-dimensional partition $\boldsymbol{\lambda} = (\lambda_k, \ldots, \lambda_1)$. The longest-memory k for $\boldsymbol{\lambda}$, call it lm-$k_{\boldsymbol{\lambda}}$, is a time lag such that:*

- *lm-$k_{\boldsymbol{\lambda}} \in \{1, ..., k\}$;*
- *lm-$k_{\boldsymbol{\lambda}}$ is a partition time;*
- *if lm-$k_{\boldsymbol{\lambda}} < k$, the set $\{$lm-$k_{\boldsymbol{\lambda}} + 1, ..., k\}$ does not contain partition times.*

Remark 16. *It is worth noting that, if the set of partition times of $\boldsymbol{\lambda}$ is not empty, lm-$k_{\boldsymbol{\lambda}}$ represents its maximum.*

An *lm-$k_{\boldsymbol{\lambda}}$* represents the maximum number of time lags that can be considered in building up a partition without loosing information: indeed, the time series values are grouped all together before that time lag (third condition of the previous definition).

We discuss now some important properties of partitions and distance indicators depending on the previous definition of *longest-memory k*. Let us consider the partitions $\boldsymbol{\lambda}$ and $\boldsymbol{\lambda}'$ with $\boldsymbol{\lambda} = (\lambda_k, \ldots, \lambda_{lm\text{-}k_{\boldsymbol{\lambda}}}, \ldots, \lambda_1)$ and $\boldsymbol{\lambda}' = (\lambda_{lm\text{-}k_{\boldsymbol{\lambda}}}, \ldots, \lambda_1)$. It is easily seen that the two partitions have the same number of classes; in addition, the existence of *lm-$k_{\boldsymbol{\lambda}}$* implies that the distance indicators should yield the same value for both the partitions $\boldsymbol{\lambda}$ and $\boldsymbol{\lambda}'$.

We can extend the properties of partitions and distance indicators to a generic time lag (not necessarily a *longest-memory* k). More precisely, we state the following theorem:

Theorem 17. *Consider a partition $\boldsymbol{\lambda} = (\lambda_k, \dots, \lambda_1)$. Define the w-penalized partition $\boldsymbol{\lambda}^{(-w)} := (\lambda_k, \dots, \lambda_{w+1}, \lambda_{w-1}, \dots, \lambda_1)$, with $w \in \{1, \dots, k\}$. Assume that:*

a. *w is not a partition time;*

b. *for any $a_z \in A$ and any couple of k-paths $\boldsymbol{a}_i$ and $\boldsymbol{a}_j$ with $a_{i,l} = a_{j,l}$ for $l = 1, ..., w-1, w+1, ..., k$, it holds $P(a_z|\boldsymbol{a}_i) = P(a_z|\boldsymbol{a}_j)$.*

Then:

1. *$|\boldsymbol{\lambda}| = |\boldsymbol{\lambda}^{(-w)}|$ (partitions $\boldsymbol{\lambda}$ and $\boldsymbol{\lambda}^{(-w)}$ have the same cardinality);*

2. *$d_{\boldsymbol{\lambda}} = d_{\boldsymbol{\lambda}^{(-w)}}$ and $v_{\boldsymbol{\lambda}} = v_{\boldsymbol{\lambda}^{(-w)}}$.*

Proof. See Appendix A. □

The theorem holds not only for a generic time lag w, but also for a set of r generic time lags $\{w_1, ..., w_r\}$, with $r > 1$.

We now introduce the important concept of *ε-active time lag*.

Definition 18. *Given $\varepsilon \in [0, 1]$ and $w \in \{1, \dots, k\}$, a time lag w is said ε-active when, for any $a_z \in A$, the following conditions are fulfilled:*

- *$|P(a_z|\boldsymbol{a}_i) - P(a_z|\boldsymbol{a}_j)| < \varepsilon$, where $\boldsymbol{a}_i$ can differ from $\boldsymbol{a}_j$ in all times but $t - w$, for any couple i, j,*
- *ε is the lowest number satisfying the previous inequality.*

In other words, the observation of the process in $t - w$ brings a "key information" to determine its evolution at time t.

This definition can be extended to combinations of several *ε-active time lags* as follows:

Definition 19. *Given $\varepsilon \in [0, 1]$ and ρ indexes $w_1, \dots, w_\rho \in \{1, ..., k\}$, the time lags $w_1, \dots, w_\rho$ are said joint ε-active when, for any $a_z \in A$, the following conditions are fulfilled:*

- *$|P(a_z|\boldsymbol{a}_i) - P(a_z|\boldsymbol{a}_j)| < \varepsilon$, where $\boldsymbol{a}_i$ can differ from $\boldsymbol{a}_j$ in all times but $t - w_1, \dots, t - w_\rho$, for any couple i, j,*
- *ε is the lowest number satisfying the previous inequality.*

Remark 20. *It does not make sense to extend the search for active ρ-tuples whose size is greater than $k - 1$, where k is the order of the Markov chain $\{X(t), t \geq 0\}$. Verifying that all the k time lags are ε-active is equivalent to find that none time is of particular importance over the others for the analysis at time t of the phenomenon described by $X(t)$.*

We now see how we can jointly use the definitions of *longest-memory* k and *joint* ε*-active time lags*. Consider the time lags which are less than or equal to the *longest-memory* k, i.e. the set $\{1, ..., lm\text{-}k_{\boldsymbol{\lambda}}\}$. If we know which time lags in $\{1, ..., lm\text{-}k_{\boldsymbol{\lambda}}\}$ are *joint* ε*-active*, we can neglect all the others and avoid to evaluate the corresponding partitions.

To be more precise, we detail here the conditions for selecting the non-dominated solutions and build the efficient frontier. Such definitions will turn out to be useful in the next section, devoted to the application of our methodology.

Definition 21. *Let us consider a couple of partitions* $\boldsymbol{\lambda}^u, \boldsymbol{\lambda}^x \in \boldsymbol{\Lambda}_k$*; we say that* $\boldsymbol{\lambda}^u$ *is d-m-non-dominated (v-m-non-dominated) by* $\boldsymbol{\lambda}^x$ *when*

$$\begin{cases} d_{\boldsymbol{\lambda}^u} \geq d_{\boldsymbol{\lambda}^x} \\ m_{\boldsymbol{\lambda}^u} \geq m_{\boldsymbol{\lambda}^x} \end{cases} \quad \text{or} \quad \begin{cases} d_{\boldsymbol{\lambda}^u} \leq d_{\boldsymbol{\lambda}^x} \\ m_{\boldsymbol{\lambda}^u} \leq m_{\boldsymbol{\lambda}^x} \end{cases} \tag{21}$$

$$\left(\begin{cases} v_{\boldsymbol{\lambda}^u} \geq v_{\boldsymbol{\lambda}^x} \\ m_{\boldsymbol{\lambda}^u} \geq m_{\boldsymbol{\lambda}^x} \end{cases} \quad \text{or} \quad \begin{cases} v_{\boldsymbol{\lambda}^u} \leq v_{\boldsymbol{\lambda}^x} \\ m_{\boldsymbol{\lambda}^u} \leq m_{\boldsymbol{\lambda}^x} \end{cases} \right).$$

According to the previous definition, dominated partitions will be discarded in our analysis; basically, the rejected partitions show no lower distance ($d_{\boldsymbol{\lambda}}$, or $v_{\boldsymbol{\lambda}}$) and no higher multiplicity ($m_{\boldsymbol{\lambda}}$), with at least a strict inequality holding.

We now turn to the optimization problems (19) and (20) and introduce the efficient frontier, defined as follows:

Definition 22. *Consider* $\bar{k} \in \{1, \ldots, N\}$.

i. *The efficient frontier* $\mathcal{F}_{m,d,\bar{k}}$ *related to optimization problem (19) is:*

$$\mathcal{F}_{m,d,\bar{k}} := \bigcup_{\gamma \in [0,1]} \{(m_{\boldsymbol{\lambda}^*}, d_{\boldsymbol{\lambda}^*}) \in [0,1] \times [0,2]\},$$

where $\boldsymbol{\lambda}^*$ *is the solution of the problem:*

$$\min_{\boldsymbol{\lambda} \in \boldsymbol{\Lambda}_{\bar{k}}} d_{\boldsymbol{\lambda}}$$

$$\text{s.t.} \quad m_{\boldsymbol{\lambda}} \geq \gamma.$$

ii. *The efficient frontier* $\mathcal{F}_{m,v,\bar{k}}$ *related to optimization problem (20) is:*

$$\mathcal{F}_{m,v,\bar{k}} := \bigcup_{\gamma \in [0,1]} \{(m_{\boldsymbol{\lambda}^*}, v_{\boldsymbol{\lambda}^*}) \in [0,1] \times [0,0.25]\},$$

where $\boldsymbol{\lambda}^*$ *is the solution of the problem:*

$$\min_{\boldsymbol{\lambda} \in \boldsymbol{\Lambda}_{\bar{k}}} v_{\boldsymbol{\lambda}}$$

$$\text{s.t.} \quad m_{\boldsymbol{\lambda}} \geq \gamma.$$

It is worth noting that we can build an efficient frontier for each value of $k \in \{1, \dots, N\}$. In practice, once k has been set equal to $\bar{k}$, the procedure to build the efficient frontiers associated to the two optimization problems (19) and (20) can be synthesized in the following points:

1. initially the researcher orders the set of admissible solutions by increasing values of their distance indicator (v or d);

2. starting from the solution with the lowest value of distance, she/he scans for the next solution with a higher distance and a higher value of multiplicity (m) and discards the intermediate solutions (dominated in the sense of Definition 21);

3. step 2. is repeated until the worst value of distance is reached.

The partitions remaining after step 3. constitute the optimal solutions and the values of their distance indicator and multiplicity measure represent the *efficient frontier* $\mathcal{F}_{m,d,\bar{k}}$ or $\mathcal{F}_{m,v,\bar{k}}$.
It is relevant to assess the finite time performance of the above 3 step procedure. Firstly, we stress that the procedure provides the solution of the optimization problems (19) and (20) as the parameter γ varies in $[0, 1]$. The complexity of the problems increases dramatically as the number of time lags and states of the Markov chain grow. The following result formalizes this aspect.

Proposition 23. *The time required to span the set of admissible solutions is* $O\left([J_N^2 B\left(J_N\right)]^k\right)$ *for optimization problem (19) and* $O\left([J_N B\left(J_N\right)]^k\right)$ *for optimization problem (20) as* $J_N \to +\infty$*, where* J_N *is the number of states and* k *is the order of a Markov chain.*

Proof. See Appendix A. □

As an example, Table 1 shows the cardinality of the set of admissible solutions for various combinations of time lags k and states J_N characterizing a Markov chain. Remember that such cardinality is equal to $[B\left(J_N\right)]^k$ (see footnote 1).

Insert Table 1 here

6 Numerical Test

To test the effectiveness of our method, we devise the following experiment:

1. we consider a Markov chain of order k, with k set to a chosen value $\bar{k}$, and artificially design the associated $\bar{k}$-path transition probability matrix. The rows on this matrix are joined following a partition, which we call here as “true” partition, where only some of the time lags are “active” and equivalent states (i.e. those generating similar transition probabilities) are grouped together. This matrix defines the effective conditional probability distribution of a Markov chain and serves as benchmark;

2. based on such matrix, we generate a simulated trajectory of $5,000$ observations;
3. an empirical transition probability matrix is then estimated from this simulated series;
4. our optimization procedure is then applied both to the benchmark and to the empirical matrices and their solutions (represented through efficient frontiers) are compared. Such procedure is replicated for both the two distance indicators analyzed here.

If the procedure is effective, then the benchmark and the empirical solutions should "largely" intersect and the true partition should be one of the preferred solutions. More specifically, our experiment consists in a severe reverse-engineering test, where some parameter estimates obtained from empirical investigation, instead of being tested for statistical significance, are compared with their "true" values, which is a definitely more conclusive result. We also expect that the method should be fairly robust to the choice of the distance indicator adopted.
We run this experiment starting with two different transition probability matrices.

6.1 k-path transition probability matrix design

The considered Markov chains (and their transition probability matrices) are defined as follows:

I. a Markov chain of order $\bar{k} = 5$ and with state space $A = \{1, 2, 3\}$, such that only time lags 3 and 2 are active in the sense of Definition 18. This means that the values observed in time lag 1, 4, and 5 have no influence on the evolution of the process. So for comparison purposes we will consider transition probability matrices $\mathcal{A}^{bench}$ and $\mathcal{A}^{empir}$ with dimensions $3^5 \times 3 = 243 \times 3$;

II. a Markov chain of order $\bar{k} = 3$ and with state space $B = \{1, 2, 3, 4, 5\}$, such that only time lag 2 and 1 are active. In this case, the transition probability matrices are denoted with $\mathcal{B}^{bench}$ and $\mathcal{B}^{empir}$ and have dimension $5^3 \times 5 = 125 \times 5$.

The four transition probability matrices are available at the web page `http://chiara.eco.unibs.it/~pelizcri/CuttedTable1andTable2new.xls`. To obtain a complete view of the information embedded in these matrices, consider Tables 2 and 3, where the true partitions are clearly represented. We call these two partitions as $\boldsymbol{\lambda}^{A,tr}$ and $\boldsymbol{\lambda}^{B,tr}$ respectively for cases **I**. and **II**.. The same tables also show which time lags are "active":

- time lags 3 and 2 in matrix $\mathcal{A}^{bench}$ are *joint* 0.23-*active* (singularly considered, $t-5$, $t-4$, $t-3$, $t-2$, and $t-1$ are ε-*active*, with ε between 0.83 and 0.84);
- time lags 2 and 1 in matrix $\mathcal{B}^{bench}$ are *joint* 0.04-*active* (singularly considered, $t-3$, $t-2$, and $t-1$ are 0.44-*active*, 0.34-*active*, and 0.39-*active*, respectively).

Insert Tables 2 and 3 here

Tables 4 and 5 show the average values of the transition probabilities associated to the states grouped following the true partitions. The black horizontal lines in the matrices help to represent the corresponding classes. These partitions are formed combining the classes defined in each time lag, as it has been discussed in the theoretical settings (see Section 3). Values are taken averaging over the non ε-active time lags. In particular in Table 4, which refers to case **I.**, each row represents a 5-path observed at active time lags 2 and 3, and the transition probabilities are obtained averaging 27 rows (i.e. the combinations of 3 states in the 3 non ε-active lags) of matrix $\mathcal{A}^{bench}$. The rows in Table 5, which refers to case **II.,** are the average probabilities calculated over the corresponding 5 rows in matrix $\mathcal{B}^{bench}$ (i.e. the 5 states in the only non ε-active time lag 3).

Insert Tables 4 and 5 here

Numbers in bold help to represent which states the process tends to evolve to preferably, conditional on its past values. As it is immediate to observe, the rows tend to be very similar when they are in the same group and change significantly from class to class.

6.2 Simulation and estimation of the empirical transition probability matrix

As anticipated at the beginning of the present section, for each case a simulated trajectory has been generated consisting of $5,000$ values. The simulation has been based on a Monte Carlo method[3]. For each simulated series the corresponding empirical transition probability matrix has been estimated, based on the usual conditional frequency calculation.

The most obvious differences between the benchmark and the empirical matrices are concerned with the values of the transition probabilities. Besides, another possible difference consists of the loss of some rows in the empirical matrix, a case which can verify if the process has very low probabilities (if not zero) to follow some paths. Finally some paths can be observed with a frequency which is too low to supply a significant estimate of the corresponding row. To estimation purposes, rows with a low frequency (i.e. less than 20) have been treated in the same way as the rows which have never been observed in the simulated series: in both cases those rows have been set to zero, following (3).

6.3 Optimization procedure

The set of admissible solutions in case **I**. is formed by $3,125$ partitions (the set of partitions on A is Λ^A, with $|\Lambda^A| = 5$, and $|(\Lambda^A)^5| = |\Lambda^A|^5 = 5^5$). For case **II**. the same calculation results in $140,608$ partitions (the set of partitions on B is Λ^B, with $|\Lambda^B| = 52$, and $|(\Lambda^B)^3| = |\Lambda^B|^3 = 52^3$).

[3] For a Markov chain of order $k \geq 1$ the simulation procedure starts by fixing an initial combination of k conditional values and finding the corresponding row on the transition probability matrix. The next value of the Markov chain is selected extracting a uniformly distributed random number and then applying it to the inverse of the transition probability distribution of the row just fixed. The selected value is then used to update the conditioning k-path, and the simulation procedure can be iterated.

To solve the two optimization problems (19) and (20), we have calculated the distance indicators and the multiplicity measure for every partition (see (14), (16), and (18)) in the set of admissible solutions of cases **I**. and **II**.. For each case the procedure has been applied both to the benchmark and the empirical transition probability matrices. Summing up the combinations, the 3 step procedure presented at the end of Section 5 has been applied 8 times (2 distance indicators × 2 cases × 2 transition probability matrices) and has generated 8 efficient frontiers $\mathcal{F}_{m,d,5}^{bench}$, $\mathcal{F}_{m,v,5}^{bench}$, $\mathcal{F}_{m,d,3}^{bench}$, $\mathcal{F}_{m,v,3}^{bench}$, $\mathcal{F}_{m,d,5}^{empir}$, $\mathcal{F}_{m,v,5}^{empir}$, $\mathcal{F}_{m,d,3}^{empir}$, and $\mathcal{F}_{m,v,3}^{empir}$.

Table 6 shows the time required to calculate the distance indicators and the multiplicity measure for each case and both the benchmark and empirical transition probability matrices. The calculation has been performed on a machine with an Intel Pentium M-processor at 2.8 Ghz.

Insert Table 6 here

6.4 Analysis of results

Tables 7, 8, 9, and 10 give details of the benchmark efficient frontiers calculated on the benchmark matrices for the two distance indicators and the two cases (i.e. $\mathcal{F}_{m,d,5}^{bench}$, $\mathcal{F}_{m,v,5}^{bench}$, $\mathcal{F}_{m,d,3}^{bench}$, and $\mathcal{F}_{m,v,3}^{bench}$). It is interesting to analyze these results moving from the partition of singletons to the all-comprehensive set partition. As more classes are aggregated the multiplicity indicator improves at the price of increasing the distance indicator. This is no surprise, but it is important to analyze the size of the increments in the two indicators passing from one point to the next on these frontiers. Indeed it is possible to observe that the true partitions $\boldsymbol{\lambda}^{A,tr}$ and $\boldsymbol{\lambda}^{B,tr}$ represent a kind of "corner point" in each case. Before these key points the increase in the multiplicity measure is paired with small increments of the distance indicators. On the contrary, after those turning points every increase in the multiplicity tends to come at a price of a consistent increase in the distance.

Insert Tables 7, 8, 9, and 10 here

The previous arguments become even more evident observing Fig. 1 and Fig. 2, where the benchmark efficient frontiers are graphically represented for cases **I**. and **II**. respectively. Each figure has two panels, i.e. (a) and (b), corresponding respectively to the two optimization problems (19) and (20). Partitions $\boldsymbol{\lambda}^{A,tr}$ and $\boldsymbol{\lambda}^{B,tr}$ separate the corresponding benchmark efficient frontiers ($\mathcal{F}_{m,d,5}^{bench}$, $\mathcal{F}_{m,v,5}^{bench}$, $\mathcal{F}_{m,d,3}^{bench}$, and $\mathcal{F}_{m,v,3}^{bench}$) in two clearly different parts.

It is also possible to observe that in both cases the partitions generating the benchmark efficient frontiers show partition times (see Definition 14) mainly coinciding with the $\varepsilon-$active times.

Insert Figures 1 and 2 here

Turning to the analysis of the empirical efficient frontiers ($\mathcal{F}_{m,d,5}^{empir}$, $\mathcal{F}_{m,v,5}^{empir}$, $\mathcal{F}_{m,d,3}^{empir}$, and $\mathcal{F}_{m,v,3}^{empir}$), in Fig. 1 and Fig. 2 it is possible to observe several confirmations about the method proposed here.

First, we observe that the true partitions belong to all the four empirical efficient frontiers. This is an important acknowledgment about the consistency of our method, since it states that we have done a successful reverse-engineering of the true mechanics governing the evolution of the Markov chains designed for cases **I**. and **II**..

Second, the general shape of the empirical efficient frontiers reproduces that of the corresponding benchmark ones, with the true partitions points acting in both cases as "corner stones".

Third, it is relevant to observe that such successful result was obtained for both the distance indicators adopted here. This is evidence that, at least in our experiment, the choice between the two distance indicators is not crucial for the method to operate correctly.

Fourth, the intersection between each pair of efficient frontiers (i.e. the benchmark and the empirical frontiers paired with the same distance and the same case) is significantly large, as Table 11 shows.

Insert Table 11 here

6.5 Reduction of the set of admissible solutions and computation time

As shown in Proposition 23, the fast growing behavior of the Bell numbers increases dramatically the computational complexity of our optimization problems. This fact explains why our didactic applications **I.** and **II.** have been kept to a small size.

The reduction of computation time as a consequence of a reduction of the elements in the set of admissible solutions is a relevant issue justifying the interest towards some heuristics as a way to apply our method in real situations, where the states and the time lags can be significantly larger than in our numerical examples.

The following table shows how the computation times change in response to a reduction of the space of admissible solutions in the two cases analyzed here. In particular the reduction has been operated through a removal of some partitions, randomly selected, up to some percentages.

Insert Table 12 here

As it was expected, computation times reduce nearly proportionally with respect to the corresponding reduction in the size of the two optimization problems.

7 Conclusions

This paper proposes an optimization method for the problem of estimating the dimension of the transition probability matrix of a Markov chain for simulation and bootstrap purposes. Several aspects were to be addressed. We discussed the necessary properties of the criteria required to identify jointly the state space and the order of a Markov chain. Such discussion is of help in avoiding the development of inappropriate criteria.

We formalized our problem as a search of the partition of the states and the order of a Markov chain which minimize the distance inside each class, subject to a minimal level of multiplicity. Two alternative distance indicators were proposed, both based exclusively on the transition probabilities. The multiplicity measure is based on the cardinality of the classes of a given partition.

Several benefits originate from this approach. Since the solution of the optimization problem is completely data driven, the optimal partition of the states and the order of a Markov chain emerge without any arbitrary choice on the side of the researcher. Bootstrap and simulation methods based on the explicit estimation of the transition probabilities can therefore adopt an objective choice.

Besides, closely to information theoretical analysis of Markov chains, our distance indicators respect fully the Kolmogorov properties required to a disorder measure.

By solving our optimization problem, we obtain an efficient frontier composed of partitions of the state space of a Markov chain reflecting its evolutive structure. A numerical test has been performed and has verified the effectiveness of the method proposed here. The efficient frontiers, obtained in the two cases analyzed in the test, allow to identify the true evolutionary law governing a Markov chain.

It is important noticing that the full search over the set of admissible solutions is not computationally feasible if the state space and the order of the Markov chain are not small enough. So the introduction of heuristic methods to restrict the search among the admissible solutions is a welcome direction for future research.

Appendix A - Proofs of Propositions 6, 9, 11, and 23 and of Theorem 17

Proof of ***Proposition 6****.* Let $\boldsymbol{a}_i$ and $\boldsymbol{a}_j$ any two paths of a transition probability matrix. Since $d_{i,j}$ in (12) is a distance, then $d_{i,j} \geq 0$, and the case $d_{i,j} = 0$ is attained if and only if $P(a_z|\boldsymbol{a}_i) = P(a_z|\boldsymbol{a}_j)$, for each $a_z \in A$.

By definition, the maximum value of $d_{i,j}$ is reached when $P(a_z|\boldsymbol{a}_i) \cdot P(a_z|\boldsymbol{a}_j) = 0$, for each $a_z \in A$, and there exist two subsets of A, say A_1 and A_2, such that

$$\sum_{z_1 : a_{z_1} \in A_1} P(a_{z_1}|\boldsymbol{a}_i) = 1 \qquad \text{and} \qquad \sum_{z_2 : a_{z_2} \in A_2} P(a_{z_2}|\boldsymbol{a}_j) = 1.$$

In that case, $d_{i,j} = 2$. By definition of the distance in a class $\boldsymbol{A}_q$, introduced in (13), then also $d_{\boldsymbol{A}_q} \in [0, 2]$.

(14) gives that $d_{\boldsymbol{\lambda}}$ is a weighted mean of the distances within the classes, and this proves the result. □

Proof of ***Proposition 9***. Consider (15). Fix an a_z and focus on the variance formula

$$\sum_{i:\boldsymbol{a}_i \in \boldsymbol{A}_q} W_i \cdot \left[P(a_z|\boldsymbol{a}_i) - P(a_z|\boldsymbol{A}_q)\right]^2 \tag{22}$$

appearing inside the curly brackets; this formula is the (weighted) variance of the transition probabilities $P(a_z|\boldsymbol{a}_i)$ related to the k-paths $\boldsymbol{a}_i$ of partition class $\boldsymbol{A}_q$. We want to show that the maximum value of this variance is 0.25 and is attained when:

- the probabilities $P(a_z|\boldsymbol{a}_i)$ are either 0 or 1,
- the sum of the weights W_i assigned to the 1's is $\frac{1}{2}$,
- the sum of the weights W_i assigned to the 0's is $\frac{1}{2}$.

First, it is easily seen that (22) is maximum if and only if each element of the sum is a global maximum in its own. To this purpose, let us consider the generic element of the summation in (22), i.e. $W_i \cdot \left[P(a_z|\boldsymbol{a}_i) - P(a_z|\boldsymbol{A}_q)\right]^2$, and put, for ease of notation,

$$W_i = x, \qquad P(a_z|\boldsymbol{a}_i) = y, \qquad \text{and} \qquad P(a_z|\boldsymbol{A}_q) = k.$$

The function $z(x, y) = x(y - k)^2$ is to be maximized in the domain $[\bar{x}_d, \bar{x}_u] \times [0, 1]$, with $0 < \bar{x}_d < \bar{x}_u < 1$; indeed, the probability y cannot take values outside the interval $[0, 1]$; moreover, the weight x is allowed to take a value strictly less than 1 and greater than 0, otherwise we would have trivial solutions: if $x = 1$, then (22) is worth 0, as the addend under scrutiny is given all the potential weight and $k = y$, independently of y; on the contrary, if $x = 0$, then the addend under scrutiny would contribute with a 0 to the value of (22), independently of y, and there is no reason in considering it. Finally, notice that also the weighted average of the probabilities, k, can take only values in $[0, 1]$.
It is easy to see that, constrained to the domain $[\bar{x}_d, \bar{x}_u] \times [0, 1]$, the function z has two points of local maximum, $(\bar{x}_u; 0)$ and $(\bar{x}_u; 1)$. Depending on k, the points of global maximum can be $(\bar{x}_u; 0)$, or $(\bar{x}_u; 1)$, or both of them:

1. if $k > 0.5$, then $(\bar{x}_u; 0)$ is the only point of global maximum and $z(\bar{x}_u, 0) = \bar{x}_u(0 - k)^2 = \bar{x}_u k^2$;
2. if $k < 0.5$, then $(\bar{x}_u; 1)$ is the only point of global maximum and $z(\bar{x}_u, 1) = \bar{x}_u(1 - k)^2$;
3. if $k = 0.5$, then both $(\bar{x}_u; 0)$ and $(\bar{x}_u; 1)$ are points of global maximum and $z(\bar{x}_u, 0) = z(\bar{x}_u, 1) = \bar{x}_u \cdot 0.25$.

Remember now that k takes the same value for each addend of (22), therefore the maximization of each addend would give the same answer in terms of y's.
Remember further that $P(a_z|\boldsymbol{A}_q) = k$ is the average of the transition probabilities $P(a_z|\boldsymbol{a}_i)$ - the y's -, therefore it depends on them, and observe two facts:

a. if all the transition probabilities $P(a_z|\boldsymbol{a}_i)$ in (22) are equal either to 0 or to 1, then their average is equal either to 0 or to 1; as a consequence, there is a contradiction in choosing the optimal probabilities as in cases 1. or 2. and forcing k to be greater than 0.5 or less than 0.5, respectively;

b. on the contrary, if we look at case 3., then *there* is a way of choosing the optimal $P(a_z|\boldsymbol{a}_i)$'s to be both 0 and 1 and their average $P(a_z|\boldsymbol{A}_q)$ to be 0.5.

To this purpose, call S_1 the sum of the weights assigned to the 1's, and $S_0 = 1 - S_1$ the sum of the weights assigned to the 0's; we can write

$$P(a_z|\boldsymbol{A}_q) = S_1 \cdot 1 + S_0 \cdot 0 = S_1,$$

and conclude that, if we choose the sum of the weights assigned to the 1's to be $S_1 = 0.5$ (and, obviously, the sum of the weights assigned to the 0's to be the same), then we fulfill the features of case 3. *jointly* for *all* the addends of (22).

If we choose the probabilities $P(a_z|\boldsymbol{a}_i)$ to be both 0 and 1, with the constraint that the weight assigned to the 1's is equal to the weight assigned to the 0's, then we maximize the variance in (22), because such variance is now the sum of *jointly* globally maximized addends. In this case, it is also easily seen that the variance is worth 0.25. We now want to consider the following k-path transition probability matrix:

$$\mathcal{M} = \begin{array}{lcc} & a_1 & a_2 \\ \hline \boldsymbol{a}_1 & 0 & 1 \\ ... & 0 & 1 \\ \boldsymbol{a}_M & 0 & 1 \\ \boldsymbol{a}_{M+1} & 1 & 0 \\ ... & 1 & 0 \\ \boldsymbol{a}_{M+N} & 1 & 0 \\ \hline \end{array}$$

The rows $\boldsymbol{a}_1$ to $\boldsymbol{a}_{M+N}$ represent the possible $M+N$ blocks of length k of the observed phenomenon. We suppose that the Markov chain possesses two states, i.e. the range of the observed series is $A = \{a_1, a_2\}$. The two columns of $\mathcal{M}$ composed by 0's and 1's represent the transition probabilities of block $\boldsymbol{a}_h$ to state a_z, with $h = 1, ..., M+N$ and $z = 1, 2$ (see (2) and (3)).

In light of the previous discussion, for the variance of the two columns of transition probabilities of $\mathcal{M}$ to be maximum, the weights assigned to the transition probabilities of the first M rows have to sum to 0.5 and the same is to be true for the transition probabilities of the remaining N rows.

Let us now introduce the possibility for the rows of $\mathcal{M}$ to be partitioned. We start by considering a simple partition of the $\boldsymbol{a}_h$'s, i.e. the all-comprehensive set partition; such partition is composed by only one class collecting all the $\boldsymbol{a}_h$'s and is denoted with

$$\boldsymbol{\lambda}^a = \{\boldsymbol{A}_1\} = \{\{\boldsymbol{a}_1, ..., \boldsymbol{a}_M, \boldsymbol{a}_{M+1}, ..., \boldsymbol{a}_{M+N}\}\}.$$

By (15), the variance of $\boldsymbol{\lambda}^a$ is equal to the variance of its unique class:

$$v_{\boldsymbol{\lambda}^a} = v_{\boldsymbol{A}_1} = \frac{1}{2} \cdot (0.25 + 0.25) = 0.25;$$

the variance of $\boldsymbol{\lambda}^a$ is obtained by averaging the variances of the two columns, and by (22) each column variance is equal to 0.25.

In order to get to a *generic* transition probability matrix partitioned in a *generic* way, observe that there are two ways to modify matrix $\mathcal{M}$ and the related all-comprehensive set partition $\boldsymbol{\lambda}^a$:

i. introducing more than two columns in $\mathcal{M}$,

ii. introducing a finer partition $\boldsymbol{\lambda}$.

In both the cases, it is easy to see that $v_{\boldsymbol{\lambda}}$ decreases or, at most, does not change.

i. Suppose that we expand our matrix $\mathcal{M}$ by adding a third column; it is easily observed that, if the new column is composed by all 0's, then it does not affect the variance of the first two columns, but now the variance of the all-comprehensive set partition becomes

$$v_{\boldsymbol{\lambda}^a} = \frac{1}{3} \cdot (0.25 + 0.25 + 0) = 0.1\bar{6}.$$

If the third column collects positive numbers strictly less than 1, a corresponding reduction of the 1's in the first two columns is needed. In this way, the third column and one or both of the first two columns do not show an extreme distribution of 0's and 1's; consequently, the variance of such columns, and of the all-comprehensive set partition, cannot be 0.25.
Finally, if we want the added column to show an extreme distribution of 1's and 0's, we should allocate some 1's to this column. Remember that the only way for the weighted variance of a column to be maximum (0.25) is to assign weights whose sum is $S_1 = 0.5$ for the 1's and $S_0 = 1 - S_1 = 0.5$ for the 0's. Because these weights have to *stay fixed across the columns*, there is no way for columns 1, 2, and 3 to *jointly* have an extreme distribution and a total weight of 0.5 for their 1's and a total weight of 0.5 for their 0's. As a result, the variance of the all-comprehensive set partition will decrease.

ii. It is easy to see that each possible partition $\boldsymbol{\lambda}$ of the rows of $\mathcal{M}$ takes a value of $v_{\boldsymbol{\lambda}}$ less than or equal to the value of the all-comprehensive set partition $\boldsymbol{\lambda}^a$. This fact is easily explained by observing that (16) is a weighted average of the variances *inside* the classes of partition $\boldsymbol{\lambda}$ and does not consider the variance *between* these classes.

This completes the proof. □

Proof of ***Proposition 11****.* The absolute multiplicity indicator $l_{\boldsymbol{\lambda}}$ attains its minimum value when, for each $w = 1, \dots, k$, it results $|\lambda_w| = J_N$. In this case, the unidimensional partitions λ_w are composed by singletons, i.e. $\lambda_w = \{\{a_1\}, \dots, \{a_{J_N}\}\}$, and have maximum cardinality, and the multidimensional partition is the partition of singletons $\boldsymbol{\lambda}^s$. Given that $C = \sum_{q=1}^{|\boldsymbol{\lambda}|} |\boldsymbol{A}_q|$, (17) becomes

$$l_{\boldsymbol{\lambda}} = \sum_{q=1}^{|\boldsymbol{\lambda}|} |\boldsymbol{A}_q|^2 = \sum_{q=1}^{|\boldsymbol{\lambda}|} 1^2 = \sum_{q=1}^{|\boldsymbol{\lambda}|} |\boldsymbol{A}_q| = C.$$

Conversely, $l_{\boldsymbol{\lambda}}$ attains its maximum value when, for each $w = 1, \dots, k$, it results $|\lambda_w| = 1$, i.e. $\lambda_w = \{A\}$. The multidimensional partition is the all-comprehensive set partition $\boldsymbol{\lambda}^a = \underbrace{(\{A\}, ..., \{A\})}_{\text{k times}}$

and consists of one class represented by set A^k; in this case, we have

$$l_{\boldsymbol{\lambda}} = \sum_{q=1}^{|\boldsymbol{\lambda}|} |\boldsymbol{A}_q|^2 = \sum_{q=1}^{1} |\boldsymbol{A}_q|^2 = C^2.$$

□

*Proof of **Proposition 23**.* It is known that the number of distinct partitions of J_N elements is the number of Bell of J_N, $B(J_N)$. Combining $B(J_N)$ partitions k times, gives the number of elements in the set of admissible solutions of the optimization problems (19) and (20). This number is equal to $[B(J_N)]^k$. Let us decompose the calculations involved in the assessment of each partition $\boldsymbol{\lambda}$ in the set of admissible solutions into three parts:

(i) computation of the distance of each class of $\boldsymbol{\lambda}$;

(ii) calculation of the distance indicator of $\boldsymbol{\lambda}$ (i.e. $d_{\boldsymbol{\lambda}}$ or $v_{\boldsymbol{\lambda}}$);

(iii) calculation of the multiplicity measure $m_{\boldsymbol{\lambda}}$.

Let us enter into the details.
We first observe that the Bell number can be decomposed into a summation of Stirling numbers of the second kind, $S(J_N, z)$, which give the number of partitions that can be obtained dividing J_N elements into z classes. In particular, it is known that

$$B(J_N) = \sum_{z=1}^{J_N} S(J_N, z). \tag{23}$$

Therefore

(i) The summation in (23) recalls that all the possible unidimensional partitions of $\boldsymbol{\lambda}$ have cardinality equal to $B(J_N)$ and can be decomposed into J_N groups, where the elements in each group are the partitions with the same cardinality z (let us call it z-th Stirling class). Depending on $v_{\boldsymbol{\lambda}}$ or $d_{\boldsymbol{\lambda}}$, the computation time of the internal distances for each partition in the z-th Stirling class is proportional respectively to the following products;
for $v_{\boldsymbol{\lambda}}$

$$z \cdot S(J_N, z) \cdot \alpha_v \frac{J_N}{z} = S(J_N, z) \cdot \alpha_v J_N,$$

that is the product of the number of classes (i.e. z), the number of partitions of J_N elements into z classes (i.e. $S(J_N, z)$), and the average number of elements in each z-th Stirling class (i.e. J_N/z); α_v is a time conversion parameter depending on the machine computing power;
for $d_{\boldsymbol{\lambda}}$

$$z \cdot S(J_N, z) \cdot \alpha_d \frac{J_N}{z} \left(\frac{J_N}{z} - 1 \right) = S(J_N, z) \cdot \alpha_d J_N \left(\frac{J_N}{z} - 1 \right).$$

In this case, the computation time increases because $d_{\boldsymbol{\lambda}}$ implies an average number of comparisons among the rows contained in each class equal to $\frac{1}{2} \frac{J_N}{z} \left(\frac{J_N}{z} - 1 \right)$;

(ii) the distance indicators we adopt are weighted averages of the class distances calculated for a given partition. The average operator implies a number of calculations proportional to the number of elements to be aggregated (z in the z-th Stirling class). Therefore, the aggregation time required by the z-th Stirling class is given by:

$$\beta_1 \cdot z \cdot S(J_N, z),$$

where $\beta_1 > 0$ is a time conversion factor;

(iii) turning to the calculation of the multiplicity measure for the z-th Stirling class, observe that it is required to calculate the square value of z terms (i.e. the cardinality of each class), so the computation time can be written as:

$$\beta_2 \cdot z \cdot S(J_N, z),$$

where $\beta_1 > 0$ is, as usual, a time conversion factor.

Recalling the Stirling decomposition in (23) and combining the computation times in the previous points, the time required to accomplish all the calculations for an entire partition of J_N elements is, in the case of $v_{\boldsymbol{\lambda}}$,

$$\sum_{z=1}^{J_N} (\alpha_v J_N + \beta z) \cdot S(J_N, z),$$

where $\beta = \beta_1 + \beta_2$ is a time conversion parameter following from those in points (ii) and (iii), and

$$\sum_{z=1}^{J_N} \left[\alpha_d J_N \left(\frac{J_N}{z} - 1 \right) + \beta z \right] \cdot S(J_N, z),$$

in the case of $d_{\boldsymbol{\lambda}}$.

Taking the average time for a partition gives, in the two cases:

$$\frac{1}{\sum_{z=1}^{J_N} S(J_N, z)} \sum_{z=1}^{J_N} S(J_N, z) \cdot (\alpha_v J_N + \beta z) \approx \alpha_v J_N,$$

as $J_N \to +\infty$ for $v_{\boldsymbol{\lambda}}$, and

$$\frac{1}{\sum_{z=1}^{J_N} S(J_N, z)} \sum_{z=1}^{J_N} S(J_N, z) \cdot \left[\alpha_d J_N \left(\frac{J_N}{z} - 1 \right) + \beta z \right] \approx \alpha_d J_N^2,$$

as $J_N \to +\infty$ for $d_{\boldsymbol{\lambda}}$.

In other words, the average time to process a partition is proportional to the number of its elementary states (i.e. the number of the rows of the transition probability matrix) in the case of $v_{\boldsymbol{\lambda}}$ and to the square of this number in the case of $d_{\boldsymbol{\lambda}}$. Since the combinations of partitions which can be obtained using k time lags increases with the k-th power of $B(J_N)$ and the number of rows in the transition probability matrix increases with the k-th power of J_N, the expected calculation time required to span the set of admissible solutions is proportional to $[\alpha_v J_N B(J_N)]^k$ in the case of $v_{\boldsymbol{\lambda}}$

and to $[\alpha_d J_N^2 B(J_N)]^k$ in the case of $d_{\boldsymbol{\lambda}}$. Concluding the proof, we have

$$[\alpha_v J_N B(J_N)]^k = O([J_N B(J_N)]^k)$$

for $v_{\boldsymbol{\lambda}}$ and

$$[\alpha_d J_N^2 B(J_N)]^k = O([J_N^2 B(J_N)]^k)$$

for $d_{\boldsymbol{\lambda}}$ as $J_N \to +\infty$. □

*Proof of **Theorem 17**.* 1. By hypothesis *a.*, we have:

$$|\boldsymbol{\lambda}| = |\lambda_1| \cdot ... \cdot |\lambda_{w-1}| \cdot |\lambda_w| \cdot |\lambda_{w+1}| \cdot ... \cdot |\lambda_k|$$

$$= |\lambda_1| \cdot ... \cdot |\lambda_{w-1}| \cdot 1 \cdot |\lambda_{w+1}| \cdot ... \cdot |\lambda_k| = |\boldsymbol{\lambda}^{(-w)}|.$$

2. We prove the result only for the distance indicator $d_{\boldsymbol{\lambda}}$, being the case of $v_{\boldsymbol{\lambda}}$ analogous. Hypothesis *b.* can be equivalently stated as in the following: for any $a_z \in A$ and any k-path $\boldsymbol{a}_h$, the probability

$$P(a_z|\boldsymbol{a}_h) = P(a_z|(a_{h,k}, \dots, a_{h,w+1}, a_{h,w}, a_{h,w-1}, \dots, a_{h,1}))$$

is independent from the value of $a_{h,w}$. Therefore:

$$P(a_z|(a_{h,k}, \dots, a_{h,w+1}, a_{h,w}, a_{h,w-1}, \dots, a_{h,1})) = P(a_z|(a_{h,k}, \dots, a_{h,w+1}, a_{h,w-1}, \dots, a_{h,1})). \tag{24}$$

By hypothesis *a.* we have $\lambda_w = \{A\}$, so that each class of $\boldsymbol{\lambda}$ can be written as:

$$\boldsymbol{A}_q = A_{q_k,k} \times \cdots \times A_{q_{w+1},w+1} \times A \times A_{q_{w-1},w-1} \times \cdots \times A_{q_1,1}. \tag{25}$$

Hence, there is a relation between the classes of $\boldsymbol{\lambda}$ and those of $\boldsymbol{\lambda}^{(-w)}$ according to (25). For ease of exposition, we set:

$$\boldsymbol{A}_q = \boldsymbol{A}_q^{(-w)} \times A,$$

where

$$\boldsymbol{A}_q^{(-w)} = A_{q_k,k} \times \cdots \times A_{q_{w+1},w+1} \times A_{q_{w-1},w-1} \times \cdots \times A_{q_1,1}.$$

By (24) and (13), we have

$$d_{\boldsymbol{A}_q} = d_{\boldsymbol{A}_q^{(-w)}}. \tag{26}$$

Moreover

$$|\boldsymbol{A}_q| = |\boldsymbol{A}_q^{(-w)}| \cdot |A| = |\boldsymbol{A}_q^{(-w)}| \cdot J_N. \tag{27}$$

By point 1., (14), (26), and (27), we obtain:

$$d_{\boldsymbol{\lambda}} = \frac{1}{(J_N)^k} \cdot \sum_{q=1}^{|\boldsymbol{\lambda}|} d_{\boldsymbol{A}_q} \cdot |\boldsymbol{A}_q|$$

$$= \frac{1}{(J_N)^k} \cdot \sum_{q=1}^{|\boldsymbol{\lambda}^{(-w)}|} d_{\boldsymbol{A}_q^{(-w)}} \cdot |\boldsymbol{A}_q^{(-w)}| \cdot J_N$$

$$= \frac{1}{(J_N)^{k-1}} \cdot \sum_{q=1}^{|\boldsymbol{\lambda}_{(-w)}|} d_{\boldsymbol{A}_q^{(-w)}} \cdot |\boldsymbol{A}_q^{(-w)}| = d_{\boldsymbol{\lambda}^{(-w)}}.$$

□

Tables

Table 1: Cardinality of the set of admissible solutions for various combinations of time lags k and states J_N of a Markov chain.

	Time lags (k)						
States (J_N)	1	2	3	4	5	6	7
1	1	1	1	1	1	1	1
2	2	4	8	16	32	64	128
3	5	25	125	625	3,125	15,625	78,125
4	15	225	3,375	50,625	759,375	11,390,625	170,859,375
5	52	2,704	140,608	7,311,616	380,204,032	19,770,609,664	1,028,071,702,528
6	203	41,209	8,365,427	1,698,181,681	344,730,881,243	69,980,368,892,329	14,206,014,885,142,800
7	877	769,129	674,526,133	591,559,418,641	518,797,610,148,157	454,985,504,099,934,000	399,022,287,095,642,000,000

This table reports the cardinality of the set of admissible solutions of the two optimization problems (19) and (20) for a Markov chain of order k and with J_N states, $k, J_N \in \{1, 2, 3, 4, 5, 6, 7\}$. See also footnote 1.

Table 2: True partition $\boldsymbol{\lambda}^{A,tr}$ associated to the 5-path transition probability matrix $\mathcal{A}^{bench}$.

$\lambda_5^{A,tr}$	$\lambda_4^{A,tr}$	$\lambda_3^{A,tr}$	$\lambda_2^{A,tr}$	$\lambda_1^{A,tr}$
$\boxed{\begin{matrix}1\\2\\3\end{matrix}}$	$\boxed{\begin{matrix}1\\2\\3\end{matrix}}$	$\boxed{1}$ $\boxed{2}$ $\boxed{3}$	$\boxed{1}$ $\boxed{2}$ $\boxed{3}$	$\boxed{\begin{matrix}1\\2\\3\end{matrix}}$

This table refers to the true partition $\boldsymbol{\lambda}^{A,tr}$=($\lambda_5^{A,tr}, \lambda_4^{A,tr}, \lambda_3^{A,tr}, \lambda_2^{A,tr}, \lambda_1^{A,tr}$) designed for case I.. Transition probabilities have been allocated in matrix $\mathcal{A}^{bench}$ so that keeping all the 3 states of the process together at time lags 5, 4, and 1, while separating them in three sets at time lags 3 and 2, will result in partition classes populated by 5-paths with highly similar transition probabilities. See also the next Table 4, which shows the average transition probabilities of the 5-paths belonging to each class of $\boldsymbol{\lambda}^{A,tr}$.

Table 3: True partition $\boldsymbol{\lambda}^{B,tr}$ associated to the 3-path transition probability matrix $\mathcal{B}^{bench}$.

$\lambda_3^{B,tr}$	$\lambda_2^{B,tr}$	$\lambda_1^{B,tr}$
1	1	1
2	2	2
3	3	3
4	4	4
5	5	5

This table refers to the true partition $\boldsymbol{\lambda}^{B,tr}$=($\lambda_3^{B,tr}, \lambda_2^{B,tr}, \lambda_1^{B,tr}$) designed for case II.. Transition probabilities have been allocated in matrix $\mathcal{B}^{bench}$ so that keeping all the 5 states of the process together at time lag 3, while separating them in two sets at time lag 2, {1,2} and $\{3, 4, 5\}$ respectively, and in three sets at time lag 1, i.e. {1,2}, {3,4}, and {5}, will result in partition classes populated by 3-paths with highly similar transition probabilities. See also the next Table 5, which shows the average transition probabilities of the 3-paths belonging to each class of $\boldsymbol{\lambda}^{B,tr}$.

Table 4: Average transition probabilities characterizing the true partition $\boldsymbol{\lambda}^{A,tr}$ associated to the 5-path transition probability matrix $\mathcal{A}^{bench}$.

					y_t		
y_{t-5}	y_{t-4}	y_{t-3}	y_{t-2}	y_{t-1}	1	2	3
-	-	1	1	-	0.137	0.164	**0.699**
-	-	1	2	-	**0.780**	0.118	0.102
-	-	1	3	-	**0.791**	0.106	0.104
-	-	2	1	-	0.110	**0.780**	0.111
-	-	2	2	-	**0.778**	0.106	0.116
-	-	2	3	-	**0.785**	0.101	0.113
-	-	3	1	-	0.105	**0.791**	0.104
-	-	3	2	-	**0.786**	0.111	0.103
-	-	3	3	-	0.116	**0.787**	0.097

This table refers to case I. (matrix $\mathcal{A}^{bench}$) and represents the classes of the true partition $\boldsymbol{\lambda}^{A,tr}$ through the average transition probabilities of its 5-paths.

Each row represents a 5-path observed at active times *t*-3 and *t*-2, irrespective of the values at times *t*-5, *t*-4, and *t*-1.

The transition probabilities in each row are obtained averaging the corresponding 27 rows of transition probabilities in matrix $\mathcal{A}^{bench}$. Indeed, for each couple of values y_{t-2} and y_{t-3} chosen in the set $\{1, 2, 3\}$, 27 alternative 5-paths can be obtained by letting y_{t-5}, y_{t-4}, and y_{t-1} vary in the same set (the 3 values the process can take for each of the 3 "non critical" time lags).

To help have a fast view of the "mechanics" of the process, average transition probabilities greater than 0.7 are reported in bold.

At time *t*-2 state 1 should be separated from states 2 and 3, look, for example, at the first three rows of average transition probabilities.

For the same time lag, states 2 and 3 cannot be put together, see the last two rows of average transition probabilities.

Similar arguments also apply for time lag 3, where states 1, 2, and 3 should be kept separated.

Table 5: Average transition probabilities characterizing the true partition $\boldsymbol{\lambda}^{B,tr}$ associated to the 3-path transition probability matrix $\mathcal{B}^{bench}$.

			y_t				
y_{t-3}	y_{t-2}	y_{t-1}	1	2	3	4	5
-	1	1	**0.366**	**0.239**	0.107	0.093	0.195
-	1	2	**0.362**	**0.236**	0.106	0.102	0.194
-	2	1	**0.360**	**0.228**	0.104	0.114	0.194
-	2	2	**0.365**	**0.234**	0.107	0.098	0.196
-	1	3	**0.356**	**0.236**	**0.303**	0.060	0.046
-	1	4	**0.371**	**0.237**	**0.307**	0.041	0.045
-	2	3	**0.370**	**0.230**	**0.303**	0.052	0.045
-	2	4	**0.370**	**0.236**	**0.306**	0.042	0.046
-	1	5	**0.366**	**0.240**	0.024	0.025	**0.345**
-	2	5	**0.372**	**0.240**	0.026	0.018	**0.343**
-	3	1	0.102	**0.286**	**0.204**	**0.362**	0.046
-	3	2	0.106	**0.290**	**0.206**	**0.355**	0.044
-	4	1	0.105	**0.286**	**0.203**	**0.362**	0.044
-	4	2	0.106	**0.279**	**0.206**	**0.365**	0.044
-	5	1	0.104	**0.285**	**0.206**	**0.360**	0.045
-	5	2	0.105	**0.279**	**0.202**	**0.370**	0.044
-	3	3	0.106	**0.289**	**0.455**	0.108	0.042
-	3	4	0.105	**0.286**	**0.454**	0.111	0.044
-	4	3	0.107	**0.277**	**0.456**	0.115	0.045
-	4	4	0.105	**0.291**	**0.453**	0.108	0.043
-	5	3	0.104	**0.282**	**0.453**	0.117	0.045
-	5	4	0.103	**0.284**	**0.457**	0.112	0.044
-	3	5	0.105	**0.286**	**0.408**	0.060	0.142
-	4	5	0.106	**0.285**	**0.404**	0.059	0.146
-	5	5	0.107	**0.292**	**0.406**	0.049	0.147

This table refers to case II. (matrix $\mathcal{B}^{bench}$) and represents the classes of the true partition $\boldsymbol{\lambda}^{B,tr}$
through the average transition probabilities of its 3-paths.
Each row represents a 3-path observed at active times *t*-2 and *t*-1, irrespective of the values at time *t*-3.
The transition probabilities in each row are obtained averaging the corresponding 5 rows of transition probabilities
in matrix $\mathcal{B}^{bench}$. Indeed, for each couple of values y_{t-2} and y_{t-1} chosen in the set $\{1,2,3,4,5\}$, 5 alternative 3-paths
can be obtained by letting y_{t-3} vary in the same set.
To help have a fast view of the "mechanics" of the process, average transition probabilities greater than 0.2 are reported in bold.
The first four classes of the partition, separated by horizontal lines, are clearly identified in terms of average transition probabilities.
Classes 5 and 6 of the partition seem to show the same average transition probabilities, although a difference
can be spot in the last two columns showing that class 5 mainly evolves to state 4, while class 6 mainly goes to state 5.

Table 6: Computation time of the distance indicators $d_{\boldsymbol{\lambda}^A}/d_{\boldsymbol{\lambda}^B}$ and $v_{\boldsymbol{\lambda}^A}/v_{\boldsymbol{\lambda}^B}$ and the multiplicity measure $m_{\boldsymbol{\lambda}^A}/m_{\boldsymbol{\lambda}^B}$ for the partitions $\boldsymbol{\lambda}^A$ of case I. and the partitions $\boldsymbol{\lambda}^B$ of case II..

Case	Transition probability matrix	Computation time of $d_{\boldsymbol{\lambda}^A}/d_{\boldsymbol{\lambda}^B}$, $v_{\boldsymbol{\lambda}^A}/v_{\boldsymbol{\lambda}^B}$, and $m_{\boldsymbol{\lambda}^A}/m_{\boldsymbol{\lambda}^B}$
I.	$\mathcal{A}^{bench}$	92 secs
	$\mathcal{A}^{empir}$	37 secs
II.	$\mathcal{B}^{bench}$	$3,123$ secs
	$\mathcal{B}^{empir}$	$2,031$ secs

Rows 1 and 2 refer to the numerical experiments of case I. based on a set of admissible solutions with $3,125$ partitions.

Rows 3 and 4 report the computation time in case II., where the set of admissible solutions has $140,608$ partitions.

Table 7: Benchmark efficient frontier $\mathcal{F}^{bench}_{m,d,5}$.

		Solutions $\boldsymbol{\lambda}^{A,*} = (\lambda_5^{A,*}, \lambda_4^{A,*}, \lambda_3^{A,*}, \lambda_2^{A,*}, \lambda_1^{A,*})$ generating $\mathcal{F}^{bench}_{m,d,5}$					Partition
$m_{\boldsymbol{\lambda}^{A,*}}$	$d_{\boldsymbol{\lambda}^{A,*}}$	$\lambda_5^{A,*}$	$\lambda_4^{A,*}$	$\lambda_3^{A,*}$	$\lambda_2^{A,*}$	$\lambda_1^{A,*}$	times
0	0	{{1},{2},{3}}	{{1},{2},{3}}	{{1},{2},{3}}	{{1},{2},{3}}	{{1},{2},{3}}	5,4,3,2,1
0.01995	0.04889	{{1},{2},{3}}	{{1,3},{2}}	{{1},{2},{3}}	{{1},{2},{3}}	{{1},{2},{3}}	5,4,3,2,1
0.04570	0.08840	{{1,3},{2}}	{{1,3},{2}}	{{1},{2},{3}}	{{1},{2},{3}}	{{1},{2},{3}}	5,4,3,2,1
0.07894	0.12321	{{1,3},{2}}	{{1,3},{2}}	{{1},{2},{3}}	{{1},{2},{3}}	{{1,3},{2}}	5,4,3,2,1
0.08473	0.18514	{{1,2,3}}	{{1},{2,3}}	{{1},{2},{3}}	{{1},{2},{3}}	{{1},{2},{3}}	4,3,2,1
0.12933	0.20840	{{1,2,3}}	{{1},{2,3}}	{{1},{2},{3}}	{{1},{2},{3}}	{{1,3},{2}}	4,3,2,1
0.13709	0.22052	{{1,2,3}}	{{1,2,3}}	{{1},{2},{3}}	{{1},{2},{3}}	{{1},{2},{3}}	3,2,1
0.19694	0.23800	{{1,2,3}}	{{1,2,3}}	{{1},{2},{3}}	{{1},{2},{3}}	{{1,3},{2}}	3,2,1
0.28764	0.27756	{{1,2,3}}	{{1,2,3}}	{{1},{2},{3}}	{{1},{2},{3}}	{{1,2,3}}	3,2
0.39128	0.56533	{{1,2,3}}	{{1,2,3}}	{{1,2},{3}}	{{1},{2},{3}}	{{1,2,3}}	3,2
0.52509	0.87978	{{1,2,3}}	{{1,2,3}}	{{1,2},{3}}	{{1,3},{2}}	{{1,2,3}}	3,2
0.54838	1.17200	{{1,2,3}}	{{1,2,3}}	{{1,2,3}}	{{1},{2},{3}}	{{1,2,3}}	2
0.72790	1.19733	{{1,2,3}}	{{1,2,3}}	{{1,2,3}}	{{1,3},{2}}	{{1,2,3}}	2
1	1.67200	{{1,2,3}}	{{1,2,3}}	{{1,2,3}}	{{1,2,3}}	{{1,2,3}}	-

Column "$m_{\boldsymbol{\lambda}^{A,*}}$" lists the values of the multiplicity measure defined in (18).

Column "$d_{\boldsymbol{\lambda}^{A,*}}$" lists the values of the distance indicator defined in (14).

Columns "$\lambda_5^{A,*}$", "$\lambda_4^{A,*}$", "$\lambda_3^{A,*}$", "$\lambda_2^{A,*}$", and "$\lambda_1^{A,*}$" show the partitions generating the benchmark efficient frontier $\mathcal{F}^{bench}_{m,d,5}$.

Each solution $\boldsymbol{\lambda}^{A,*}$ is displayed through the 1-dimensional partitions $\lambda_5^{A,*}$, $\lambda_4^{A,*}$, $\lambda_3^{A,*}$, $\lambda_2^{A,*}$, and $\lambda_1^{A,*}$ of the time series values - 1, 2, and 3 - for each of the $\bar{k} = 5$ time lags. The benchmark efficient frontier is the output of the optimization procedure described in Subsection 6.3. In particular, optimization problem (19) has been solved according to the 3 step procedure presented at the end of Section 5 and based on the 5-path transition probability matrix $\mathcal{A}^{bench}$ described in Subsection 6.1.

The last column reports the partition times (see Definition 14). $\mathcal{F}^{bench}_{m,d,5}$ is plotted in Fig. 1.

Table 8: Benchmark efficient frontier $\mathcal{F}^{bench}_{m,v,5}$.

		Solutions $\boldsymbol{\lambda}^{A,*} = (\lambda_5^{A,*}, \lambda_4^{A,*}, \lambda_3^{A,*}, \lambda_2^{A,*}, \lambda_1^{A,*})$ generating $\mathcal{F}^{bench}_{m,v,5}$					Partition
$m_{\boldsymbol{\lambda}^{A,*}}$	$v_{\boldsymbol{\lambda}^{A,*}}$	$\lambda_5^{A,*}$	$\lambda_4^{A,*}$	$\lambda_3^{A,*}$	$\lambda_2^{A,*}$	$\lambda_1^{A,*}$	times
0	0	{{1},{2},{3}}	{{1},{2},{3}}	{{1},{2},{3}}	{{1},{2},{3}}	{{1},{2},{3}}	5,4,3,2,1
0.01995	0.00018	{{1},{2},{3}}	{{1,3},{2}}	{{1},{2},{3}}	{{1},{2},{3}}	{{1},{2},{3}}	5,4,3,2,1
0.04570	0.00029	{{1,3},{2}}	{{1},{2},{3}}	{{1},{2},{3}}	{{1},{2},{3}}	{{1,3},{2}}	5,4,3,2,1
0.07894	0.00038	{{1,3},{2}}	{{1,3},{2}}	{{1},{2},{3}}	{{1},{2},{3}}	{{1,3},{2}}	5,4,3,2,1
0.08473	0.00090	{{1,3},{2}}	{{1},{2},{3}}	{{1},{2},{3}}	{{1},{2},{3}}	{{1,2,3}}	5,4,3,2
0.12933	0.00094	{{1,3},{2}}	{{1,3},{2}}	{{1},{2},{3}}	{{1},{2},{3}}	{{1,2,3}}	5,4,3,2
0.13709	0.00102	{{1},{2},{3}}	{{1,2,3}}	{{1},{2},{3}}	{{1},{2},{3}}	{{1,2,3}}	5,3,2
0.19694	0.00103	{{1},{2,3}}	{{1,2,3}}	{{1},{2},{3}}	{{1},{2},{3}}	{{1,2,3}}	5,3,2
0.28764	0.00106	{{1,2,3}}	{{1,2,3}}	{{1},{2},{3}}	{{1},{2},{3}}	{{1,2,3}}	3,2
0.39128	0.01451	{{1,2,3}}	{{1,2,3}}	{{1,2},{3}}	{{1},{2},{3}}	{{1,2,3}}	3,2
0.52509	0.02632	{{1,2,3}}	{{1,2,3}}	{{1},{2,3}}	{{1},{2,3}}	{{1,2,3}}	3,2
0.54838	0.04197	{{1,2,3}}	{{1,2,3}}	{{1,2,3}}	{{1},{2},{3}}	{{1,2,3}}	2
0.72790	0.04727	{{1,2,3}}	{{1,2,3}}	{{1,2,3}}	{{1},{2,3}}	{{1,2,3}}	2
1	0.08247	{{1,2,3}}	{{1,2,3}}	{{1,2,3}}	{{1,2,3}}	{{1,2,3}}	-

Column "$m_{\boldsymbol{\lambda}^{A,*}}$" lists the values of the multiplicity measure defined in (18).

Column "$v_{\boldsymbol{\lambda}^{A,*}}$" lists the values of the distance indicator defined in (16).

Columns "$\lambda_5^{A,*}$", "$\lambda_4^{A,*}$", "$\lambda_3^{A,*}$", "$\lambda_2^{A,*}$", and "$\lambda_1^{A,*}$" show the solutions generating the benchmark efficient frontier $\mathcal{F}^{bench}_{m,v,5}$.
Each solution $\boldsymbol{\lambda}^{A,*}$ is displayed through the 1-dimensional partitions $\lambda_5^{A,*}$, $\lambda_4^{A,*}$, $\lambda_3^{A,*}$, $\lambda_2^{A,*}$, and $\lambda_1^{A,*}$
of the time series values - 1, 2, and 3 - for each of the $\bar{k} = 5$ time lags. The benchmark efficient frontier is the output
of the optimization procedure described in Subsection 6.3. In particular, optimization problem (20) has been solved
according to the 3 step procedure presented at the end of Section 5 and based on the 5-path transition probability matrix $\mathcal{A}^{bench}$
described in Subsection 6.1.

The last column reports the partition times (see Definition 14). $\mathcal{F}^{bench}_{m,v,5}$ is plotted in Fig. 1.

Table 9: Benchmark efficient frontier $\mathcal{F}^{bench}_{m,d,3}$.

		Solutions $\boldsymbol{\lambda}^{B,*} = (\lambda_3^{B,*}, \lambda_2^{B,*}, \lambda_1^{B,*})$ generating $\mathcal{F}^{bench}_{m,d,3}$			Partition
$m_{\boldsymbol{\lambda}^{B,*}}$	$d_{\boldsymbol{\lambda}^{B,*}}$	$\lambda_3^{B,*}$	$\lambda_2^{B,*}$	$\lambda_1^{B,*}$	times
0	0	{{1},{2},{3},{4},{5}}	{{1},{2},{3},{4},{5}}	{{1},{2},{3},{4},{5}}	3,2,1
0.01800	0.01069	{{1,2},{3},{4},{5}}	{{1},{2},{3},{4},{5}}	{{1},{2},{3},{4},{5}}	3,2,1
0.03929	0.02093	{{1,2},{3},{4},{5}}	{{1},{2},{3,4},{5}}	{{1},{2},{3},{4},{5}}	3,2,1
0.04747	0.02424	{{1,2,5},{3,4}}	{{1},{2},{3},{4},{5}}	{{1},{2},{3},{4},{5}}	3,2,1
0.06449	0.03069	{{1,2},{3},{4},{5}}	{{1},{2},{3,4},{5}}	{{1},{2},{3,4},{5}}	3,2,1
0.07416	0.03114	{{1,2},{3},{4},{5}}	{{1},{2},{3,4,5}}	{{1},{2},{3},{4},{5}}	3,2,1
0.10575	0.03941	{{1,2},{3},{4},{5}}	{{1},{2},{3,4,5}}	{{1,2},{3},{4},{5}}	3,2,1
0.11787	0.04022	{{1,2,5},{3,4}}	{{1},{2},{3,4,5}}	{{1},{2},{3},{4},{5}}	3,2,1
0.13306	0.04760	{{1,2},{3},{4,5}}	{{1},{2},{3,4,5}}	{{1,2},{3},{4},{5}}	3,2,1
0.15747	0.04864	{{1,2,5},{3},{4}}	{{1},{2},{3,4,5}}	{{1},{2},{3,4},{5}}	3,2,1
0.17042	0.05216	{{1,2,4,5},{3}}	{{1},{2},{3,4,5}}	{{1},{2},{3},{4},{5}}	3,2,1
0.17974	0.05760	{{1,2,3},{4,5}}	{{1},{2},{3,4,5}}	{{1,2},{3},{4},{5}}	3,2,1
0.19170	0.05765	{{1,2,5},{3},{4}}	{{1},{2},{3,4,5}}	{{1,2},{3,4},{5}}	3,2,1
0.21964	0.05877	{{1,2,4,5},{3}}	{{1},{2},{3,4,5}}	{{1},{2},{3,4},{5}}	3,2,1
0.22756	0.06616	{1,2,3,4,5}	{{1},{2},{3,4,5}}	{{1},{2},{3},{4},{5}}	2,1
0.26220	0.06675	{{1,2,4,5},{3}}	{{1},{2},{3,4,5}}	{{1,2},{3,4},{5}}	3,2,1
0.28725	0.07280	{1,2,3,4,5}	{{1},{2},{3,4,5}}	{{1,2},{3},{4},{5}}	2,1
0.29360	0.07405	{{1,2,4,5},{3}}	{{1,2},{3,4,5}}	{{1,2},{3,4},{5}}	3,2,1
0.32083	0.07888	{1,2,3,4,5}	{{1,2},{3,4,5}}	{{1,2},{3},{4},{5}}	2,1
0.33886	0.07992	{1,2,3,4,5}	{{1},{2},{3,4,5}}	{{1,2},{3,4},{5}}	2,1
0.37694	0.08624	{1,2,3,4,5}	{{1,2},{3,4,5}}	{{1,2},{3,4},{5}}	2,1
0.38499	0.28608	{1,2,3,4,5}	{{1},{2},{3,4,5}}	{{1},{2},{3,4,5}}	2,1
0.42709	0.29192	{1,2,3,4,5}	{{1,2},{3,4,5}}	{{1},{2},{3,4,5}}	2,1
0.47285	0.29736	{1,2,3,4,5}	{{1,2},{3,4,5}}	{{1,2},{3,4,5}}	2,1
0.50249	0.44792	{1,2,3,4,5}	{{1},{2},{3,4,5}}	{{1,2,3,4},{5}}	2,1
0.55483	0.45344	{1,2,3,4,5}	{{1,2},{3,4,5}}	{{1,2,3,4},{5}}	2,1
0.56071	0.67600	{1,2,3,4,5}	{{1,2},{3,4},{5}}	{1,2,3,4,5}	2
0.63025	0.67920	{1,2,3,4,5}	{{1},{2},{3,4,5}}	{1,2,3,4,5}	2
0.69372	0.68160	{1,2,3,4,5}	{{1,2},{3,4,5}}	{1,2,3,4,5}	2
0.80739	0.88680	{1,2,3,4,5}	{1,2,3,4,5}	{{1,2,3,4},{5}}	1
1	1.16200	{1,2,3,4,5}	{1,2,3,4,5}	{1,2,3,4,5}	-

Column "$m_{\boldsymbol{\lambda}^{B,*}}$" lists the values of the multiplicity measure defined in (18).

Column "$d_{\boldsymbol{\lambda}^{B,*}}$" lists the values of the distance indicator defined in (14).

Columns "$\lambda_3^{B,*}$", "$\lambda_2^{B,*}$", and "$\lambda_1^{B,*}$" show the solutions generating the benchmark efficient frontier $\mathcal{F}^{bench}_{m,d,3}$.

Each solution $\boldsymbol{\lambda}^{B,*}$ is displayed through the 1-dimensional partitions $\lambda_3^{B,*}$, $\lambda_2^{B,*}$, and $\lambda_1^{B,*}$ of the time series values - 1, 2, 3, 4, and 5 - for each of the $\bar{k} = 3$ time lags. The benchmark efficient frontier is the output of the optimization procedure described in Subsection 6.3. In particular, optimization problem (19) has been solved according to the 3 step procedure presented at the end of Section 5 and based on the 3-path transition probability matrix $\mathcal{B}^{bench}$ described in Subsection 6.1.

The last column reports the partition times (see Definition 14). $\mathcal{F}^{bench}_{m,d,3}$ is plotted in Fig. 2.

Table 10: Benchmark efficient frontier $\mathcal{F}^{bench}_{m,v,3}$.

		Solutions $\boldsymbol{\lambda}^{B,*} = (\lambda_3^{B,*}, \lambda_2^{B,*}, \lambda_1^{B,*})$ generating $\mathcal{F}^{bench}_{m,v,3}$			Partition
$m_{\boldsymbol{\lambda}^{B,*}}$	$v_{\boldsymbol{\lambda}^{B,*}}$	$\lambda_3^{B,*}$	$\lambda_2^{B,*}$	$\lambda_1^{B,*}$	times
0	0	{{1},{2},{3},{4},{5}}	{{1},{2},{3},{4},{5}}	{{1},{2},{3},{4},{5}}	3,2,1
0.04747	0.00001	{{1},{2},{3},{4},{5}}	{{1},{2},{3,4,5}}	{{1},{2},{3},{4},{5}}	3,2,1
0.11787	0.00002	{{1,2,5},{3},{4}}	{{1},{2},{3,4,5}}	{{1},{2},{3},{4},{5}}	3,2,1
0.17042	0.00003	{{1,2,3,5},{4}}	{{1},{2},{3,4,5}}	{{1},{2},{3},{4},{5}}	3,2,1
0.28725	0.00004	{1,2,3,4,5}	{{1},{2},{3,4,5}}	{{1,2},{3},{4},{5}}	2,1
0.37694	0.00005	{1,2,3,4,5}	{{1,2},{3,4,5}}	{{1,2},{3,4},{5}}	2,1
0.47285	0.00209	{1,2,3,4,5}	{{1,2},{3,4,5}}	{{1,2},{3,4,5}}	2,1
0.55483	0.00408	{1,2,3,4,5}	{{1,2},{3,4,5}}	{{1,2,3,4},{5}}	2,1
0.69372	0.00612	{1,2,3,4,5}	{{1,2},{3,4,5}}	{1,2,3,4,5}	2
0.80739	0.00998	{1,2,3,4,5}	{{1,3,4,5},{2}}	{1,2,3,4,5}	2
1	0.01235	{1,2,3,4,5}	{1,2,3,4,5}	{1,2,3,4,5}	-

Column "$m_{\boldsymbol{\lambda}^{B,*}}$" lists the values of the multiplicity measure defined in (18).
Column "$d_{\boldsymbol{\lambda}^{B,*}}$" lists the values of the distance indicator defined in (16).
Columns "$\lambda_3^{B,*}$", "$\lambda_2^{B,*}$", and "$\lambda_1^{B,*}$" show the solutions generating the benchmark efficient frontier $\mathcal{F}^{bench}_{m,v,3}$.
Each solution $\boldsymbol{\lambda}^{B,*}$ is displayed through the 1-dimensional partitions $\lambda_3^{B,*}$, $\lambda_2^{B,*}$, and $\lambda_1^{B,*}$ of the time series values - 1, 2, 3, 4, and 5 - for each of the $\bar{k} = 3$ time lags. The benchmark efficient frontier is the output of the optimization procedure described in Subsection 6.3. In particular, optimization problem (20) has been solved according to the 3 step procedure presented at the end of Section 5 and based on the 3-path transition probability matrix $\mathcal{B}^{bench}$ described in Subsection 6.1.
The last column reports the partition times (see Definition 14). $\mathcal{F}^{bench}_{m,v,3}$ is plotted in Fig. 2.

Table 11: Partitions generating both the benchmark and the empirical efficient frontiers.

Case	Efficient frontier	Number of partitions generating the efficient frontier	Number of partitions generating both the benchmark and the empirical efficient frontiers
I.	$\mathcal{F}^{bench}_{m,d,5}$	14	
	$\mathcal{F}^{empir}_{m,d,5}$	40	7 (50% of benchmark)
	$\mathcal{F}^{bench}_{m,v,5}$	14	
	$\mathcal{F}^{empir}_{m,v,5}$	28	10 (71% of benchmark)
II.	$\mathcal{F}^{bench}_{m,d,3}$	31	
	$\mathcal{F}^{empir}_{m,d,3}$	73	9 (29% of benchmark)
	$\mathcal{F}^{bench}_{m,v,3}$	11	
	$\mathcal{F}^{empir}_{m,v,3}$	44	5 (45% of benchmark)

Table 12: Computation time of the distance indicators and the multiplicity measure for the partitions $\boldsymbol{\lambda}^A$ of case I. and the partitions $\boldsymbol{\lambda}^B$ of case II. in case of a reduction of the set of admissible solutions.

	Computation time					
Size of the set	Case **I.** with matrix $\mathcal{A}^{empir}$			Case **II.** with matrix $\mathcal{B}^{empir}$		
of admissible solutions	Number of partitions	Secs	% reduction	Number of partitions	Secs	% reduction
100%	$3,125$	37	-	$140,608$	$2,031$	-
90%	$2,812$	16	56.8%	$126,542$	807	60%
50%	$1,562$	8	78.4%	$70,302$	470	76.9%
10%	312	1	97.3%	$1,412$	6	99.9%

Computation times of the two distance indicators $d_{\boldsymbol{\lambda}^A}/d_{\boldsymbol{\lambda}^B}$ and $v_{\boldsymbol{\lambda}^A}/v_{\boldsymbol{\lambda}^B}$ and of the multiplicity indicator $m_{\boldsymbol{\lambda}^A}/m_{\boldsymbol{\lambda}^B}$ in cases I. and II. if the empirical matrices are selected.
The last three rows show the computation time of distances and multiplicity for randomly reduced sets of admissible solutions.

Figures

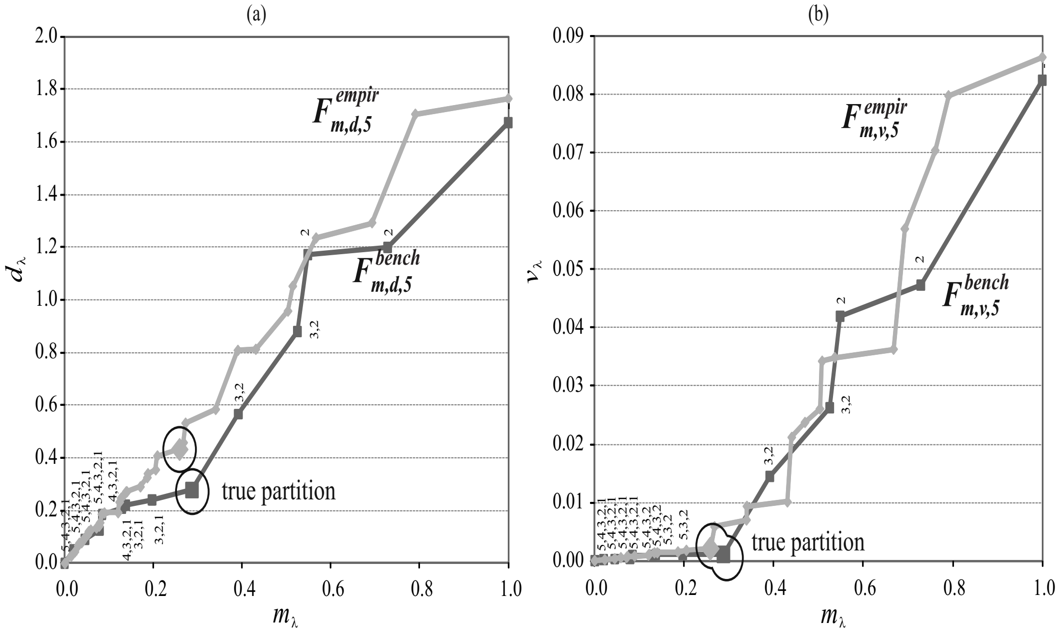

Figure 1: Panel (a) shows the benchmark and empirical efficient frontiers $\mathcal{F}^{bench}_{m,d,5}$ and $\mathcal{F}^{empir}_{m,d,5}$ representing the solutions $\boldsymbol{\lambda}^{A,*} = (\lambda_5^{A,*}, \lambda_4^{A,*}, \lambda_3^{A,*}, \lambda_2^{A,*}, \lambda_1^{A,*})$ of optimization problem (19). Panel (b) shows $\mathcal{F}^{bench}_{m,v,5}$ and $\mathcal{F}^{empir}_{m,v,5}$ representing the solutions of optimization problem (20). Both optimization problems have been solved according to the 3 step procedure presented at the end of Section 5. The procedure has been applied to the 5-path transition probability matrices $\mathcal{A}^{bench}$ and $\mathcal{A}^{empir}$ described in Subsection 6.1. Each point of the benchmark efficient frontiers is labelled with its partition times (see Tables 7 and 8). The circled big squares and diamonds indicate the true partition $\boldsymbol{\lambda}^{A,tr}$.

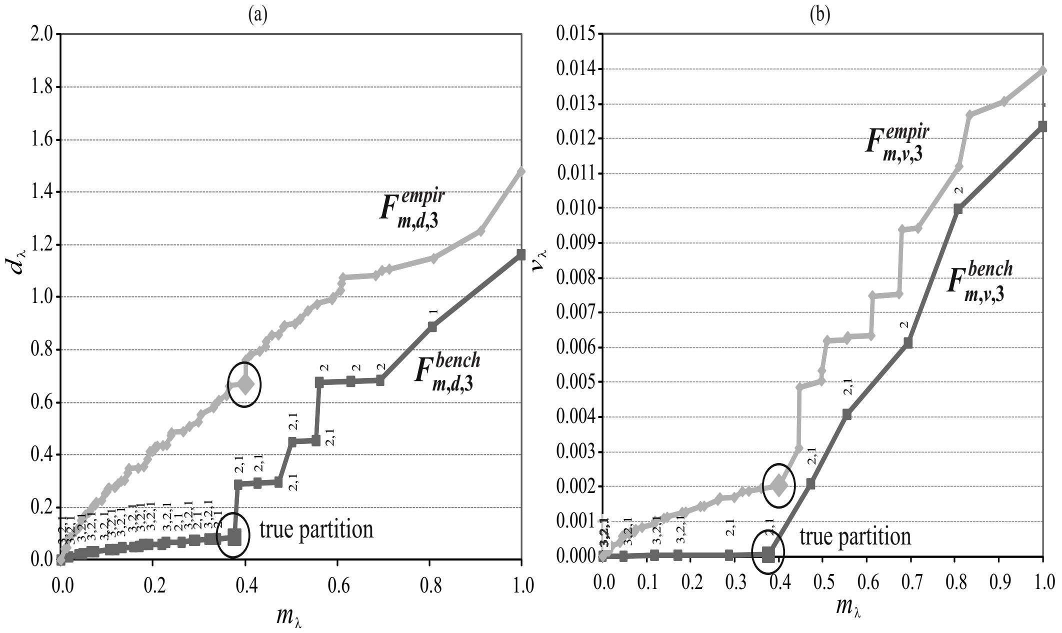

Figure 2: Panel (a) shows the benchmark and empirical efficient frontiers $\mathcal{F}^{bench}_{m,d,3}$ and $\mathcal{F}^{empir}_{m,d,3}$ representing the solutions $\boldsymbol{\lambda}^{B,*} = (\lambda_3^{B,*}, \lambda_2^{B,*}, \lambda_1^{B,*})$ of optimization problem (19). Panel (b) shows $\mathcal{F}^{bench}_{m,v,3}$ and $\mathcal{F}^{empir}_{m,v,3}$ representing the solutions of optimization problem (20). Both optimization problems have been solved according to the 3 step procedure presented at the end of Section 5. The procedure has been applied to the 3-path transition probability matrices $\mathcal{B}^{bench}$ and $\mathcal{B}^{empir}$ described in Subsection 6.1. Each point of the benchmark efficient frontiers is labelled with its partition times (see Tables 9 and 10) The circled big squares and diamonds indicate the true partition $\boldsymbol{\lambda}^{B,tr}$.

References

Akaike H (1970) On a decision procedure for system identification. In: Kyoto Symposium on System Engineering Approach to Computer Control (ed) Proceedings of the IFAC Kyoto Symposium on System Engineering Approach to Computer Control. Kyoto Symposium on System Engineering Approach to Computer Control, Kyoto, Japan, pp 485–490.

Anatolyev S, Vasnev A (2002) Markov chain approximation in bootstrapping autoregressions. Economics Bulletin 3:1–8.

Athreya KB, Fuh CD (1992) Bootstrapping Markov chains: Countable case. Journal of Statistical Planning and Inference 33:311–331.

Barron A, Rissanen J, Yu B (1998) The minimum description length principle in coding and modeling. IEEE Transactions on Information Theory 44:2743–2760.

Basawa IV, Green TA, McCormick WP, Taylor RL (1990) Asymptotic bootstrap validity for finite Markov chains. Communications in Statistics - Theory and Methods 19:1493–1510.

Bertail P, Clémençon S (2006) Regenerative block bootstrap for Markov chains. Bernoulli 12:689–712.

Bertail P, Clémençon S (2007) Second-order properties of regeneration-based bootstrap for atomic Markov chains. Test 16:109–122.

Brock W, Lakonishok J, LeBaron B (1992) Simple technical trading rules and the stochastic properties of stock returns. The Journal of Finance 47:1731–1764.

Bühlmann P (1997) Sieve bootstrap for time series. Bernoulli 3:123–148.

Bühlmann P (2002) Sieve bootstrap with variable-length Markov chains for stationary categorical time series. Journal of the American Statistical Association 97:443–456.

Bühlmann P, Künsch HR (1999) Block length selection in the bootstrap for time series. Computational Statistics & Data Analysis 31:295–310.

Bühlmann P, Wyner AJ (1999) Variable length Markov chains. The Annals of Statistics 27:480–513.

Carlstein E (1986) The use of subseries values for estimating the variance of a general statistic from a stationary sequence. The Annals of Statistics 14:1171–1179.

Cha S-H (2007) Comprehensive survey on distance/similarity measures between probability density functions. International Journal of Mathematical Models and Methods in Applied Sciences 1:300-307.

Chambaz A, Garivier A, Gassiat E (2009) A minimum description length approach to hidden Markov models with Poisson and Gaussian emissions. Application to order identification. Journal of Statistical Planning and Inference 139:962–977.

Ching W-K, Ng MK, Fung ES (2008) Higher-order multivariate Markov chains and their applications. Linear Algebra and Its Applications 428:492–507.

Csiszár I (2002) Large-scale typicality of Markov sample paths and consistency of MDL order estimators. IEEE Transactions on Information Theory 48:1616–1628.

Csiszár I, Shields PC (2000) The consistency of the BIC Markov order estimator. The Annals of Statistics 28:1601–1619.

Datta S, McCormick WP (1992) Bootstrap for a finite state Markov chain based on i.i.d. resampling. In: LePage R, Billard L (eds) Exploring the Limits of Bootstrap. John Wiley & Sons, New York, NY, USA, pp 77–97.

Datta S, McCormick WP (1993) Regeneration-based bootstrap for Markov chains. The Canadian Journal of Statistics 21:181–193.

Efron B (1979) Bootstrap methods: Another look at the jackknife. The Annals of Statistics 7:1–26.

Efron B, Tibshirani RJ (1986) Bootstrap methods for standard errors, confidence intervals, and other measures of statistical accuracy. Statistical Science 1:54–75.

Efron B, Tibshirani RJ (1993) An Introduction to the Bootstrap. Chapman & Hall, New York, NY, USA.

Feder M, Merhav N, Gutman M (1992) Universal prediction of individual sequences. IEEE Transactions on Information Theory 38:1258–1270.

Finesso L (1992) Estimation of the order of a finite Markov chain. In: Kimura H, Kodama S (eds) Recent Advances in Mathematical Theory of Systems, Control, Networks, and Signal Processing: Proceedings of the International Symposium MTNS-91. Mita Press, Tokyo, Japan, pp 643–645.

Freedman DA (1984) On bootstrapping two-stage least-squares estimates in stationary linear models. The Annals of Statistics 12:827–842.

Freedman DA, Peters SC (1984) Bootstrapping a regression equation: Some empirical results. Journal of the American Statistical Association 79:97–106.

Hall P (1985) Resampling a coverage pattern. Stochastic Processes and their Applications 20:231–246.

Hall P, Horowitz JL, Jing B-Y (1995) On blocking rules for the bootstrap with dependent data. Biometrika 82:561–574.

Horowitz JL (2003) Bootstrap methods for Markov processes. Econometrica 71:1049–1082.

Kieffer JC (1993) Strongly consistent code-based identification and order estimation for constrained finite-state model classes. IEEE Transactions on Information Theory 39:893–902.

Kolmogorov AN (1965) Three approaches to the quantitative definition of information. Problemy Peredachi Informatsii 1:3–11.

Kulperger RJ, Prakasa Rao BLS (1989) Bootstrapping a finite state Markov chain. Sankhya, The Indian Journal of Statistics 51:178–191.

Künsch HR (1989) The jackknife and the bootstrap for general stationary observations. The Annals of Statistics 17:1217–1241.

Lahiri SN (2003) Resampling Methods for Dependent Data. Springer-Verlag, New York, NY, USA.

Lahiri SN, Furukawa K, Lee Y-D (2007) A nonparametric plug-in rule for selecting optimal block lengths for block bootstrap methods. Statistical Methodology 4:292–321.

Liu C-C, Narayan P (1994) Order estimation and sequential universal data compression of a hidden Markov source by the method of mixtures. IEEE Transactions on Information Theory 40:1167–1180.

Liu RY, Singh K (1992) Moving blocks jackknife and bootstrap capture weak dependence. In: LePage R, Billard L (eds) Exploring the Limits of Bootstrap. John Wiley & Sons, New York, NY, USA, pp 225–248.

Merhav N, Gutman M, Ziv J (1989) On the estimation of the order of a Markov chain and universal data compression. IEEE Transactions on Information Theory 35:1014–1019.

Morvai G, Weiss B (2005) Order estimation of Markov chains. IEEE Transactions on Information Theory 51:1496–1497.

Paparoditis E, Politis DN (2001a) Tapered block bootstrap. Biometrika 88:1105–1119.

Paparoditis E, Politis DN (2001b) A Markovian local resampling scheme for nonparametric estimators in time series analysis. Econometric Theory 17:540–566.

Paparoditis E, Politis DN (2002a) The tapered block bootstrap for general statistics from stationary sequences. The Econometrics Journal 5:131–148.

Paparoditis E, Politis DN (2002b) The local bootstrap for Markov processes. Journal of Statistical Planning and Inference 108:301–328.

Peres Y, Shields PC (2008) Two new Markov order estimators. `http://arxiv.org/PS_cache/math/pdf/0506/0506080v1.pdf`. Retrieved on 9 February 2013.

Politis DN, Romano JP (1992) A general resampling scheme for triangular arrays of α-mixing random variables with application to the problem of spectral density estimation. The Annals of Statistics 20:1985–2007.

Politis DN, Romano JP (1994) The stationary bootstrap. Journal of the American Statistical Association 89:1303–1313.

Politis DN, White H (2004) Automatic block-length selection for the dependent bootstrap. Econometric Reviews 23:53–70.

Rajarshi MB (1990) Bootstrap in Markov-sequences based on estimates of transition density. Annals of the Institute of Statistical Mathematics 42:253–268.

Rissanen J (1978) Modeling by shortest data description. Automatica 14:465–471.

Rissanen J (1983) A universal data compression system. IEEE Transactions on Information Theory IT-29:656–664.

Rissanen J (1986) Complexity of strings in the class of Markov sources. IEEE Transactions on Information Theory IT-32:526–532.

Rissanen J, Langdon Jr. GG (1981) Universal modeling and coding. IEEE Transactions on Information Theory IT-27:12–23.

Schwarz G (1978) Estimating the dimension of a model. The Annals of Statistics 6:461–464.

Sullivan R, Timmermann A, White H (1999) Data-snooping, technical trading rule performance, and the bootstrap. The Journal of Finance 54:1647–1691.

Ullah A (1996) Entropy, divergence and distance measures with econometric applications. Journal of Statistical Planning and Inference 49:137-162.

Weinberger MJ, Lempel A, Ziv J (1992) A sequential algorithm for the universal coding of finite memory sources. IEEE Transactions on Information Theory 38:1002–1014.

Weinberger MJ, Rissanen JJ, Feder M (1995) A universal finite memory source. IEEE Transactions on Information Theory 41:643–652.

Ziv J, Merhav N (1992) Estimating the number of states of a finite-state source. IEEE Transactions on Information Theory 38:61–65.