POLITECNICO DI TORINO

Physics of complex systems

Master's Degree Thesis

PREDICTIVE RELEVANCE IN DYNAMICAL SYSTEMS

Supervisors

Prof. Luca DALL'ASTA

Prof. Matteo MARSILI

Dr. Iacopo MASTROMATTEO

Prof. Michael BENZAQUEN

Candidate

Davide STRAZIOTA

2020/2021

Summary

Complex systems have been the object of studies across diverse fields, comprising both hard and soft sciences. The data revolution and the enormous amount of data available in recent years allowed for their quantitative analysis. According to the problem under investigation, one can focus on their stationary properties, thus adopting a static description, or analyze their dynamics and their out-of-equilibrium properties. The goal of this thesis is to investigate the dynamical behavior of a complex system using dimensional reduction, a technique aiming to reduce the number of degrees of freedom of the system by constructing a synthetic, more effective representation. Here we focus on a fully unsupervised approach to dimensional reduction by coupling clustering techniques with recent ideas of maximally informative representations. The performance of prediction algorithms built around these ideas has been tested. The relation between prediction power and information content, of the clustering label set, has been examined by considering two numerical experiments. The first experiment is designed for finding the best agglomerative methods' inter-cluster linkage, while the second experiment for investigating whether, starting from a system state representation, it has been possible to detect time series underlying generative model properties. The experimental results have been used for drawing general conclusions about this dimensional reduction technique.

Acknowledgements

This work is dedicated to those who loved me and supported me in all the choices that brought me to reach this milestone. In particular to my father, my first supporter and friend, my mother, who has been always a shoulder to cry on. They are my heroes, because they taught me to always work hard for perciving my desires. My thesis is also dedicated to my sister, my best friend and confessor. She helped me in all the hardest situation of my life. To Ilaria, my girlfriend and, first of all, my partner in crime. She always supported me in all my choices, even the not shared ones. She always believed in me. This is also dedicated to my grandparents (Nonna Rosa, Nonna Nina, nonno Gigi e Nonno Vito) and my aunts (zia Loredana e zia Alessia), the fans of all my adventures, even the one bringing me far from them. This work is dedicated to all of you. I know it has not been easy for you to have a son, brother, boyfriend or grandson like me, who never replied to phone call. I would also say thanks to the Econophysix group and the CFM, and in particular to Matteo Marsili, Iacopo Mastromatteo, Michael Benzaquen, Jean-Philippe Bouchaud and Samy Lakhal, who helped me realization of this thesis work. They have been always available for suggestions and further explanations, at each time of the day. It has been a wonderful experience to work with you. A special thanks also to Luca Dall'Asta, who helped my in all the possible ways. I would finally say thanks to all my friends, and in particular to Rocco, Giuseppe, Max, Alessandro, Caterina, Pietro, Alice, Saverio, Mattia, Matteo e Alice, who always believed in me and aid me in the most difficult moment of the last years, but, most important, who always tried to jumped through hoops to meet me, despite the distance that separated us, the costs of the flight and my disorganization. I know that it has been hard to be a friend of mine. If I reached the finish line of the academical path, it is also thanks to all them. Thank you for everything.

Table of Contents

List of Tables VII

List of Figures VIII

1 Introduction 1

2 Dynamical complex systems 4
- 2.1 Definitions and ideas . . . 4
- 2.2 Dimensional reduction . . . 6
- 2.3 Prediction . . . 7
- 2.4 Statistical validation . . . 9
  - 2.4.1 $\mathcal{S}$-space predictions . . . 9
  - 2.4.2 $\mathcal{Y}$-space predictions . . . 10

3 Maximally informative representations 11
- 3.1 Relevance vs predictive relevance . . . 11
- 3.2 Relevance, resolution and maximally informative representations . . . 12

4 Datasets and methods 16
- 4.1 Introduction to datasets . . . 16
  - 4.1.1 ARCH model . . . 16
  - 4.1.2 Real dataset . . . 19
- 4.2 Reshaping and clustering procedures . . . 20
- 4.3 Prediction algorithms . . . 21
  - 4.3.1 Unsupervised predictors . . . 22
  - 4.3.2 Supervised predictors . . . 26
- 4.4 Relevance, resolution and total information . . . 31

5 Application I: comparison of clustering methods 32
- 5.1 Clustering method optimization . . . 32
- 5.2 ARCH model: Results . . . 33

5.3 S&P 500: results 38
5.4 Information bottleneck method 40

6 Application II: detecting degrees of freedom of a system 43
6.1 Relevance and model structure 43
6.2 ARCH model: results 44

7 Conclusions 47

A Causal states 50

B Information content and prediction for clustering algorithms 53

Bibliography 55

List of Tables

4.1 Values of the parameters chosen for the ARCH model dataset . . . 17

List of Figures

4.1 Squared returns vs time, for an ARCH model time series, $T = 1000$ and $q = 4$ . . . 18
4.2 Squared volatilities vs time, for an ARCH model time series, $T = 1000$ and $q = 4$ . . . 18
4.3 Squared returns vs time, for S&P 500 stock, 5 years daily returns . 19

5.1 Prediction error vs Resolution, ARCH dataset (hidden variable space, unsupervised prediction) . . . 34
5.2 Prediction error vs Resolution, ARCH dataset (real variable space, unsupervised prediction) . . . 35
5.3 Prediction error vs Resolution, ARCH dataset (Supervised predictor) 36
5.4 Total information vs Resolution, ARCH dataset . . . 37
5.5 Relevance vs Resolution, ARCH dataset . . . 37
5.6 Prediction error vs Resolution, S&P 500 dataset (hidden variable space, unsupervised prediction) . . . 38
5.7 Total information vs Resolution, S&P 500 dataset . . . 39
5.8 Relevance vs Resolution, S&P 500 dataset . . . 39
5.9 Accuracy vs k, ARCH dataset . . . 41
5.10 Relevance vs k, ARCH dataset . . . 42

6.1 Relevance vs k, $T = 100$ . . . 44
6.2 Relevance vs k, $T = 500$ . . . 44
6.3 Relevance vs k, $T = 1000$ . . . 44
6.4 Prediction error vs k, $T = 100$ . . . 45
6.5 Prediction error vs k, $T = 500$ . . . 45
6.6 Prediction error vs k, $T = 1000$ . . . 45

A.1 Effective states graphical representation . . . 51
A.2 Causal states graphical representation . . . 52

B.1 Relevance vs k, Average Linkage, ARCH model . . . 53
B.2 Relevance vs k, Complete Linkage,ARCH model . . . 53

B.3 Relevance vs k, Single Linkage, ARCH model 54
B.4 Relevance vs k, Ward Linkage,ARCH model 54
B.5 Relevance vs k, Average Linkage, S&P 500 54
B.6 Relevance vs k, Complete Linkage, S&P 500 54
B.7 Relevance vs k, Single Linkage, S&P 500 54
B.8 Relevance vs k, Ward Linkage, S&P 500 54

Chapter 1

Introduction

Complex systems have been the object of studies across different fields, comprising both hard and soft sciences. The data revolution and the enormous amount of data available in recent years allowed for their quantitative analysis. Complex systems are characterized by several variables performing specific functions, interacting with each other. A simple example of a complex system with many variables interaction is DNA melting, described by the Poland-Scheraga model. Usually, the effect of these interactions is not trivial and can be unexpected, i.e. the phase transition characterizing DNA melting. According to the problem under investigation, one can focus on their stationary properties, thus adopting a static description, or analyze their dynamics and their out-of-equilibrium properties. In this paper, we devote our attention to the second type of complex system descriptions, and, in particular, to their realization: time series, sequences of observables labeled by time.

Nowadays, the interest in predicting time series future behavior has increased exponentially, due to its infinite number applications, starting from financial investments, to neuronal potential analysis. This prediction has been approached using several methods and techniques, of which one of the most important is dimensional reduction. In dimensional reduction, given the sequence of time labeled observations, we build a new representation of the system state, by dividing data into groups, to ease future prediction. In other words, it is a technique aiming to reduce the number of degrees of freedom of the system by constructing a synthetic, more effective representation. Up to now, there is no universally acknowledged approach for realizing dimensional reduction, and several methods have been developed to reach this goal. In [1], Crutchfield and Shalizi define causal states as the set of all the past observation sequences such that the value of the conditional probability between them and the future is constant, proposing them as the optimal representation for system state. Several automatic techniques for reconstructing causal states have been implemented. The most famous causal states

reconstruction algorithms are the REMAPF algorithm, realized by N. Brodu in [2], and CSSR algorithm, devised by Shalizi et al. in [3]. A different method for approaching dimensional reduction is the information bottleneck method, devised by Bialek et al. in [4]. It consists of an optimization problem of a Lagrangian function containing the linear combination of two terms: the mutual information between the past of the time series and the chosen representation for the system state, and the mutual information between the new representation and the relevant variables for the underlying generative model for the system.

There is a common aspect in these two dimensional reduction approaches: the optimal representation can be found, if and only if, we know the future of the time series or its generative model. It is important to underline that, future and generative knowledge are strictly related. These approaches, however, are suitable for investigating only well-known complex systems but could be problematic in real situations. We aim to find a dimensional reduction technique that can make predictions independently from future behavior and model knowledge. In this framework, how can we identify relevant representations? Is there a way to define a relevant representation for system state, from a predictive inference point of view? Is it possible to do it without knowing the future of the time series, or its generative model? In this thesis, we aim to answer these questions, by using an investigation approach inspired by the theory of maximally informative representation (MIR), developed in [5] by Marsili et al. Maximally informative representation have been developed for addressing the static investigation of complex systems, like photographic images. However, the independence from the problem's nature and the nonspecific framework in which the theory has been developed prompted us to suppose that this approach can be generalized to the dynamical analysis of complex systems. In particular, it exists a strict relation between the MIR-inspired approach and the information bottleneck method, which is deeply investigated in [6] by Song et al. In this thesis, we try also to investigate this relation from a dynamical point of view.

In Chapter 2, we provide important definitions for describing dynamical systems. Starting from these definitions, we propose a dimensional reduction technique, and we explore the predictability of time series future behavior, starting from the new representation, and its statistical validation. In Chapter 3, we introduce the concepts of relevance, resolution, and maximally informative representations, trying to relate the predictive power of a system state representation and the information it has been able to extract from a dataset. Here, we also develop a proper formalism to approach time series analysis. In Chapter 4, we describe the different datasets and the techniques used for the investigation of dimensional reduction. In Chapter 5 and Chapter 6, two numerical experiments are described. The aim of these experiments is parallel: the first try to detect whether the approach based on the comparison of information content and predictive power of a representation

can classify clustering algorithm, while the second experiment tries to detect the dimensionality and degrees of freedom for the underlying generative model, using the new representation for the system state. Finally, in Chapter 7, we discuss the numerical results and we propose some ideas for further investigations. In Appendix A, a brief introduction to the concept of causal states is realized and in Appendix B the behavior of MIR quantities is shown for all the inter-cluster distances used in the thesis.

Chapter 2

Dynamical complex systems

In Chapter 1, we introduced the concept of dimensional reduction, which is the subject of our investigation. In particular, as previously mentioned, the aim of this thesis work is to devise an approach for detecting relevant representation for complex systems states, starting from its specific realizations.

In Section 2.1, some general concepts regarding time series are introduced, i.e. the definition of past, future, and observer knowledge state. In section 2.2, we introduce the mathematical definition of time series dimensional reduction. Finally, in section 2.3 we introduce some theoretical background for time series future prediction and, in section 2.4, we make some consideration about their statistical validation.

2.1 Definitions and ideas

As previously mentioned, time series are special realizations of complex systems, represented by a sequence of observables labeled by time. To deal with them, we need to describe time series general properties and define a proper formalism. These properties should be independent of the specific nature of the system.

Let's consider a non specific sequence of observables, sampled from an undefined stochastic process, $\mathbf{y} = \{\dots, y_0, \dots, y_t, \dots\}$, with y_0 the first detected observable, i.e. the origin of the reference system, and y_t the value of the observable detected at a fixed t, where the index t is a time label assuming integer values in the interval $\{0, \dots, \infty\}$. In a real situation, it is not possible to observe a phenomenon for an infinite amount of time, thus defining T the total observation time length, it is possible to state that the index t belongs to $\{0, \dots, T\}$. Each observation belongs to an observable space $\mathcal{Y}$

$$y_t \in \mathcal{Y} \qquad \forall t \in \{0, \dots, T\}$$

where $\mathcal{Y}$ can be either continuous or discrete, depending on the system of study. In the case of financial returns, the one in which we are interested in, the space $\mathcal{Y}$ is continuous.

Once the basic mathematical definition of time series has been provided, we can introduce two fundamental concepts: past and future. For the sake of completeness, assuming to be at time t and having observed y_t, let's define

$$\overrightarrow{Y}_t = \{y_{t+1}, y_{t+2}, \dots\} \qquad \overleftarrow{Y}_t = \{\dots, y_{t-2}, y_{t-1}\}$$

where $\overrightarrow{Y}_t$ is the future and $\overleftarrow{Y}_t$ the past. By knowing these quantities it is possible to define the concept of observer knowledge. At time t, it is encoded in $x_t \in \mathcal{X}$. Observer knowledge is a filtration of the past, thus for example we can assume $x_t = \overleftarrow{Y}_t$. However, this is not a realistic situation, since no observer can have full knowledge of time series realization. A good alternative is to assume that it exists a maximal memory length, q', up to which the observer can look at. In this case, the observer knowledge is represented by

$$x_t = \{y_t, y_{t-1}, \dots, y_{t-q'+1}\}$$

It is straightforward noticing that T and q' are strictly related. But it is fundamental understanding that, knowing the sequence $\{y_t, y_{t-1}, \dots, y_{t-q'+1}\}$, also every other sequence of function of y_t is known, and thus the filtration can be written in terms of functions of observables. In particular, for our investigation, the observer knowledge is the set of squared observable values.

Before going further in our investigation, let's take some considerations. Usually, in dynamical complex systems, the underlying stochastic process for the sequence $\mathbf{y}$ has a structure. In other words, the sequence of observables is not uniformly sampled from observable space, but it is sampled in such a way to minimize a cost function, i.e. maximize a Lagrangian function ruling the system behavior. Let's consider for example a mechanical system with a particle in quiet, subject a potential. The sequence of positions occupied by the particle represents the time series, that we want to consider. In this framework, the observations are not random, but they are determined through the minimization of the potential, which corresponds to the Lagrangian function, change of sign. However, in general, it is not true that the specific sequence $\mathbf{y}$ is directly related to this optimization problem, and, as observers, we aim to predict, given the history of the system, the future observable values. In other words, it is possible that the minimization of the cost function is not realized through the sequence $\mathbf{y}$, but maybe through a sequence of hidden variables, someway related to them. We hope that it exists a dimensional reduction technique able to extract this particular set of variables. We deepen this idea in the next sections.

This framework is frequently observed in various fields of research. A simple example is represented by the neuronal system in animals' brains, where $\mathbf{y}$ is the sequence of neural effective potential spikes. Another simple example is represented by the sequence of financial stock returns at time t, indicated by y_t. The development of a general approach for devising dimensional reduction could help to progress in many field of research.

2.2 Dimensional reduction

As mentioned in section 2.1, the specific value $y_t \in \mathcal{Y}$, and thus x_t, could be not directly connected to the optimization problem of the cost function. If we directly try to predict the future behavior of the system using them, a problem can arise: the prediction could be wrong. For overcoming this problem we should find a relevant representation for the system, i.e. we should find a good set of (hidden) states, whose sequence is the realization of the optimization problem previously mentioned. There have been several attempts to find such a set of states. As previously mentioned, a first attempt to detect a relevant representation for the system state has been described by J. P. Crutchfield in [1], with the concept of causal states and $\epsilon-$machines. For a brief introduction to the concepts of causal states and $\epsilon-$machines consult Appendix A. However, to find a good representation for the state of the system without knowing the underlying stochastic process of the time series, we decided to consider the problem from a more machine learning-related perspective, using clustering algorithms. We suppose that it should exist a clustering algorithm, whose clustering labels represent a good complex system state representation, from a dynamical perspective.

To investigate these representations, let's suppose we are given a time series $\mathbf{y}$, whose related observer knowledge at time t is x_t. After clustering x_t with the clustering algorithm m, we obtain

$$s_t = \mathcal{C}_\alpha^m(x_t) \qquad \forall t \in \{0, \dots, T\} \quad s.t. \quad y_t \in \mathcal{Y} \tag{2.1}$$

the set of clustering labels associated with x_t. In this definition, α is the parameter controlling the number of states, and thus related to the number of clusters, in which we decided to divide our data (resolution). Here, with resolution, we refer to a function measuring the chosen coarse-graining level. A remark is required: resolution is not related to the relevance of the representation of the state. The relation between relevance and resolution and their mathematical definitions is described in Chapter 3. Before going further, for the aim of clarity, two extreme cases need to be specified:

- for $\alpha = 0$ we map all the x_t in a single state (low resolution);

- for $\alpha = \infty$ we map all the x_t in the highest number states (high resolution).

thus high values of α are related to a high number of clusters, while low values are related to few clusters. The procedure we just defined is a real dimensional reduction technique. It is clear, indeed, that for finding the relevant representation, we need to investigate the hidden states provided by the clustering labels. However, we still have not defined the way to investigate how relevant these representations are, and how the concept of relevance is related to their predictive power and information content. In the next section, we explore the relationship between relevance and predictive power.

2.3 Prediction

As introduced in Section 2.2, we aim to find connections between relevant representations of the system state, and the possibility predicts its future. It is important to underline how with the word state we do not mean strictly the next observable value but in general the next relevant label provided by the clustering algorithm. As previously introduced, there exists a strict relation between relevant representation and observable value, and they coincide in the case of $\alpha = \infty$.

Once the clustering labels s_t associated to the knowledge x_t have been computed, we can try to predict s_{t+1} value, from previous labels, and then directly estimate the value of the next observation y_{t+1}, from it, using the properties of the clustering algorithm and from the definition of cluster centroid. Graphically, the prediction procedure can be represented by the following sequence

$$x_t \xrightarrow[\mathcal{C}_\alpha^m]{} s_t \to s_{t+1} \to y_{t+1} \tag{2.2}$$

For formalizing this procedure from a mathematical point of view, we can state that:

$$P(y_{t+1}|x_t) \propto \sum_{s_t, s_{t+1}} P(y_{t+1}|s_{t+1}) P(s_{t+1}|s_t) P(s_t|x_t) \tag{2.3}$$

Since the hidden state at time t is obtained through clustering algorithms, it is straightforward noticing that

$$P(s_t|x_t) = \delta_{s_t - \mathcal{C}_\alpha^m(x_t)}$$

This is not the general expression of the $P(y_{t+1}|x_t)$, but it is supposed to be the right one for relevant representations, for which we can assume $P(s_{t+1}|s_t)$ to satisfy Markov property, and thus to assume the exactness of the factorization proposed in Eq.(2.3). For having a deeper comprehension of Eq.(2.3), we should make some

remarks on the dependence of predictions from the coarse-graining level parameter α:

- For $\alpha = \infty$, no dimensional reduction has been implemented, and obviously, no information is lost, however, we have weak predictive power. This is related to the fact that having all clusters with just one point (i.e the number of clusters is equal to the number of points), we have not enough points, in a cluster, for realizing valid predictions and for extracting sufficient information[1].

- For $\alpha = 0$, instead, all the data belong to one cluster, we have an excessive dimensional reduction and thus all the "information" is lost. In this situation the prediction in the hidden space is perfect, but at the same time meaningless, the prediction for future labels is always the same and the one for future observations is a constant value: the cluster centroid.

- The only interesting case is $0 < \alpha < \infty$. Reducing the value of the coarse-graining level, on the one end we lose information, but on the other one, we gain predictive power, since the degrees of freedom has been reduced (from x_t to s_t). The prediction, since $\alpha \neq 0$ is meaningful. In this case, we have a trade-off between predictive power and information loss, which can be translated in the choice of a good set of hidden states, i.e a good set of labels provided by $\mathcal{C}_\alpha^m$.

In general, for having the best prediction of the future, we need to compute (and then optimize) $P(y_{t+1}|x_t)$. Before going further, it could be interesting to analyze in detail the various terms in Eq.(2.3). Let's start considering the third term: it is dependent on the chosen clustering algorithm and coarse-graining level. Its optimization is exactly the main subject of this work. The second term of Eq.(2.3) can be estimated and optimized quite easily under the assumption that s_t follows a Markov dynamics, it should not be too restrictive, because it is always possible to enrich it with extra variables, and, as previously mentioned, this assumption is supposed to be exact in the case of relevant representations. Assuming Markovianity of this term, we can estimate empirically $P(s_{t+1}|s_t)$ via a transition matrix, and, as previously mentioned, using clustering algorithms properties and the concepts of the centroid and variance of a cluster, we can estimate empirically the first term too.

[1]Up to now, we have not talked about information, but this reference is going to be clear once the analysis of Chapter 3 is realized

2.4 Statistical validation

Once the prediction has been realized, we aim to investigate the quality of the prediction. Concerning this analysis, we should make a distinction between two prediction possibilities: the ones working in the space of the hidden variables, $\mathcal{S}$, which try to predict future clustering labels, and the ones working in the space of real observables $\mathcal{Y}$, which try to predict the future value of considered observables. However, Eq.(2.3) shows how the two different predictions are strictly related. Let's consider them separately.

2.4.1 $\mathcal{S}$-space predictions

From a mathematical point of view, predicting in $\mathcal{S}$-space consists in estimating the second term in Eq.(2.3), which is the conditional probability of the label s_{t+1}, given the knowledge s_t. It is related to amount of times in which a jump between label $\tilde{s} = s_{t+1}$ and label $s = s_t$ has been observed. The information on this quantities is contained in the hidden state transition matrix, $\tilde{M}_{s,\tilde{s}}$. The full explanation of how this prediction has been implemented using $\tilde{M}_{s,\tilde{s}}$, and it is described in Chapter 4.

Predictions in $\mathcal{S}$-space have pros and cons. Their main problem is related to the intrinsic nature of the prediction algorithm: working in the space of auxiliary variable, and not in the observable space, the prediction could be rigged. In other words, prediction in the hidden space might perform well, but predictions in real variables space are wrong, i.e. the "mutual information" between variables in real and hidden variables spaces is null. Let's consider, for example, the unphysical situation, in which we have one cluster with all data points. In this case, the prediction of the hidden state is exact, but the real state prediction is wrong. This implies that good predictions in the hidden space are not related, for sure, to good real state predictions. However, this is false if the dimensional reduction provides a relevant representation of the state of the system. In this case, we should expect that the representation should have good enough predictive power both in hidden space and observable one.

There are several ways to implement a prediction in $\mathcal{S}$-space, but a lot of them are problematic. Naively, one hidden space prediction can be devised by counting the fraction of future states correctly predicted, using a transition matrix. However, there could be a problem with it: if a state has been observed rarely, i.e. few jumps out of it have been detected, by using empirical transfer matrix, as probability estimation, we have a sharp probability distribution, without a robust statistic. This can distort our prediction. Another subtle prediction problem is that to be robust in our predictions, we cannot just have a deterministic prediction of the future state of the system, but we should try to get a probability distribution for all the possible incoming states. A solution to all these problems is described in

Chapter 4 using the famous Bayes theorem.

2.4.2 $\mathcal{Y}$-space predictions

The prediction of the future hidden states enables us to predict the values of real observables, which corresponds in estimating, given the dataset, $P(y_{t+1}|s_{t+1})$. To make real space prediction, it is possible to use the concept of the centroid of a cluster and the standard deviation of cluster points. These kinds of predictions are really important because classical applications require to control the statistical properties of y_t or of its functions, and not of s_t. This is the strength of $\mathcal{Y}$-space predictions: they enable us to predict the value of different functions and moments of the observables.

However, also for real space prediction, a problem arises. Since not all the representations of the state of the system are equal, the particular parametrization of the state of the system in the real space would affect the prediction itself. So is there a good observable function to be taken into account for investigating the performance of future prediction? We expect it to be dependent on the complex system itself.

This question is again strictly related to the concept of causal states introduced by Crutchfield in [1] and to the information bottleneck method, described by Bialek in [4]. Starting from this consideration, various prediction algorithms could be implemented. In Chapter 4, these problems are faced from a practical point of view, and both real space and hidden space predictors are described.

Chapter 3

Maximally informative representations

In Chapter 2, we introduced general ideas about dimensional reduction of time series, and about the possibility to predict system future behavior, starting from state representation. In this chapter, we introduce the concepts of relevance, resolution, and maximally informative representations. Using these concepts we deepen the analysis of relevant representation, by investigating the relationship between them and the underlying generative model information extracted from clustering algorithms. This investigation could help us in relating predictive power and stochastic model information content for system state representations.

3.1 Relevance vs predictive relevance

In Chapter 1, introducing the concept of dimensional reduction, we presented the Causal states approach proposed by Crutchfield [1] and the Information bottleneck method defined by Bialek [4]. They require the knowledge of time series future and underlying generative model. However, as previously mentioned, we are interested in finding a dimensional reduction method, independent of the future and generative model knowledge. In this thesis, as previously stated, we propose a different dimensional reduction technique consisting of optimizing C_α^m, using clustering algorithms. However, up to now, apart from prediction power, no general criterion has been suggested for realizing such an optimization both in a static way, so without looking at the time evolution of the system, and in a non-supervised way, without teaching the system what to look at. As mentioned in previous sections, we think that help to solve this problem can be provided by the concepts of relevance and resolution, variables quantifying the information content of the system state representation. We suppose that relevant representation of the state of the system

provides good predictions of future observations. This supposition is driven by the fact that, for sure, bad representations, i.e. the ones extracting only noise from a time series, could not provide a proper future prediction.

In the next section, by defining relevance, resolution, and maximally informative representation, we investigate the information content of label sequences. This analysis is driven by one question: What can we say about the amount of generating process information encoded in relevant representations? Naively, we should expect that the most relevant representation should also be maximally informative about the underlying stochastic process, from which the time series have been sampled. This idea leads us to presume that representations with higher information content, should also have higher prediction powers. From this idea, we believe that a relevant representation should be both maximally predictive and maximally informative.

3.2 Relevance, resolution and maximally informative representations

In the previous section, we have mentioned the concept of maximally informative representations, without explicitly defining it. Here, we define them properly, showing their characteristic behaviors. In [7], Marsili et al., starting from a very general framework, introduce this concept concerning the information content of a sample, and its entropy. In [5], they also introduce the fundamental concepts for investigating the information content of different samples: relevance, resolution, and total information. Now, we generalize these ideas from samples to representations. For this reason, let's first recap how these quantities are defined.

Let's suppose to have a sequence of observables, whose associated observer knowledge is

$$\{x_1, \dots, x_N\}$$

Following the procedure described in Chapter 2, we can cluster the observer knowledge with a fixed coarse-graining level α and method m, obtaining a sequence of hidden states, i.e. clustering labels

$$\{s_1, \dots, s_N\}$$

By indicating with K_s the number of times the label s was observed in the sample, we get

$$K_s = \sum_{i=1}^{N} \delta_{s_i,s}$$

Similarly, by indicating with m_k the number of clustering labels observed exactly k times, it is possible to obtain

$$m_k = \sum_s \delta_{k,K_s}$$

Having obtained the specific expressions for K_s and for m_k, the related entropies can be computed, which are named respectively as resolution and relevance:

$$\hat{H}[s] = -\sum_s \frac{K_s}{M} \log \frac{K_s}{M} \tag{3.1}$$

$$\hat{H}[k] = -\sum_k \frac{km_k}{M} \log \frac{km_k}{M} \tag{3.2}$$

From these mathematical definitions, it is possible to understand that resolution represents the number of bits needed to locate an observation fixed the number of states, and so it is strictly related to the chosen coarse-graining level α. On the other hand, being the relevance proportional to the minimal number of necessary bits per state to optimally encode the output of the experiment, it quantifies the number of states that the sample allows to distinguish and so it provides insights about the generating model of the time series. A different way to look at the relevance is to notice that it quantifies the number of bits that are available to probe the system Lagrangian structure. Moreover, since we are dealing with a complex system from a dynamic point of view, to fully know the true Lagrange function we should have complete knowledge of the system. This function is high dimensional, concerning our data, i.e. there are more parameters with respect to the number of observations, thus it is not possible to fully know it. Conscious of it, what we can try to do, is just to obtain the best representation of the system state, to be able to predict most accurately the future behavior of the system.

It is clear, considering relevance definition, that relevant representations of the state of the system are also informative about the generating process. At this point, it is important to notice that we have not considered in the analysis any specific feature of the process, but just general ones. For this reason, in case of favorable results, this procedure could be used for analyzing the most various dynamical systems and time series, starting from financial ones to neural ones. Since we are dealing with stochastic processes, it could be convenient to quantify the noise amount in the sample as

$$\hat{H}_{noise}[s|k] = \hat{H}[s] - \hat{H}[k]$$

The last important definition required for dealing with maximally informative

representations is the one of total information contained in a representation,

$$\hat{H}_{tot}[s,k] = \hat{H}[s] + \hat{H}[k] \tag{3.3}$$

which, in practice, represent the sum of relevance and resolution.

These concepts lead us to mathematically state that maximally informative representations are representations in which the relevance is maximal at a fixed resolution value. In other words, for finding the maximally informative representation we need to maximize the following functional

$$\mathcal{F} = \hat{H}[k] + \mu\left(\hat{H}[s] - H_0\right) + \lambda\left(\sum_k k m_k - N\right) \tag{3.4}$$

with, μ and λ two Lagrange multipliers to be adjusted and H_0 the fixed resolution level. As shown in [8] and in [5], the optimization of the functional described in Eq.(3.4), can be realized analytically, and the result is that maximally informative representations exhibits a power law frequency distribution described by

$$m_k \sim c k^{-1-\mu} \tag{3.5}$$

with c a normalization constant and μ quantifying the trade-off between resolution and relevance. The point in which $\mu = 1$ sets the limit beyond which further reduction in $\hat{H}[s]$ results in lossy compression, in fact, for $\mu < 1$ the increase in the resolution cannot compensate the loss in relevance. In this limit, $\mu = 1$ we recover the well known Zipf's law

$$m_k \sim c k^{-2} \tag{3.6}$$

From this result, Marsili et al. have been able to extract the following general result: mostly informative representations are those for which the frequency of observations covers the largest possible dynamic range, providing information on the system's optimal behavior in the wider range of possible circumstances.

Strong of these results, we are interested in understanding how they relate to the possibility of predicting the future of a particular realization of a dynamical system, starting from a particular system state representation. In other words, if we have a system with a sufficiently complex structure is there a way to use relevance and resolution to drive the choice of $\mathcal{C}_\alpha^m(x_t)$, to extract the highest amount of information about the generating stochastic model, and the lowest amount of noise? The definition of relevance and resolution leads us to suppose that, after fixing the resolution value, representation with higher relevance, should contain more information about the generating process, and thus should provide better future predictions. In other words, by having relevant and maximally informative representations of the system state, it could be possible to have access to a highly

predictive form of the Lagrange function, affecting the system behavior, even if the observation length is finite, as mentioned in Chapter 2. To investigate this idea we have devised two numerical experiments, which are described in Chapter 5 and Chapter 6.

Chapter 4

Datasets and methods

In this chapter, we present the datasets used for relevant representation investigation, in Section 4.1. Once they have been defined and their properties described, a fundamental reshaping procedure of data is shown, in Section 4.2. This reshaping procedure is fundamental for the definition of the predictions algorithms and of the algorithms devised for computing relevance, resolution, and total information, in the last two sections. All these elements have primary importance in the numerical investigation, which take place in Chapter 5 and Chapter 6.

4.1 Introduction to datasets

For relevant representations investigation, as just stated, we decided to use two different datasets, the first of them is synthetic, a time series sampled from an ARCH model, while the second is a real financial time series. In the following sections, these time series are shown and their most important properties described.

4.1.1 ARCH model

For the first part of relevant representation analysis, to be able to check that qualitative and quantitative expected behaviors are observed, we decided to consider a synthetic time series, sampled from a known stochastic process. We have devised an ARCH model time series of total duration $T = 1000$ step, so to be sure that T is much longer than the auto-correlation time. This assumption is required for having an ergodic process. The ergodicity requirement is fundamental for having a set of independent and identically distributed x_t.

ARCH model is an auto-regressive statistical model, whose volatility, σ_t, at a fixed time t, is function of past returns up to a memory length q, [9]. For the aim of consistency, we denote the time series of returns by $\mathbf{y}$. Once the volatility is

known, it is possible to sample the y_t variable value. Its analytical behaviour is described by the following equations:

$$y_t = \sigma_t \cdot z_t \qquad \sigma_t^2 = a + b_1 \cdot y_{t-1}^2 + b_2 \cdot y_{t-2}^2 + \cdots + b_q \cdot y_{t-q}^2 \tag{4.1}$$

where $z_t = \mathcal{N}(0,1)$, and consequently $y_t \sim \mathcal{N}(0, \sigma_t^2)$. The memory length of the process q is defined as the maximal time distance of past squared return affecting the value of the squared volatility. In this framework, the observer knowledge depends on how long he is able to look in the past. So by defining the observer memory length as q', which in principle could be different from the real memory length of the process, it is possible to define the observer knowledge as

$$x_t = (y_t^2, \ldots, y_{t-q'+1}^2)$$

where y_t^2 represents exactly the squared return value at time t. It is important to underline, that x_t is a collection of square returns. The choice of q' is strictly related to the concept of causal states, described in Appendix A. For our investigations, we decided to sample a time series characterized by a memory length $q = 4$. The parameters chosen for the simulations are reported in the following table.

parameters	value
a	4
b_1	0.25
b_2	0.25
b_2	0.25
b_2	0.25

Table 4.1: Values of the parameters chosen for the ARCH model dataset

As already introduced in section 3.1, we do not expect the result to be dependent on the parameters choice, reported in Tab.4.1. Given the stochastic nature of the process, it is neither useful to consider identical values for the b's parameters. The dimensional reduction applied to the dataset is unconscious of the model from which the data have been sampled. Furthermore, we should remember that the definition of relevance and resolution is independent of the model structure.

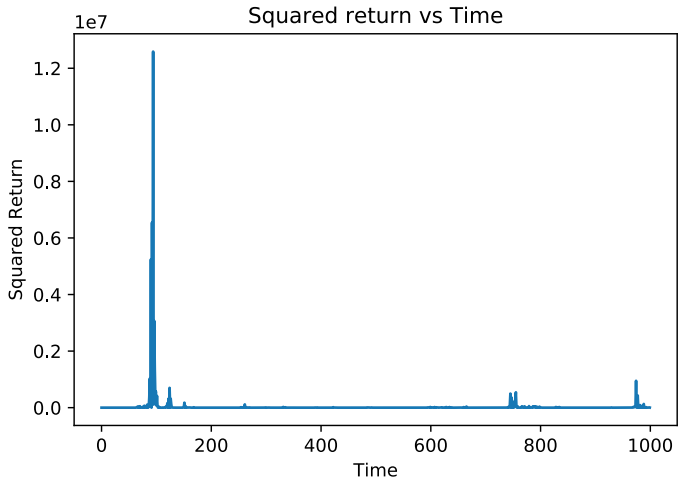

Figure 4.1: Squared returns vs time, for an ARCH model time series, $T = 1000$ and $q = 4$

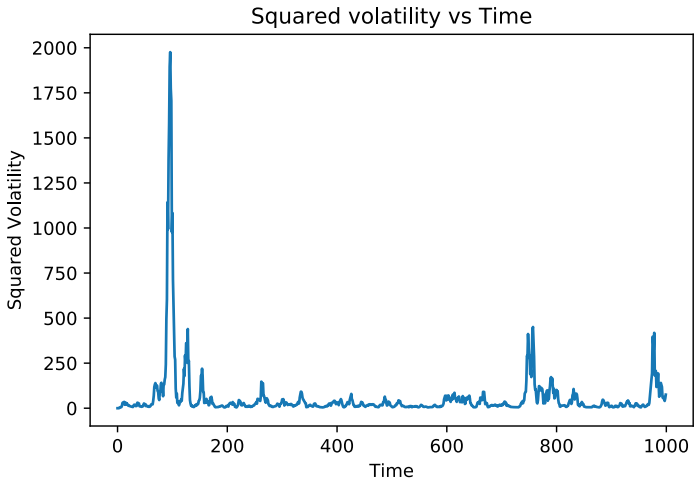

Figure 4.2: Squared volatilities vs time, for an ARCH model time series, $T = 1000$ and $q = 4$

Before going further in the investigation of relevance representation, some remarks about the independence of subsequent observer knowledge need to be presented. To be able to use MIR theory, we require the sample, of past filtrations, to be made of i.i.d. elements. For having such a sequence of $\mathbf{x}$, we should proceed in the following way: at first sample an ARCH model time series, wait for the

convergence to the stationary state, to guarantee ergodicity, and then store just one x_t value, repeating this procedure for T times. This procedure is computationally expensive and so we tried to understand whether results change using a single time series, for which we waited for stationary state convergence. The results of this comparison are not explicitly shown in this thesis, but the two procedures lead to the same behavior. The similarity of the behaviors does not surprise us, and it can be explained by looking at the autocorrelation time of the process, τ. This quantity represents, as previously stated, the time after which two observations can be considered uncorrelated. In other words, considering a single time series, is identical to consider a sample of i.i.d. $\mathbf{x}$ of a total length of the order $\frac{T}{\tau}$. This is due to the ergodicity property owned by the system.

4.1.2 Real dataset

For dealing with a more realistic dataset, we decided to investigate maximally informative representation for non-synthetic data, being conscious that, in this case, we do not know the exact underlying process from which data have been sampled. For the sake of continuity with the ARCH model, we considered the time series defined by the returns of the S&P 500 stock[1]. The total observation time has been fixed to 5 years, to be comparable with the observation time of the ARCH time series. The observation has been started on June 6^{th} 2016 and ended on June 6^{th} 2021.

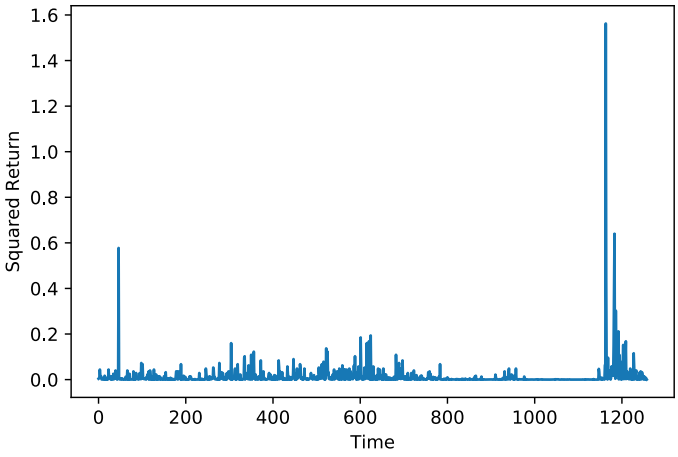

Figure 4.3: Squared returns vs time, for S&P 500 stock, 5 years daily returns

[1]The time series of S&P 500 has been collected using yahoo finance

By looking at Fig.4.1 and at Fig.4.3, it is possible to notice a completely different behaviour of squared returns, implying a different underlying generative process. From this, it is possible to conclude that, in general, it is wrong to assume the S&P 500 time series to be a realization of an ARCH model.

Once again, some remarks on the independence of $\mathbf{x}$ in the time series are required. Here, since we do not know exactly the underlying generative process of the time series, we are not anymore sure it to be stationary and the sequence of $\mathbf{x}$ to be i.i.d. Several financial effects can affect the stationarity of the process, i.e. cyclic phenomena or seasonalities. However, to keep coherence with MIR theory, we decided to consider the underlying process to be stationary, and forget about the possible observed seasonalities. We choose to work with S&P 500 time series since is known to be less affected by the cyclicity of the market. Further more, dealing with not stationary processes could provide some insights about the applicability of the dimensional reduction technique to them.

4.2 Reshaping and clustering procedures

In the previous sections, we have introduced the concepts of memory length as q and observer memory length q', which are fundamental parameters of our system. However, in the time series native form, there is no explicit reference to them. For this reason, to fix the observer memory length of each investigation, we devised a reshaping procedure for squared returns, to explicit the choice of q'. Let's consider the time series $\mathbf{y}$, which has an observer memory length q'. Being $\mathbf{x}$ a filtration of the past, it should have the form

$$\mathbf{x} = \{x_i \in \mathrm{R}^{q'} : x_i = (x_i, \ldots, x_{i-q'+1})\}$$

with $i = q', \ldots, T$.From this expression, it is easy to notice that the total length of the vector $\mathbf{x}$ is equal to $T' = T - q$, and that $\mathbf{x} \in \mathrm{R}^{T' \times q'}$.

The reshaped data can now be clustered and used for the investigation of time series dimensional reduction. As described in Chapter 3, the value of relevance and resolution depend on the coarse-graining level. For the sake of comprehensibility, and for having as continuous as possible entropies, we decided to change the coarse-graining level α by unity steps, at each algorithm iteration. In other words, the number of clusters, k, span the range $k \in \{1, \ldots, T'\} = K$. Once the interval of k's has been defined, it is possible to cluster data, following one simple requirement: the clustering method should work at a fixed k. Since we need to change the number of clusters by one at each iteration, we decided to use agglomerative methods. An advantage of these methods is that they are defined with different inter-clusters

distances, allowing us to compare them, using relevance[2] and prediction errors. For our investigation, it has been decided to use four different inter-clusters distances:

- Single linkage: where the inter-cluster distance is the closest one between two objects belonging to two different clusters;
- Complete linkage: where the inter-cluster distance is the one between two most remote objects belonging to two different clusters;
- Average linkage: where the inter-cluster distance is the average one between all the objects belonging to two different clusters;
- Ward linkage: where instead of measuring the inter-cluster distances directly, analyzes the variance of clusters.

The full description of agglomerative methods and the inter-clusters distances, used in the investigation, is provided by Mentha et al. in [10]. It is straight forward noticing that each linkage, in the general theory of dimensional resolution for time series, represents the index m in the clustering function $\mathcal{C}_\alpha^{(m)}$.

The implemented clustering procedures provide label sequences, which can be seen as our hidden states $\mathbf{s}$, living in $\mathcal{S}$-space. They are used for studying respectively predictive power and information content of the representation, described in the next sections using ideas described in Chapter 2. Before going further, it is important to underline that, given the time dependence of the time series, if we use clustering techniques with Euclidean metrics we work in a sub-optimal framework: euclidean metrics cannot provide the right weight to each observation. In a real-world situation, we are aware that the most recent observations affect in a stronger way future behaviors. The best idea for detecting optimal clustering techniques is to use Vanilla techniques. However, for the sake of simplicity, we decided to use Euclidean metrics. To not penalize too much this choice, we fixed the ARCH model parameters to the one shown in Tab.4.1.

4.3 Prediction algorithms

In this section, we introduce the prediction algorithms used in the investigation. Two main classes of prediction algorithms have been devised:

- Unsupervised predictors, in which no knowledge about the underlying stochastic process is used. For this reason, they can be used for both the synthetic dataset and the S&P 500 one.

[2]The behavior of the of information related quantities is reported in Appendix B

- Supervised predictors, in which the properties of the generating process are used. It is a straightforward understanding that this kind of predictors can be used just for the ARCH model dataset, since in the case of the S&P 500 one we have no knowledge about the generating process, and we have not to make any assumption on it.

In the next sections, we define mathematically the algorithm of both these prediction classes, even if the second predictor is used, just, for checking whether those relevant representations have behaviors that are in agreement with ARCH model properties.

4.3.1 Unsupervised predictors

As already mentioned in Chapter 2, in the case of unsupervised predictors, supposing Markovian dynamics in the hidden space, it is possible to model the system as a jump process between states represented by the clustering labels. Starting from hidden space predictions it is possible to predict real observables future values. So it is convenient to distinguish prediction algorithms working in real space and the ones working in hidden space. The predictivity in the $\mathcal{Y}$-space is affected from the one in the $\mathcal{S}$-space, as it is possible to deduce from Eq.(2.3). For the sake of consistency, let's introduce for first the predictors working in $\mathcal{S}$-space.

Given the time series, fixed the observer memory length q' and the number of clusters k, and once the reshaping of the dataset has been implemented, it is possible, applying a clustering algorithm, obtaining clustering labels, representing the hidden states of the process. For predicting the future s-state of the process, at first, we divided the observer knowledge and the corresponding labels, into two groups, test and train set, respectively with total length T'_{Test} and T'_{Train}.

$$\mathbf{x}^{Train} = \left(x_1^{Train}, \dots, x_{T'_{Train}}^{Train}\right) \qquad \Rightarrow \qquad \mathbf{s}^{Train} = \left(s_1^{Train}, \dots, s_{T'_{Train}}^{Train}\right)$$

$$\mathbf{x}^{Test} = \left(x_1^{Test}, \dots, x_{T'_{Test}}^{Test}\right) \qquad \Rightarrow \qquad \mathbf{s}^{Test} = \left(s_1^{Test}, \dots, s_{T'_{Test}}^{Test}\right)$$

By considering the train set, we computed the empirical probabilities of observing a jump from the hidden state i to the hidden state j, the transition matrix $\tilde{M}_{i,j}$, with size $k \times k$,

$$\tilde{M}_{i,j} = \frac{N_{i,j}}{N_i} \tag{4.2}$$

with $N_{i,j}$ the number of transition between i-state and j-state, and N_i the number of jumps starting from state i. All these quantities have been computed in the train set. The transfer matrix can be used as a predicting tool for the time series future

behavior. In fact, it contains an unbiased estimate of the jumping probability among clusters, if we assume that the time for which we observed the phenomenon is much longer than the autocorrelation time of the process. This can be checked in the case of an ARCH sample, but not in the case of real financial data, since we should assume the data to be sampled from a well-defined stochastic process.

Now, using the test set, it is possible to study the performance of predictions realized with the transfer matrix, $\tilde{M}_{i,j}$. For studying transfer matrix performance, it has been decided to compute a prediction error, $\tilde{\epsilon}_{t+1,t}$, defined as

$$\tilde{\epsilon}_{t+1,t} = 1 - M_{s_t,s_{t+1}}$$

where s_t is the hidden state of the squared return at the time t, and $M_{s_t,s_{t+1}}$ is not a transfer matrix but a function that can be easily computed starting from its knowledge, using Bayesian inference. What is the reason for which we have not used the transition matrix? It is described further in this section, but it is strictly related to the strong empiricity of the transfer matrix. It is clear that for $k = 1$, the performance of the prediction is perfect, and this is in perfect agreement with Chapter 2 findings. However, it is possible to notice that a problem arises, in the case of $k = T'$. In this case, each line of the transfer matrix have a unitary entrance and all the others equal to 0, in other words, there are not enough statistics to evaluate the predictor's performance.

For solving this problem, we have decided to assign a weight for each prediction

$$\epsilon_{t+1,t} = \frac{1 - M_{s_t,s_{t+1}}}{\Sigma^2_{s_t,s_{t+1}}} \tag{4.3}$$

where $\Sigma^2_{s_t,s_{t+1}}$, is the variance associated to the jump between state s_t and state s_{t+1}. In order to compute $\Sigma^2_{s_t,s_{t+1}}$ and to consider a prior knowledge on possible values of the transfer matrix, we approached the problem using Bayesian inference. The simplest way to look at the problem is to look at $\tilde{\epsilon}_{t,t+1}$ as the mean value a Bernoulli variable, with probability $M_{s_t,s_{t+1}}$ to make a good prediction and $1 - M_{s_t,s_{t+1}}$ to mistake it[3]. At the same time we can use Bayesian inference for estimating the posterior jumping probabilities, the entrances of the transfer matrix. They follow a multinomial distribution. If we assume to be stuck in hidden state i, and having N_i jumps out from it, and if we indicate with $N_{i,j}$ the number of jumps between i and j, the likelihood follows the expression

$$P(N_{i,1}, \ldots, N_{i,k}, N_i | M_{i,1}, \ldots, M_{i,k}) = \frac{N_i!}{N_{i,k}! \ldots N_{i,1}!} M_{i,1}^{N_{i,1}} \ldots M_{i,k}^{N_{i,k}}$$

[3]It is important to underline that the value one of the variable related to the prediction error is associated to the mistake in the prediction, and the value zero to the possibility to get the right prediction

such that

$$\sum_{l=1}^{k} N_{i,l} = N_i \qquad \text{and} \qquad \sum_{l=1}^{k} M_{i,l} = 1$$

with k the total number of clusters, $l \in \{0, \dots, k\}$, and T'_{train} the total number of points. Using the Train set we need to estimate $\{M_{i,1}, \dots, M_{i,k}\}$ $\forall i \in \{0, \dots, k\}$, and, knowing the cluster to which each Test point belongs, we are able to check the prediction performance. For estimating $\{M_{i,1}, \dots, M_{i,k}\}$ $\forall i \in \{0, \dots, k\}$ we used Bayesian model averaging. Let's start defining what the prior knowledge on the entrances of the transfer Matrix is. Using the concept of conjugacy of probability distributions is possible to understand that the correct prior to use is the Dirichlet Distribution:

$$P(M_{i,1}, \dots, M_{i,k} | \alpha_{i,1}, \dots, \alpha_{i,k}, \alpha_i) = \frac{\Gamma(\alpha_i)}{\Gamma(\alpha_{i,1}) \dots \Gamma(\alpha_{i,k})} M_{i,1}^{\alpha_{i,1}-1} \dots M_{i,k}^{\alpha_{i,k}-1}$$

such that

$$\sum_{l=1}^{k} \alpha_{i,l} = \alpha_i \qquad \sum_{l=1}^{k} M_{i,l} = 1$$

How can we choose the values of the $\{\alpha_{i,j}\}_{i,j=1\dots k_s}$ in order to be agnostic? The choice we made is to fix

$$\alpha_{i,j} = c \qquad \forall i, j \in \{1, \dots, k\}$$

with c a constant to be fixed depending on the simulation, and

$$\alpha_i = k \cdot c \qquad \forall i \in \{1, \dots, k\}$$

Once these quantities have been determined, we can compute the posterior probability:

$$P(M_{i,1}, \dots, M_{i,k} | N_{i,1}, \dots, N_{i,k}, N_i, \alpha_{i,1}, \dots, \alpha_{i,k}, \alpha_i) \propto M_{i,1}^{N_{i,1}+\alpha_{i,1}-1} \cdots M_{i,k}^{N_{i,k}+\alpha_{i,k}-1}$$

Now, using the Bayesian model average theory, it is possible to compute the probability that a test jump is observed between cluster i and j. Assuming that x_t belongs to cluster i, what is the probability that x_{t+1} belongs to cluster j? This can be computed via marginalization, using the expected value of the Dirichlet distribution, and the final result is that

$$\boxed{M_{i,j} = P(s_{t+1} \in j | s_t \in i, N_{i,1}, \ldots, N_{i,k}, N_i) = \frac{N_{i,j} + \alpha_{i,j}}{\alpha_i + N_i}}$$

where, for keeping the notation as simpler as possible we did not explicit that $M_{i,j}$, is the expectation value of the jump process.

Using this result and Eq.(4.3), it is now possible to compute the prediction, assuming to know $\Sigma_{i,j}$. For computing the explicit value of the variance $\Sigma_{i,j}$, we can use the assumption that the variable $\tilde{\epsilon}_{t,t-1}$ is the mean value of a variable distributed according to a Bernoulli distribution

$$\boxed{\Sigma^2_{i,j} = M_{i,j}\left(1 - M_{i,j}\right)}$$

Putting all these results together we get

$$\boxed{\epsilon^{s,unsupervised}_{t+1,t} = \frac{1 - M_{s_t,s_{t+1}}}{M_{s_t,s_{t+1}}\left(1 - M_{s_t,s_{t+1}}\right)} = \frac{1}{M_{s_t,s_{t+1}}}} \tag{4.4}$$

Starting from the prediction realized in the hidden space, as previously mentioned, it is possible to consider real space predictions. To predict the future real state the first step is to determine what is the most probable future hidden state, assuming to be stuck in state i at time t. It is defined as follows

$$\hat{s}_{t+1} = argmax_{j=1,\ldots,k}\{P(s_{t+1} \in j | s_t \in i, N_{i,1}, \ldots, N_{i,k}, N_i)\} =$$

$$= argmax_{j=1,\ldots,k}\left\{\frac{N_{i,j} + \alpha_{i,j}}{\alpha_i + N_i}\right\}$$

and the empirical probability associated with it as

$$\hat{M}_{i,j} = \max_{j=1,\ldots,k}\{P(s_{t+1} \in j | s_t \in i, N_{i,1}, \ldots, N_{i,k}, N_i)\} = \max_{j=1,\ldots,k}\left\{\frac{N_{i,j} + \alpha_{i,j}}{\alpha_i + N_i}\right\}$$

By knowing the most probable hidden state, it is possible to predict the new real observable value. The simplest way to address this prediction is using the concept of cluster centroid and standard deviation of the same cluster. In this way, our estimate for the new value of the observable coincide with the same centroid, and the variance, required for weighting the prediction error is equal to the squared value of the standard deviation of the cluster. From a mathematical point of view

$$\boxed{\hat{y}_{t+1} = \mathrm{E}_{\hat{s}_{t+1}}[y]}$$

where E[·] represents the empirical mean in the cluster and

$$\boxed{\hat{\Sigma}^2_{t+1} = \mathrm{V}_{\hat{s}_{t+1}}[y]}$$

From these results it is possible to extract the prediction error, related to real space prediction,

$$\boxed{\epsilon^{y,unsupervised}_{t,t+1} = \frac{y_{t+1} - \hat{y}_{t+1}}{\hat{\Sigma}^2_{t+1}}} \tag{4.5}$$

The behaviour of the predictions, for the two numerical experiments, are shown shown in Chapter 5 and Chapter 6.

4.3.2 Supervised predictors

As previously mentioned, it is possible to devise, in the case of synthetic time series, a supervised prediction algorithm. This algorithm predicts the future observable value of the time series, by knowing the stochastic model from which the time series has been sampled. It is clear that in the case of the ARCH model dataset, it has been possible to construct such a predictor, but not in the case of the real financial dataset. It is obvious that using the same ARCH model supervised predictor for S&P 500 dataset, the prediction error would have been very high. The supervised prediction algorithm, devised for the investigation of relevant representations for the ARCH model sampled time series is now described.

For devising this algorithm, once again, we took advantage of Bayesian inference for studying the expected value of each cluster volatility. The procedure described below is realized, for simplicity for the case of ARCH(1) model, but it can be easily generalized to our framework by considering observation as i.i.d. Using Eq.(4.1) and that $y_t \sim \mathcal{N}(0, \sigma_t^2)$ it is possible to realize that y_t^2 is distributed according a χ^2-distribution

$$y_t^2 = \sigma_t^2 z_t^2 = \sigma_t^2 \omega_t \qquad \text{with} \qquad \omega_t \sim P(\omega) = \frac{1}{\sqrt{2\pi\omega}} e^{-\frac{\omega}{2}} \theta(\omega)$$

whose statistics are

$$\mathrm{E}(\omega) = \int_0^\infty \omega P(\omega) = 1$$

$$\mathrm{E}(\omega^2) = \int_0^\infty \omega^2 P(\omega) = 3$$

This change of variable is required since the ARCH model has a linear structure. From the previous expressions it is clear that defining $\tau_t = y_t^2$:

$$\tau_t = \sigma_t^2 \omega_t \qquad \rightarrow \qquad \tau_t \sim P(\tau) = \frac{1}{\sigma\sqrt{2\pi\tau}} e^{-\frac{\tau}{2\sigma^2}} \theta(\tau)$$

where we forget about the dependence from time.

By looking at this framework from a Bayesian point of view, it is possible to realize that $P(\tau)$ is the likelihood probability of the system. We are interested in finding the Posterior probability of observing a particular squared volatility value, by fixing the cluster. In the following equation, k is the total number of clusters, at a particular iteration of the algorithm, and k_s the number of observations whose label is s. Using Bayes theorem, it is possible to compute the posterior probability as

$$P(\sigma^{(s)}|\{\tau_1, ...\tau_{k_s}\}) = \frac{P(\{\tau_1, ...\tau_{k_s}\}|\sigma^{(s)})P(\sigma^{(s)})}{\int_{\mathcal{D}(s)} P(\{\tau_1, ...\tau_{k_s}\}|\sigma^{(s)})P(\sigma^{(s)})}$$

Prior distribution knowledge is required. For this reason, we choose to consider an uninformative prior, known as Jeffreys' prior [11], whose definition is the following

$$P(\sigma^{(s)}) = \sqrt{|\mathcal{I}(\sigma^{(s)})|} \sim \frac{1}{\sigma^{(s)}}$$

where $\mathcal{I}$ is the Fisher information matrix. Before going further, it is important noticing how the result is similar to the one found for the Gaussian likelihood. This suggests that, maybe, for devising the supervised prediction algorithm, it could have been possible to consider directly y_t and not y_t^2, but to be sure further investigation are required. Furthermore, since the likelihood is distributed according to χ^2-distribution, it is straightforward to notice that it can be factorized. This assumption derive also from the requirement of independence of observations and ergodicity for the process[4]. The final expression for the Bayes theorem is:

$$P(\sigma^{(s)}|\{\tau_1, ...\tau_{k_s}\}) = \frac{\left[\prod_{l=1}^{k_s} P(\tau_l|\sigma^{(s)})\right] P(\sigma^{(s)})}{\int_{\mathcal{D}(s)} \left[\prod_{l=1}^{k_s} P(\tau_l|\sigma^{(s)})\right] P(\sigma^{(s)})}$$

One of the problems of the previous expression is that χ^2 distribution is not a stable distribution, so it is difficult to compute the expectation value of $\sigma^{(s)}$, in the hidden state s. Since the distribution is not stable, we have

$$P(\tau_1) \cdot P(\tau_2) = \frac{1}{\sqrt{2\pi\tau_1}\sigma^{(s)}} e^{-\frac{\tau_1}{2\sigma^{(s)2}}} \cdot \frac{1}{\sqrt{2\pi\tau_2}\sigma^{(s)}} e^{-\frac{\tau_2}{2\sigma^{(s)2}}} = \frac{1}{2\pi\sqrt{\tau_1\tau_2}\sigma^{(s)2}} e^{-\frac{\tau_1+\tau_2}{2\sigma^{(s)2}}}$$

[4]Look at the first section of this chapter.

However, even if the distribution is unstable, we can estimate the average value of the volatility in each cluster as follows

$$\mathrm{E}_{P(\sigma^{(s)}|\{\tau_1,\dots\tau_{k_s}\})}[\sigma^{(s)}] = \frac{\int_0^\infty d\sigma^{(s)} \frac{P(\sigma^{(s)})\cdot\sigma^{(s)}}{\sqrt{(2\pi\sigma^{(s)2})^{k_s}\tau_1\dots\tau_{k_s}}} e^{-\frac{\tau_1+\dots+\tau_{k_s}}{2\sigma^{(s)2}}}}{\int_0^\infty d\sigma^{(s)} \frac{1}{\sqrt{(2\pi\sigma^{(s)2})^{k_s}\tau_1\dots\tau_{k_s}}} e^{-\frac{\tau_1+\dots+\tau_{k_s}}{2\sigma^{(s)2}}} \cdot P(\sigma^{(s)})}$$

with

$$\int_0^\infty d\sigma^{(s)} \frac{1}{\sqrt{(2\pi\sigma^{(s)2})^{k_s}\tau_1\dots\tau_{k_s}}} e^{-\frac{\tau_1+\dots+\tau_{k_s}}{2\sigma^{(s)2}}} = \int_0^\infty d\sigma^{(s)} \frac{1}{A\sigma^{(s)k_s}} e^{-\frac{B}{2\sigma^{(s)2}}}$$

It is an inverse gamma distribution, in which $A = \sqrt{(2\pi)^{k_s}\tau_1\dots\tau_{k_s}}$ and $B = \tau_1+\dots+\tau_{k_s}$. Instead of working with the squared Volatility we can try to work with the squared precision, whose distribution is a gamma distribution. Let's try to compute the following integral:

$$\int_0^\infty d\sigma^{(s)} \frac{1}{A\sigma^{(s)k_s}} e^{-\frac{B}{2\sigma^{(s)2}}} = \frac{1}{2AB^{\frac{k_s-1}{2}}} \int_0^{+\infty} dy y^{\frac{k_s-3}{2}} e^{-\frac{y}{2}} = \frac{2^{\frac{k_s-3}{2}}\Gamma(\frac{k_s-1}{2})}{AB^{\frac{k_s-1}{2}}}$$

The result has been computed using Wolfram Mathematica software. The denominator of the previous expression behaves in the same way: for this reason, the expected value of the volatility in a cluster is:

$$\mathrm{E}_{P(\sigma^{(s)}|\{\tau_1,\dots\tau_{k_s}\})}[\sigma^{(s)}] = \frac{\frac{2^{\frac{k_s-3}{2}}\Gamma(\frac{k_s-1}{2})}{AB^{\frac{k_s-1}{2}}}}{\frac{2^{\frac{k_s-2}{2}}\Gamma(\frac{k_s}{2})}{AB^{\frac{k_s}{2}}}} = \frac{\sqrt{B}\Gamma(\frac{k_s-1}{2})}{2^{\frac{1}{2}}\Gamma(\frac{k_s}{2})}$$

Using similar reasoning it is possible to compute also the expected value of the squared volatility and the volatility raised to the fourth power:

$$\boxed{\mathrm{E}_{P(\sigma^{(s)}|\{\tau_1,\dots\tau_{k_s}\})}[\sigma^{(s)2}] = \frac{B\Gamma(\frac{k_s-2}{2})}{2\Gamma(\frac{k_s}{2})}}$$

$$\mathrm{E}_{P(\sigma^{(s)}|\{\tau_1,\dots\tau_{k_s}\})}[\sigma^{(s)4}] = \frac{B^2\Gamma(\frac{k_s-4}{2})}{2^2\Gamma(\frac{k_s}{2})}$$

Finally, the variance of the squared volatility is:

$$\boxed{\mathrm{V}_{P(\sigma^{(s)}|\{\tau_1,\dots\tau_{k_s}\})}[\sigma^{(s)2}] = \frac{B^2[\Gamma(\frac{k_s-4}{2})\Gamma(\frac{k_s}{2}) - \Gamma(\frac{k_s-2}{2})^2]}{4\Gamma(\frac{k_s}{2})^2}}$$

The result seems to be correct since we need to reach 4 elements for each cluster for computing such quantities, and if the number of elements is lower than 4, it is impossible to compute them. This can be a problem, but for solving it we can instead study the expected value of $\sigma^{(s)}$ and its variance. The result is pretty similar to the previous one.

$$\mathrm{V}_{P(\sigma^{(s)}|\{\tau_1,\dots\tau_{k_s}\})}[\sigma^{(s)}] = \frac{B[\Gamma(\frac{k_s-2}{2})\Gamma(\frac{k_s}{2}) - \Gamma(\frac{k_s-1}{2})^2]}{2\Gamma(\frac{k_s}{2})^2}$$

In conclusion, we can see that the non-Stability of the χ^2 Distribution is not a problem, since using conjugacy we can compute all the interesting quantities. However, it is important to notice how we need at least 2 points per cluster to compute the mean and the variance of clusters volatility. From some point of view, this result is reassuring: it means that to predict properly the value of cluster volatility we need to have enough good statistics, i.e. enough data.

By trying to implement this predictor another problem can be seen: it could happen that, for some values of k, we need to compute $\Gamma(u)$ with $u > 150$. These values are not computable or difficult to compute. For solving this problem we need to use some approximations for the Gamma function need to be used, the Stirling approximation formula. It states, assuming k_s to be an integer,

$$\Gamma(k_s + 1) = \sqrt{2\pi k_s}\left(\frac{k_s}{e}\right)^{k_s}$$

It is important to underline that this approximation work in a good way for high values of k_s, but not for small ones, for this reason, some problem can arise in the undersampling region of relevance resolution curves. Before taking any decision regarding the possible way to face these discrepancies. We can try to look at computational results for having an idea of what is the amount of this discrepancy. First of all, we need to compute the approximation provided by, Stirling formula for the quantities previously computed. They are the following

$$\mathrm{E}_{P(\sigma^{(s)}|\{\tau_1,\dots\tau_{k_s}\})}[\sigma^{(s)}] = \frac{\sqrt{B}\Gamma(\frac{k_s-1}{2})}{2^{\frac{1}{2}}\Gamma(\frac{k_s}{2})} \sim \left(\frac{k_s-3}{k_s-2}\right)^{\frac{k_s-1}{2}}\left(\frac{eB}{k_s-3}\right)^{\frac{1}{2}}$$

$$\mathrm{E}_{P(\sigma^{(s)}|\{\tau_1,\dots\tau_{k_s}\})}[\sigma^{(s)2}] = \frac{B\Gamma(\frac{k_s-2}{2})}{2\Gamma(\frac{k_s}{2})} \sim \left(\frac{k_s-4}{k_s-2}\right)^{\frac{k_s-1}{2}}\left(\frac{eB}{k_s-4}\right)$$

$$\mathrm{E}_{P(\sigma^{(s)}|\{\tau_1,...\tau_{k_s}\})}[\sigma^{(s)4}] = \frac{B^2\Gamma(\frac{k_s-4}{4})}{2\Gamma(\frac{k_s}{4})} \sim \left(\frac{k_s-6}{k_s-2}\right)^{\frac{k_s-1}{2}} \left(\frac{eB}{k_s-6}\right)^2$$

It is, thus, possible to generalize this to any moment

$$\mathrm{E}_{P(\sigma^{(s)}|\{\tau_1,...\tau_{k_s}\})}[\sigma^{(s)h}] \sim \left(\frac{k_s-2-h}{k_s-2}\right)^{\frac{k_s-1}{2}} \left(\frac{eB}{k_s-2-h}\right)^{\frac{h}{2}}$$

From these results it is also possible to compute the new values for the volatilities:

$$\mathrm{V}_{P(\sigma^{(s)}|\{\tau_1,...\tau_{k_s}\})}[\sigma^{(s)2}] \sim \left(\frac{eB}{k_s-6}\right)^2 \left[\left(\frac{k_s-6}{k_s-2}\right)^{\frac{k_s-1}{2}} - \left(\frac{k_s-4}{k_s-2}\right)^{(k_s-1)} \left(\frac{k_s-6}{k_s-4}\right)^2\right]$$

There are some important elements on which it is important to focus our attention:

- First of all, we can notice how the second term in the expression for the Variance is smaller than the first because the basis is smaller than one and it is raised to the square. However, there could be some situations in which the variance can be negative. Since the expression is not simple to be studied, we can try to look at the numerical solution for approach the discussion for this possibility;
- It is important to underline how the use of the Stirling approximation enable-senables us also to study the situation in which $k_s < 4$, however, this is an unphysical case that we want to avoid;
- Again we would like to underline how this approximation works well just in the situation in which we have a high amount of points for each cluster. This is compatible with the first comment we made. Before implementing this solution we need to make some other checks.
- Another problem is that we are in a multidimensional case, but this problem can be solved by observing that the various variables are independent.

From the numerical analysis, it is possible to find that the result is computable just in the following range of values:

$$k_s \in \mathbb{N}_0$$

This implies that for our interest, we can use the following approximation just in the case of $k_s \geq 4$, while, in the other case, we can use the real formula for the volatility since, in that situation, it is possible to compute it, however, better estimations can

be implemented, for example keeping using the right expression of the variance for high enough values of k_s and then pass to the approximation provided by Stirling formula. The last k_s that has been computed using the exact formula is $k_s = 150$, and for this reason, we have decided to use Stirling approximation just in the case of $k_s > 150$. Independently from this, from now on we can implement the new Bayesian supervised predictor. Its results are shown in the next Chapters.

For what concerning the performance of supervised prediction algorithm, we decided to compute the prediction error in a similar way to the unsupervised prediction error

$$\boxed{\epsilon_{t,t+1}^{supervised} = \frac{\sigma_{t+1}^2 - \mathrm{E}_{P(\sigma^{(s)}|\{\tau_1,\dots\tau_{k_s}\})}[\hat{\sigma}_{t+1}^2]}{\mathrm{V}_{P(\sigma^{(s)}|\{\tau_1,\dots\tau_{k_s}\})}[\hat{\sigma}_{t+1}^2]}} \tag{4.6}$$

where $\hat{\sigma}_{t+1}$ is the estimate of volatility, by knowing the hidden state, s_{t+1}, of the observation at time $t + 1$.

4.4 Relevance, resolution and total information

Once the prediction has been realized, we can compare prediction error results with the results provided by the analysis of relevance, resolution and total information graphs. In this graphs, we plot the entropies as a function of the coarse graining level α. From what we have mentioned in Chapter 2 and in Chapter 3, we should expect that representation characterized by higher relevance are also the ones having better prediction performances. In other words, from a graphical point of view, we should expect that higher relevance-resolution curve, for different inter clusters distances, should have lower prediction errors, both in the unsupervised and supervised case. In the next two chapters, two experiment are described, and comparing prediction power and relevance levels, we are able to extract some general informations about relevant representations, and thus maximally informative ones.

Chapter 5

Application I: comparison of clustering methods

For investigating numerically relevant representations we decided to devise two numerical experiments, the first of which is introduced in this chapter. The ideas that brought us to devise it are described in the first section, while in the second section, we show the numerical results and we take some conclusions, trying to extend them to more general frameworks.

5.1 Clustering method optimization

As stated in Chapter 2, to find a system relevant representation, we need to optimize the function $\mathcal{C}_\alpha^m$, in such a way to obtain a maximally informative and maximally predictive representation of system state. This optimization problem can be divided into two sub-problems. The first consisting of the optimization of the coarse-graining level α, once we fixed the clustering algorithm m. While the second sub-problem, consists in optimizing $\mathcal{C}_\alpha^m$ as a function of m. The relevant representation is obtained when both the coarse-graining level α and the clustering method m have been optimized. As described in [7], the first sub-problem is related to the optimization of the functional, reported in Eq.(3.4). Since functional optimization problems have been widely studied, in this thesis, we have not focused our attention on finding a solution for it. However, the optimization problem of m has not a solution, and for this reason in this section we describe the experiment we set up to solve it.

The keystone to solve this optimization problem is the assumption that relevant representation should be both maximally informative and maximally predictive. By looking at the prediction power of a clustering algorithm, and at the amount of information about the generating process it can extract from the dataset, it is

possible to understand that not all clustering methods m are identical. Driven by this hypothesis, we are wondering whether is there relevance and prediction-driven approach to detect what clustering method provides the most relevant representation. In the case of agglomerative clustering methods, the problem of optimizing m can be translated in finding the most predictive inter-cluster linkage. Is this property universal or does it depend on the dataset itself? At first glance, we can suppose that the hierarchy of clustering methods is a property dependent on the charcteristics of the dataset itself. Furthermore, it is important to underline another aspect: the used clustering methods are devised by using Euclidean distances in $\mathbb{R}^{q'}$. However, since we are dealing with time-dependent processes, it could be true that the most predictive clustering algorithm is the one characterized by a non-parametric distance. For answering these questions, we start considering an ARCH model dataset. The results found for the synthetic dataset are then compared with the ones of the S&P 500 dataset. We started our analysis from a synthetic dataset because, by considering the results proposed by Crutchfield and Shalizi in [1], it is analytically possible to detect ARCH model causal states. The causal state of an ARCH model, at time $t+1$, is the squared returns set, considered from time t to time $t-q+1$, assuming the process to have a memory length equal to q. In other words, the causal states consist in the filtration of the past with $q' = q$, and their knowledge enables us to have the best "theoretical" predictive power.

Starting from these considerations, we decided to consider an ARCH model with $q = 4$, and consequently, we fixed the user knowledge memory length to $q' = 4$. After the dataset has been generated, and the various clustering procedures implemented, at first we represented prediction error vs relevance. After, we computed relevance as a function of the same α, for each method m. In this way, we should be able to obtain two hierarchies, one for the predictive power and one for generative model information content extracted by inter-cluster distances. The results of this investigation are shown in the next section.

5.2 ARCH model: Results

In this section, we present the computational results of the first numerical experiment developed for investigating relevant representations. In particular with this experiment, as previously mentioned we could find two different type of hierarchies, one dependent on predictive power and one on information content of clustering labels. Regarding the prediction power, it is possible to obtain several different hierarchies, depending on the type of prediction we consider (supervised or unsupervised, real space prediction, or hidden space prediction). Let's start considering the unsupervised, $\mathcal{S}$-space, prediction: it is shown in Fig.5.1, where we

plotted prediction error as a function of the resolution, i.e. a way to encode the coarse-graining level.

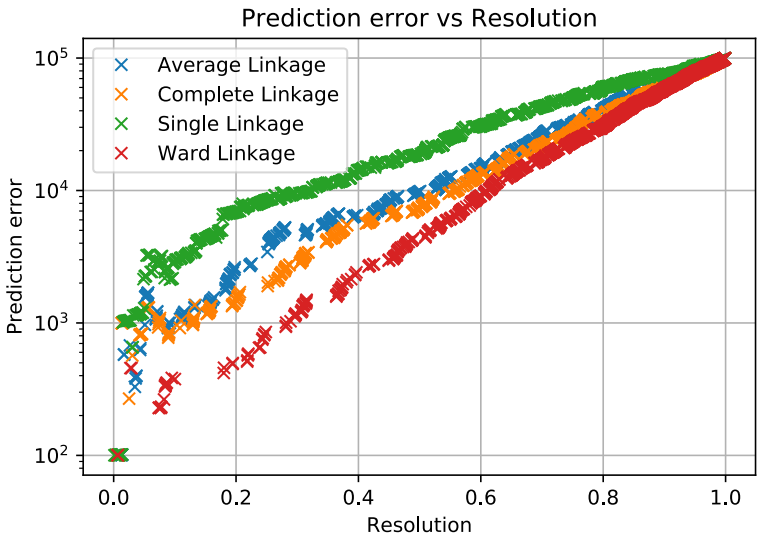

Figure 5.1: Prediction error vs Resolution, ARCH dataset (hidden variable space, unsupervised prediction)

From this figure, it is possible to detect a hierarchy in the inter clusters distances: the most predictive method m is the Ward Linkage, while the less predictive is the Single linkage. Average linkage and Complete linkage behave in similar ways[1]. It is important noticing how higher resolutions are characterized by higher prediction errors, this is some way expected since with higher coarse-graining levels we have a higher number of clusters and so fewer statistics for estimating the next step label, causing the prediction to get worse and worse.

Once the hidden space predictions have been realized, as previously mentioned, it is possible to consider predictions in the real space, $\mathcal{Y}$. For translating hidden space predictions in real space ones, we need to deal with the properties of the clustering algorithms and the definition of their centroids, see Chapter 2. A plot analogous to the previous one is shown in the following figure, where the prediction error, computed in the $\mathcal{Y}$-space is shown as a function of resolution

[1]We suppose that this result is related to the definition of the agglomerative methods linkages

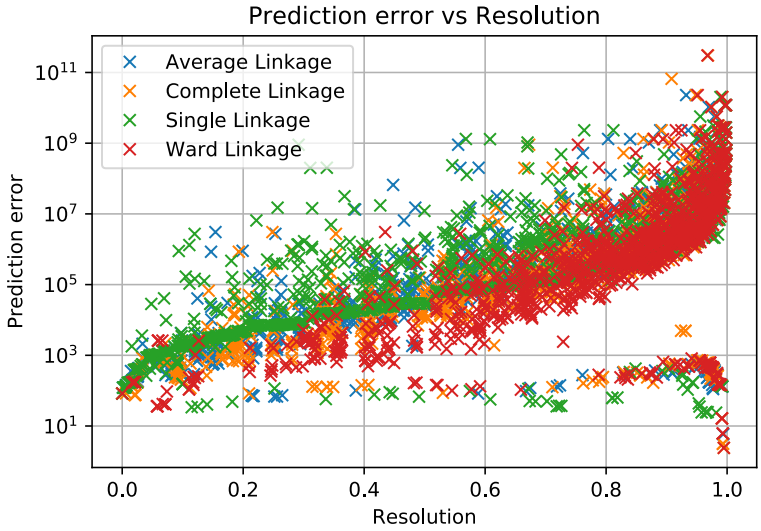

Figure 5.2: Prediction error vs Resolution, ARCH dataset (real variable space, unsupervised prediction)

The hierarchy of clustering algorithms found in Fig.5.2 seems the same detected using the prediction in the hidden space, even if it is very noisy. This result reassures us about the strict connection existing between the hidden space, we constructed using clustering algorithms and their labels, and the space in which real observables live. However a problem seems to appear: the prediction is very noisy. The noise detected in the prediction can be easily explained by noticing that the prediction in the hidden space is, in some sense, discrete, while the prediction in the real space is continue. In particular this effect can also be addressed to the fact that the prediction is not built from the beginning in the real space, but in the hidden variable one. In fact, we are able to predict the future value of the observable only after predicting the next label value (it can be seen as a sort of translation). A double uncertainty affect the prediction: a part coming from the $\mathcal{S}$-space to $\mathcal{Y}$-space translation, and the other coming from the empirical nature of the proposed prediction algorithm. This is the effect of dimensional reduction. However, this problem does not affect the hierarchy detected, in fact as previously mentioned, it is identical to the one found for prediction error in hidden space. Further confirmation of this hierarchy correctness could be obtained by analyzing supervised predictions.

Before comparing these results with values of relevance and total information, we would like to compare the prediction errors computed using transfer matrix techniques, with the prediction errors computed using an algorithm having precise knowledge of the underlying generative model. Since the prediction of the two algorithms has been developed using different quantities, we decided to keep

prediction plots separated. In Fig.5.3, supervised prediction error is shown as a function of resolution.

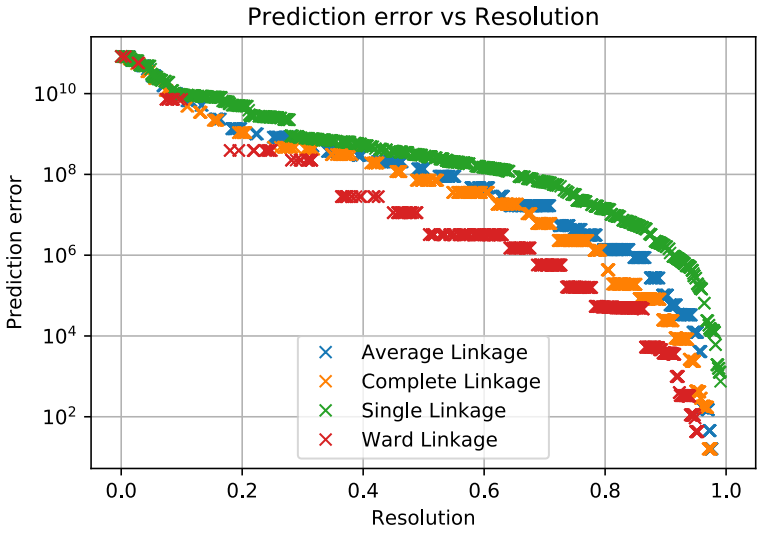

Figure 5.3: Prediction error vs Resolution, ARCH dataset (Supervised predictor)

Once again the hierarchy detected is in perfect agreement with the one found considering unsupervised prediction error, in the $\mathcal{S}$-space. Even if we used different prediction algorithms, Ward Linkage can predict in a better way future behaviors of the system. There are some differences in the curves of unsupervised and supervised predictions: for the first type of prediction algorithms the curve is increasing, while in the second it is decreasing. However, these different behaviors do not imply the presence of problems in the prediction, because the intrinsic nature of the algorithms is different. In fact, for explaining the behavior shown by unsupervised prediction algorithms, we can see what happens if we increase k: the transition matrix has fewer statistics, and so predictions get worse and worse. This is not the case for supervised predictions, whereby by increasing k, we reduce the number of points used for estimating volatility, and thus we have always a better estimate. By looking at the figures, it is clear that the results are in perfect agreement with theoretical expectations. It is important to underline that this result is however only approximate, since better algorithms for ARCH model volatility predictions can be implemented. This is not a problem since we used it just as a backup check.

Since the predictions we make are in agreement, we can try to study the behavior of the relevance and total information as a function resolution. The behaviour of these two quantities are shown respectively in Fig.5.5 and Fig.5.4.

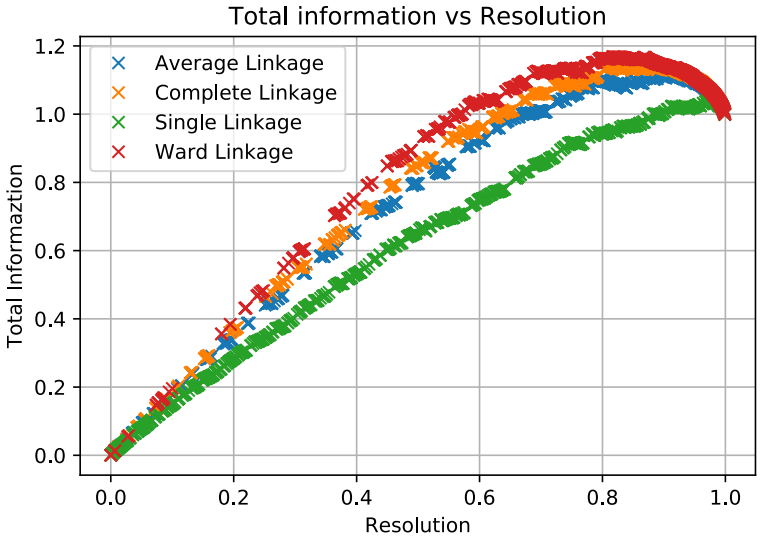

Figure 5.4: Total information vs Resolution, ARCH dataset

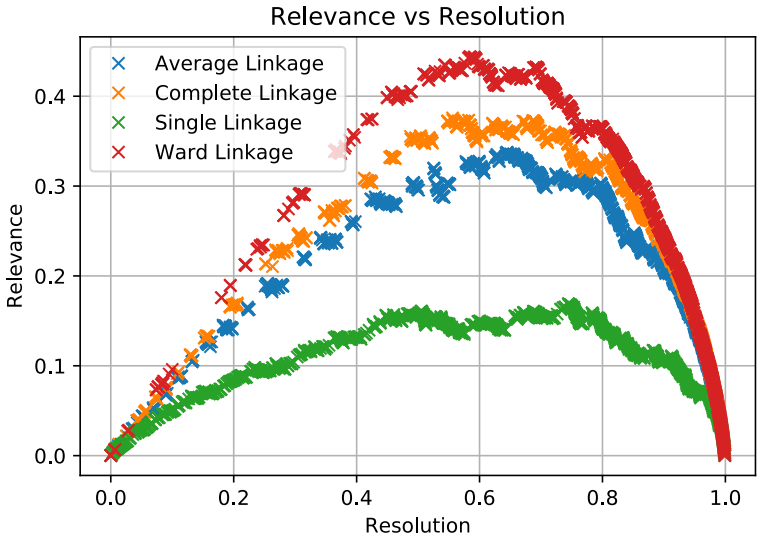

Figure 5.5: Relevance vs Resolution, ARCH dataset

By looking at Fig. 5.4, we can notice a hierarchy among inter clusters distances, meaning that the highest amount of total information[2] extracted is the one of Ward

[2]We should remember that, as mentioned in Chapter 3, it represents the sum of relevance and resolution

linkage. However, being total information the sum of relevance and resolution, it does not contain only information about the generating process, but it is also affected by other components: the amounts of bits required to assign each observation to a cluster and, thus, by noise[3]. As previously mentioned, the information about the generating process is quantified by relevance. Its behaviour is shown in Fig. 5.5. Once again, a hierarchy is found, and this hierarchy is in perfect agreement with the one found in the analysis of prediction errors and total information. This can be seen as a first proof of the strict relation existing between prediction power and information content of a time series. The discussion about this result and some ideas for further investigations are developed in Chapter 7.

5.3 S&P 500: results

A similar analysis for the one realized for the ARCH model time series has been implemented for S&P 500 time series, the only difference is that for this case we do not know exactly the underlying model from which data has been generated. For this reason, it has not been possible to construct a supervised prediction algorithm. The results are shown in Fig.5.6, Fig.5.7 and Fig.5.8.

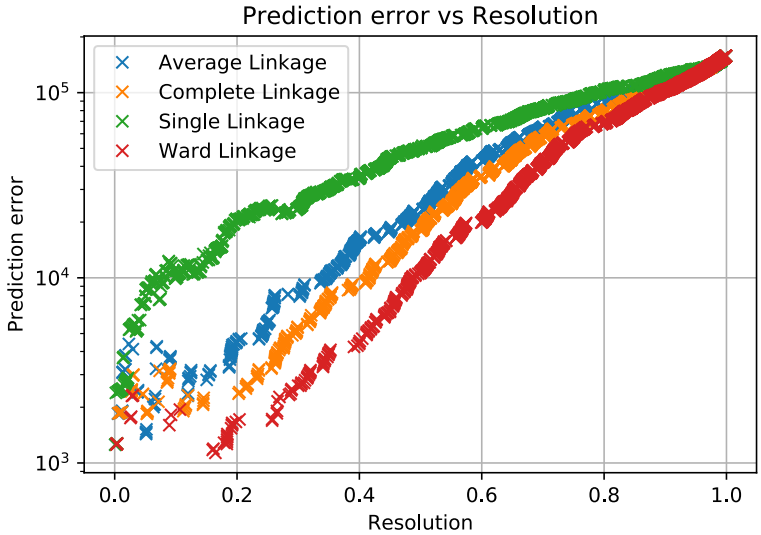

Figure 5.6: Prediction error vs Resolution, S&P 500 dataset (hidden variable space, unsupervised prediction)

[3]These concepts are described in Chapter 3

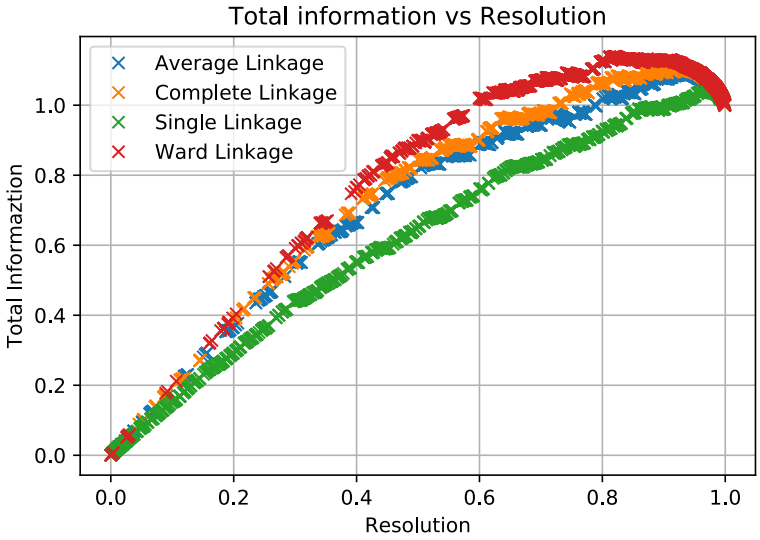

Figure 5.7: Total information vs Resolution, S&P 500 dataset

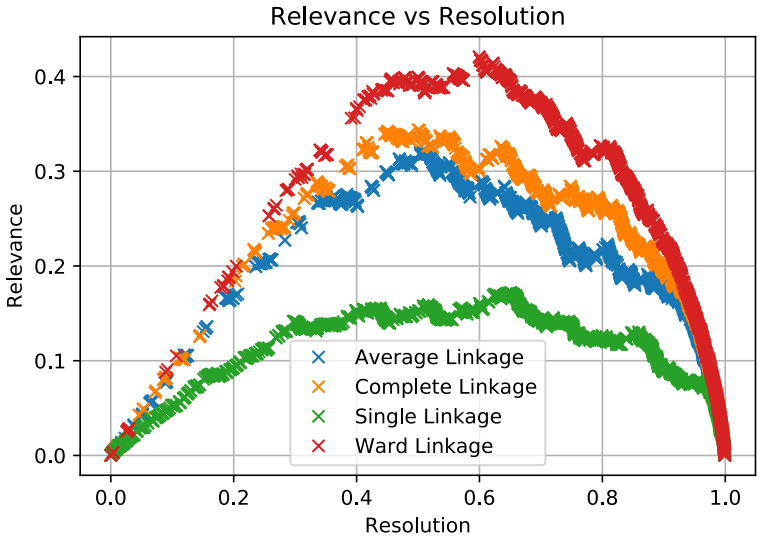

Figure 5.8: Relevance vs Resolution, S&P 500 dataset

The results of S&P 500 show hierarchies among inter-cluster linkages similar to the ones found in the case of the synthetic time series, in the case of information content and prediction error in hidden space. Once again the hierarchy found using prediction errors hidden variable spaces is in perfect agreement with the classification provided by relevance plots. The results we found are even stronger,

the classification detected using the ARCH model and S&P 500 time series are identical. This result could suggest that these properties of agglomerative methods linkages are not specific, but maybe they have some sort of universality, or maybe they are strictly dependent from dataset properties. However, since we considered just two datasets, it could be seen just as supposition. For investigating the dependence of the hierarchies from dataset properties, further investigations are required. A good hierarchy, unfortunately, is found just for the case of hidden space predictions, for the real space predictions it is not any more clear whether this hierarchy still exists. Furthermore the same problem detected in the case of ARCH model time series arises: the prediction is very noisy. In the case of this prediction, some numerical instabilities can be detected, and thus we decided to consider this result as not reasonable, and to not admit it in the investigation. However this does not influence the result we found, since the only problem detected is the high noise of the real space prediction. In further investigation, $\mathcal{Y}$-space prediction is going to be mastered, and new prediction algorithms, less affected by noise, are going to be built. As previously mentioned, a deeper discussion about the results and their explanation is reported in the last chapter of this thesis work.

5.4 Information bottleneck method

In this section, we try to compare the classification of agglomerative methods linkages determined in the previous investigation with the one detected using the Information bottleneck method, introduced by Bialek et al. in [4] and [12]. This technique was devised for finding the best trade-off between accuracy and complexity when clustering a random variable $\mathbf{x}$, given a joint probability distribution $p(\mathbf{x}, \mathbf{w})$ between $\mathbf{x}$ and an observed relevant variable $\mathbf{w}$. The information bottleneck can also be viewed as a rate-distortion problem, with a distortion function that measures how well $\mathbf{w}$ is predicted from a compressed representation $\mathbf{s}$ compared to its direct prediction from $\mathbf{x}$. This interpretation provides a general iterative algorithm for solving the information bottleneck trade-off and calculating the information curve from the distribution $p(\mathbf{x}, \mathbf{w})$. It is clear, from Bialek's definition, that the information bottleneck method is a technique for investigating the dimensional reduction quality of high dimensional data. For realizing the dimensional reduction, Bialek et al. defined two important quantities: the compression rate, the mutual information between data and compressed representation, $I(\mathbf{x}, \mathbf{s})$, and accuracy rate, the mutual information between compressed representation and relevant variable, $I(\mathbf{w}, \mathbf{s})$. To find the best representation for the state of the system they devised a Lagrangian function to be optimized

$$\mathcal{L}(\mathbf{x}, \mathbf{s}, \mathbf{y}) = I(\mathbf{x}, \mathbf{s}) - \beta I(\mathbf{w}, \mathbf{s})$$

where β is a Lagrange multiplier. It is clear that some similarities exist between the information bottleneck method and the technique we devised for detecting relevant representation: the compression rate can be seen as the analogous of resolution and accuracy rate can be compared to relevance. A more mathematical analysis of the strict relation between relevance and accuracy rate is described in [6].

Starting from this comparison it is clear that, the information bottleneck method can provide a classification of clustering methods, based on accuracy concept [13]. We would like to compare the two classifications. First of all, we should define what **w** is in time series framework. Unfortunately. it is easy to define **w** just for the ARCH model time series because we know the causal states of the problem. For a real time series, it is not possible to detect causal states. For this reason, we decided to compare the linkage hierarchies just for the ARCH model time series. As previously mentioned, one of the main inconveniences of this method is the dependence on generative model structural knowledge, and in the case of real time series it is not possible to know them. Before going further in the comparison it is fundamental to define, for our synthetic time series, what **w** and **s** are. **s** are the labels obtained through the clustering procedure, while **w** is the exact value of return squared volatility at time t. By knowing them it is possible to compute accuracy as a function of k and compare it with the relevance. The mutual information has been computed using precompiled libraries in Python, according to the results described in [14]. The two hierarchies are shown in Fig.5.9 and 5.10.

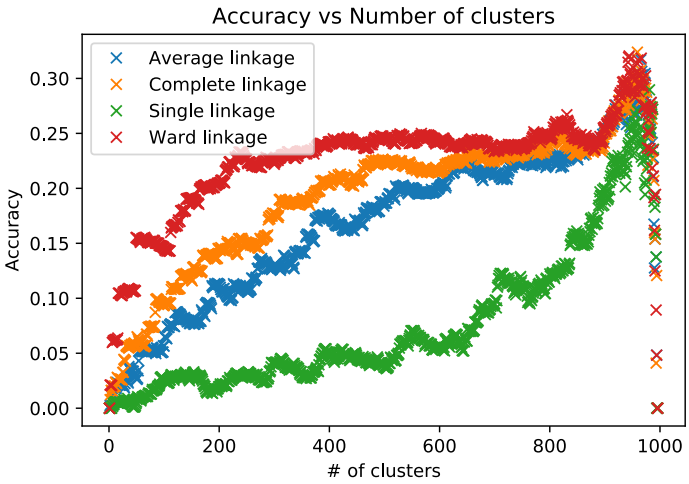

Figure 5.9: Accuracy vs k, ARCH dataset

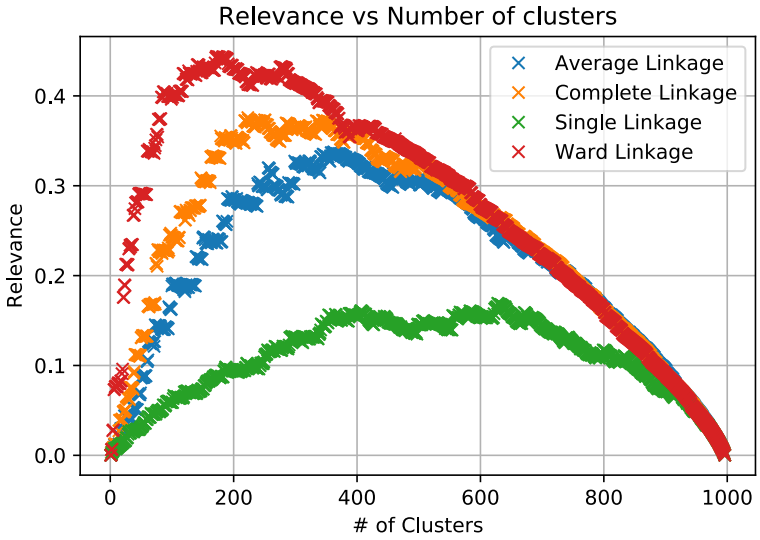

Figure 5.10: Relevance vs k, ARCH dataset

These results show how the detected hierarchy between clustering methods is the same. However, for having a better comparison of the two different approaches, it could be possible, since we know the causal states in the case of ARCH model, to solve the information bottleneck problem analytically. In further works, we aim to solve it analytically. A deeper discussion about these results is realized in Chapter 7.

Chapter 6

Application II: detecting degrees of freedom of a system

6.1 Relevance and model structure

The previous chapter showed an experiment devised for detecting the most predictive clustering linkage, given a time series. This property depends on both data structure and inter-clusters distance properties. In this chapter, instead, we are interested in investigating the relationship between the generative model of a time series and its relevant representation. Using relevance, resolution, and total information concepts, is it possible to discriminate more complex models from simpler ones? In this framework, comparing the information content of the representation and their prediction powers, we are interested in understanding whether is it possible to approximate complex models with simpler ones, in the case of a small sample size, by keeping a good predictive power. Can these simpler models predict in a better way the future behavior of time series, with respect to complex ones? For addressing these questions, we devised a numerical experiment, based on the ARCH model datasets, whose properties are described in Chapter 4.

Let's consider, once again, an ARCH model time series with a memory length $q = 4$. For this investigation, however, unlike the previous numerical experiment, we have considered the observer memory length q' assuming several values

$$q' \in \{1,2,4,6,8\}$$

By considering different values of q', we implemented a dimensional reduction, providing us a sequence of labels. The set of labels is thus used for computing

prediction errors in an unsupervised way. We consider once again, only unsupervised prediction, because we are voluntarily mistaking the choice of causal state, and thus the result of the supervised prediction would have been both trivial and meaningless. At this point, we are interested in investigating, which value of q' is the most predictive. Is it the one characterized by $q' = q = 4$? For the sake of conciseness, the results of the investigation is shown just for the Ward linkage case. This restriction is not problematic, in fact, in the previous chapter we showed Ward linkage to be the most predictive and informative inter-cluster linkage. However, the results for the other inter-cluster distances are identical to the ones found for it. Is then the analysis of predictive power and information content of a representation able to detect the true dimensionality, i.e. the real memory length, of the system? In the following experiment, we try to answer these questions, by comparing prediction errors and information content of the representation.

6.2 ARCH model: results

In this section, we compare the results obtained using prediction algorithms, and relevance, resolution, and total information estimates, for the different values of q' and different values of the sample size T. For the problems encountered in Chapter 5 regarding real space unsupervised predictions, we decided to consider only prediction algorithms working in hidden space. The total length of the time series spanned the set $T \in \{100,500,1000\}$, since for higher value it would have required to high computational power.

The plots of the information content-related quantities, for $T \in \{100, 500, 1000\}$, are here reported with the same order.

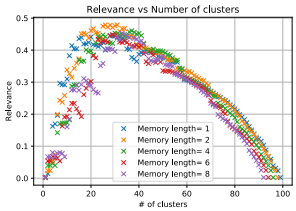

Figure 6.1: Relevance vs k, $T = 100$

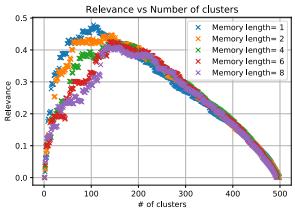

Figure 6.2: Relevance vs k, $T = 500$

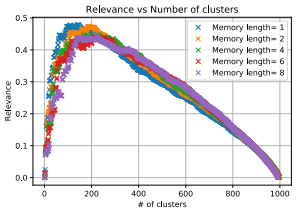

Figure 6.3: Relevance vs k, $T = 1000$

The plots of the prediction errors in the hidden space, for $T \in \{100, 500, 1000\}$, are here reported with the same order.

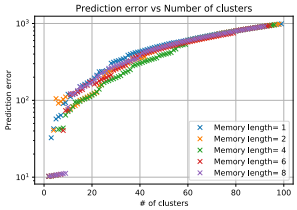

Figure 6.4: Prediction error vs k, $T = 100$

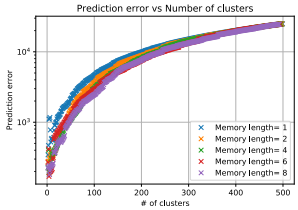

Figure 6.5: Prediction error vs k, $T = 500$

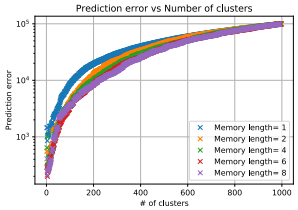

Figure 6.6: Prediction error vs k, $T = 1000$

Let's look, at first, separately at the results provided by prediction error plots and by relevance plots. By looking at prediction error behavior, in the hidden space, it is clear that models with $q' \geq 4$ predict future labels in a better way, for almost all the value of T. From a theoretical point of view, this result is someway expected since, considering longer memory than $q = 4$, helps in catching the real model structure. However, in the case of short time series, this is not completely true, and, as expected, all the prediction could have similar behavior. This idea is in agreement with the numerical results. At first sight, it could be strange that also for $q' = 6{,}8$ we have similar results to $q' = 4$. However, we should not be amazed by this behavior of the system, because, in this case, the prediction algorithm just considers a higher number of parameters, that we can suppose to be averaged out for T large enough. The results for the relevance values as a function of k show a completely different behavior: the higher relevance curve is the one for $q' = 1$, and the curve for $q' = 4{,}6{,}8$ behave almost identically. This second observation guarantees us that the useless parameters, in the case of $q' = 6{,}8$, should average out. On the other hand, there is no more agreement between prediction error behaviors and information content: the hierarchy shown is different. How can we explain this result? By looking at the relevance behavior as a function of k, it looks like that there are less relevant degrees of freedom caught from the entropic investigation. A sort of effective dimension notion should be defined: it is not a property of the same underlying model, which in principle is unknown, but it is a property of the dataset itself, which should change with its dimension. The behavior of relevance as a function of T, in some way, confirms our supposition. In fact, by increasing the value of T, it seems that the relevance hierarchy is getting flatter. Regarding the real space prediction error, we found once again a very noisy plot. In this case, the hierarchy has not been detected. As mentioned in the previous chapter, this result is related to continuous nature of the prediction algorithm in the real space, and to the too naive prediction we have implemented. It is important to remember, in fact, that in opposition to the prediction devised in the hidden space, realized considering probability distributions, real space prediction has been realized by

choosing a particular value for future labels (using $argmax$ function[1]), instead of a p.d.f. Obviously, now a question arise: is the prediction rigged? To answer this question, further and more specific investigations are required. However, we suppose that these results are not a symptoms of rigged predictions, but they are due to the too coarse nature of the implemented prediction error. This result does not affect our assumption for the existence of an effective memory length extracted by the information content.

In conclusion, we have decided to not implement the same analysis for S&P 500 dataset, because, without knowing the exact generative model, i.e. without knowing the exact memory length of the process, no exact conclusion can be drawn. The discussion of these results and some ideas for improvements of the prediction algorithms are present in Chapter 7.

[1]See Chapter 4

Chapter 7

Conclusions

In this section, we make a quick recap of the results of the two numerical experiments, taking some conclusions about general features and properties of relevant representations of the system state. Lately, we try to propose some ideas for starting future investigations.

Let's start considering the results of the first set of numerical experiments. By looking at both the synthetic and real datasets, we can notice that the hierarchy found by the prediction error analysis and the one of generative process information content are in agreement among themselves, as shown in Fig.5.1, Fig.5.5, Fig.5.6 and Fig.5.8. This result suggests that, for a general time series, the representation, containing the highest amount of information regarding the underlying stochastic process is also the one providing the best prediction about the future label of the time series, at least in the case of agglomerative clustering algorithms. This behavior seems to be observed also in real space predictions, at least for the ARCH model time series. In this case, the hierarchy is still valid, even if Fig.5.2 is strongly affected by noise. The main reason behind the detected noise, as mentioned in Chapter 5, is related to the continuous nature of real space predictivity. In other words, observables do not belong to a set of integers, as the values of the labels, but they belong to real numbers. For S&P 500 dataset, the noise effect is so strong to delete completely the hierarchy detected by hidden space predictions. There are various reasons for this result: the first is related to the nature of the time series, and the correlated memory length. In fact, being S&P 500 dataset a real observable sequence, it is plausible that more recent pasts affect strongly the actual return value, so euclidean metrics cannot catch this generative process property. There is a second explanation: the prediction algorithm we implemented is too naive for generalizing predictions from hidden space to real space. For this reason, in future investigations, less coarse predictions need to be implemented. However, this result does not worry us, because on the other hand, by using supervised prediction error in real space, we obtain, once again, the same hierarchy. Starting

from these results, we can state that clustering methods extracting a higher amount of generative process information are also the ones that predict in a better way the future behavior of the system. The problem of finding the optimal coarse-graining level is still unsolved, but, as previously mentioned, it consists in a Lagrangian optimization problem. To be sure about these conclusions, it would be necessary to consider different clustering techniques. As reported in Chapter 2, for the sake of simplicity, we used only sub-optimal methods[1] for clustering time series. The next step for this investigation could be to consider labels obtained through the vanilla technique, a tool introduced in Chapter 3. For having further confirmation that the hierarchy detected is the right one, we can try to compare it with the analytical results of the optimization problem contained in the information bottleneck method, of which up to now we used only the numerical approximation.

Regarding the second numerical experiment, we observed that the hierarchy found analyzing representations information content is opposite to the one provided by the prediction error investigation, in the hidden space. It means that we cannot extract the true dimensionality, i.e. memory length of the process. However, if we pay more attention, it is possible to realize that, increasing the total length of the process T, a change in the hierarchies can be detected. What does this mean? The first explanation to this result consists in the fact that, instead of extracting the true generating process memory length, relevance-driven approaches can extract an effective memory length, independent from the properties of the underlying stochastic process. It seems that this effective memory length depends on the intrinsic nature of the considered dataset. It is not the only explanation for the opposite hierarchy detected. In fact, it is possible to realize that we are comparing different representations of the system state, obtained by clustering data with different observer memory lengths. In Chapter 2, we said that the jump process is Markovian in the hidden space, if and only if the representation is relevant. However, now a doubt arises: are we sure that, considering different observer memory lengths, are we still dealing with relevant representations? And that the assumption of Markovianity is still valid? It is true in the case $q' = q = 4$, where we are dealing with the causal states of the system, but for all the other values of $q' \neq 4$ we should check whether Markovian property is satisfied or not. What we can suppose, at this point, is that, since efficient representations are Markovian, too-compressed representations do not satisfy Markov property, and so they are not relevant.

By all this investigation, we can conclude that, using the concept of relevant and maximally informative representation of the system state, it seems possible

[1]Remember, as previously mentioned, that agglomerative methods are devised with euclidean metrics, but depending on the time series, this could cause a sub-optimal prediction

to find a new dimensional reduction tool based on clustering algorithms and not requiring the knowledge of the future and of the generative process of the time series. As we have just observed, starting from clustering labels, it is possible to predict the future behavior of the time series, at least in the hidden space. Using prediction error and information content analysis a hierarchy between clustering methods can be found. And by optimizing the functional 3.4, we are able to find the optimal value of the coarse-graining level. This whole procedure provides a full solution for the relevance-resolution optimization problem, [12]. However, there are still black spots in relevant representation study. One of them arose in the second experiment, where we have not been able to detect the real value of q for the ARCH process. Another problem consists in the too noisy real space prediction. The detected problems do not disprove our supposition of the existence of a connection between maximally informative and predictive representations, as shown by the first experimental results. To solve them, further investigations and better prediction algorithms are required. In future works, the first step for investigating relevant representation could consist in studying the relation between effective memory length and intrinsic properties of the dataset, like T. An interesting investigation direction consists in considering also the degree of Markovianity of the set of labels. Markovianity analysis can provide some insights about effective degrees of freedom detected by the MIR approach for time series.

In conclusion, even if this thesis work is preliminary and some problems have been detected, many strong results have been found. They helped us discover new properties of relevant representations, and confirmed that clustering methods can be used as a static and unsupervised dimensional reduction tools.

Appendix A

Causal states

In this chapter, we provide a very brief introduction to the concept of causal states described by Crutchfield and Shalizi in [1].

For introducing the causal states' idea let's consider a time series, described in the paper as a bi-infinite sequence of observables labeled by time:

$$\overleftrightarrow{Y} = \{\dots y_{-1}, y_0, y_1 \dots\} \qquad y_t \in \mathcal{Y}, t \in (-\infty, \infty)$$

where we can define the past and the future of the time series respectively as

$$\overleftarrow{Y}_t = \{\dots y_{t-2}, y_{t-1}\} \qquad \text{and} \qquad \overrightarrow{Y}_t = \{y_{t+1}, y_{t+2}, \dots\}$$

with t respectively the ending point and starting point of past and future sequences of observations. In particular, if the process is stationary, we can neglect their time dependence, i.e. we can generally refer to $\overleftarrow{Y}$ and $\overrightarrow{Y}$.

Once the notions of past and future have been introduced, we can approach the dimensionality reduction, by defining effective states, $\mathcal{R}$, the set of past leading to the same future. For defining an effective state we can define a function

$$\eta : \overleftarrow{Y} \to \mathcal{R}$$

From this definition, it is clear that all the histories belonging to the same effective state are equivalent in future predictions. A graphical representation of effective states is shown in Fig.A.1.

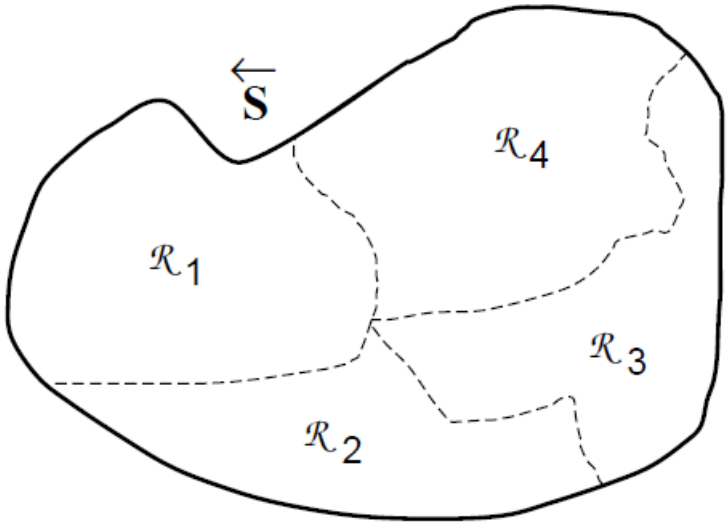

Figure A.1: Effective states graphical representation

However, the definition of effective states is too general, there are several choices for them. In Crutchfield and Shalizi's paper, the optimal choice of effective states are causal states, $\mathcal{S}$, which can be constructed starting from the following function

$$\epsilon\left(\overleftarrow{Y}\right) = \left\{\overleftarrow{Y}' | P\left(\overrightarrow{Y} | \overleftarrow{Y}'\right) = P\left(\overrightarrow{Y} | \overleftarrow{Y}\right), \forall \overrightarrow{Y} \text{ and } \forall \overleftarrow{Y}\right\}$$

They represent an equivalence class. A graphical representation of causal states is shown in Fig.A.2.

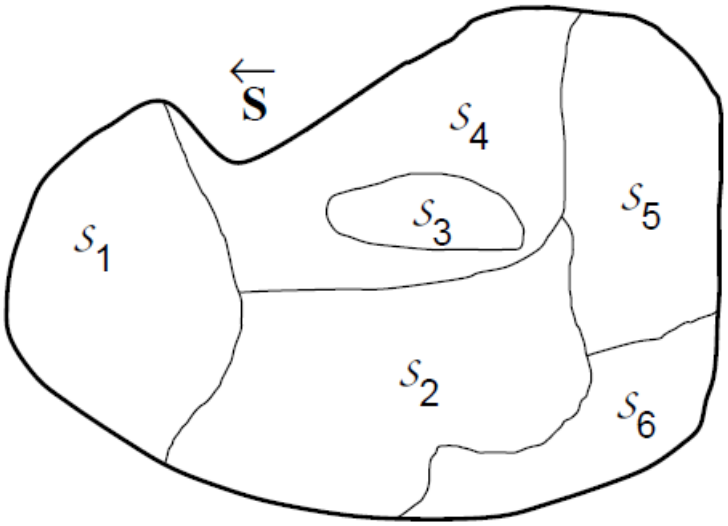

Figure A.2: Causal states graphical representation

The causal states just describe have important quantities:

- Maximal prescience: Given any set of effective states, we have that

$$H\left[\overrightarrow{Y}|\mathcal{R}\right] \geq H\left[\overrightarrow{Y}|\mathcal{S}\right];$$

- They are a sufficient statistic for approaching future prediction:
- They are the set of states, characterized by the minimal complexity, to every other kind of collection of pasts;
- They are unique.

Starting from the concept of causal states, Crutchfield and Shalizi defined dynamics using the concept of ϵ-machines, however, this goes further to our research interest. For having a deeper understanding of ϵ-machines construction, see [1]. Another way to reconstruct causal states and $\epsilon-$machines given a time series is proposed in [15].

Appendix B

Information content and prediction for clustering algorithms

In this section, we show, for each inter-cluster linkage the plots of relevance, resolution, and total information as a function of the number of clusters. They are reported in the following figures, starting from the case of the ARCH model time series.

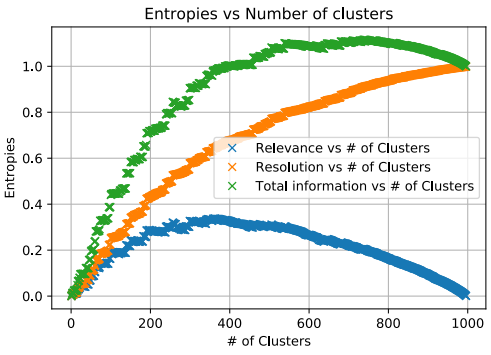

Figure B.1: Relevance vs k, Average Linkage, ARCH model

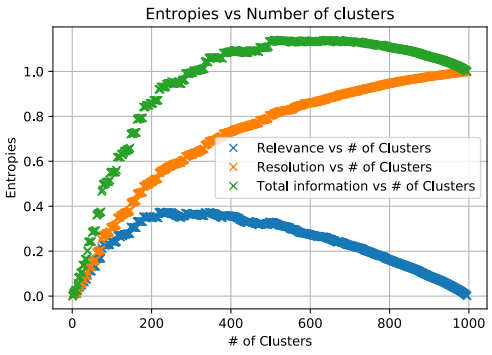

Figure B.2: Relevance vs k, Complete Linkage,ARCH model

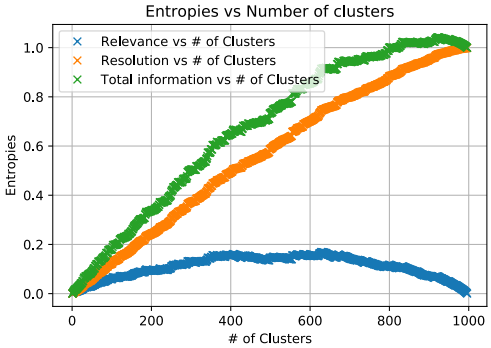

Figure B.3: Relevance vs k, Single Linkage, ARCH model

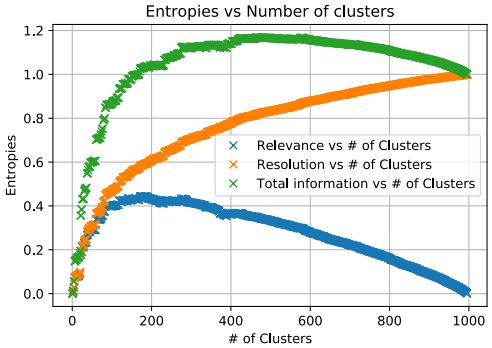

Figure B.4: Relevance vs k, Ward Linkage,ARCH model

The same results for the case of the S&P 500 time series are the following

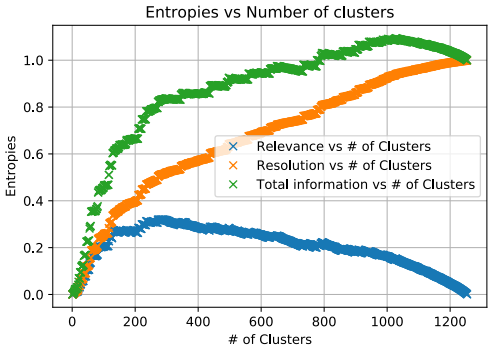

Figure B.5: Relevance vs k, Average Linkage, S&P 500

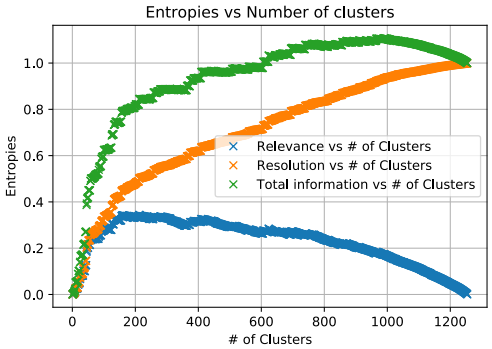

Figure B.6: Relevance vs k, Complete Linkage, S&P 500

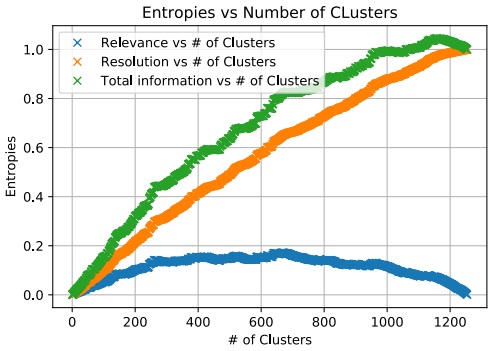

Figure B.7: Relevance vs k, Single Linkage, S&P 500

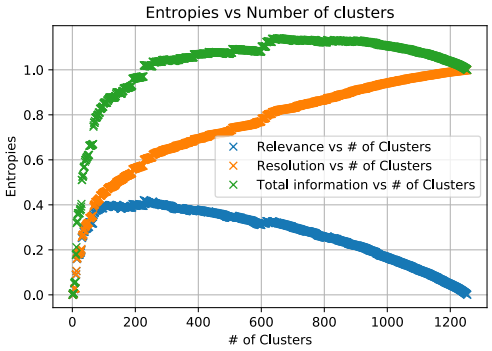

Figure B.8: Relevance vs k, Ward Linkage, S&P 500

Bibliography

[1] J.P. Cruchfield and C. R. Shalizi. «Computational Mechanics: Pattern and Prediction, Structure and Simplicity». In: *J. Stat. Mech.* 104 (June 2001), p. 816 (cit. on pp. 1, 6, 10, 11, 33, 50, 52).

[2] N. Brodu. «Reconstruction of Epsilon-Machines in Predictive Frameworks and Decisional States». In: *Advances in complex systems* 14 (Feb. 2011), p. 761 (cit. on p. 2).

[3] C. R. Shalizi and K. L. Shalizi. «Blind Construction of Optimal Nonlinear Recursive Predictors for Discrete Sequences». In: *UAI-P* 1 (Aug. 2004), p. 504 (cit. on p. 2).

[4] F. C. Pereira N. Tishby and W. Bialek. «The information bottleneck method». In: *arXiv* 1 (Apr. 2000), p. 1 (cit. on pp. 2, 10, 11, 40).

[5] I. Mastromatteo M. Marsili and Y. Roudi. «On sampling and modeling complex systems». In: *J. Stat. Mech.* 20 (Nov. 2013), P09003 (cit. on pp. 2, 12, 14).

[6] J. Jo J. Song M. Marsili. «Resolution and relevance trade-offs in deep learning». In: *Journal of statistical mechanics: theory and experiments* 12 (Mar. 2018), p. 123406 (cit. on pp. 2, 41).

[7] R. J. Cubero M. Marsili and Y. Roudi. «Minimum Description Length Codes Are Critical». In: *MDPI* 10 (Oct. 2018), p. 755 (cit. on pp. 12, 32).

[8] A. Haimovici and M. Marsili. «Criticality of mostly informative samples: A bayesian model selection approach». In: *J. Stat. Mech. Theory Exp.* 10 (Apr. 2015), P10013 (cit. on p. 14).

[9] J. P. Bouchaud and M. Potters. *Theory of financial risk and derivative pricing.* Cambridge, England: Cambridge university press, 2003 (cit. on p. 16).

[10] P. Mehta; M. Bukov; C. H. Wanga; A. G. R. Day; C. Richardson; C. K. Fisher and D. J. Schwab. «A high bias, low variance introduction to Machine Learning for physicists». In: *Physics reports* 810 (Mar. 2019), p. 1 (cit. on p. 21).

[11] H. Jeffreys. «An invariant form for the prior probability in estimation problems». In: *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences.* 186 (1946), p. 451 (cit. on p. 27).

[12] I. Nemenman N. Tishby and W. Bialek. «Predictability, Complexity, and Learning». In: *Massachusetts Institute of Technology* 13 (Mar. 2001), p. 2406 (cit. on pp. 40, 49).

[13] N. Tishby and N Slonim. «Agglomerative Information Bottleneck». In: *NIPS'99¦ Proceedings of the 12th International Conference on Neural Information Processing Systems* 1 (Apr. 2000), p. 618 (cit. on p. 41).

[14] B. C. Ross. «Mutual Information between Discrete and Continuous Data Sets». In: *PLOS* 2 (Apr. 2014), p. 9 (cit. on p. 41).

[15] J. Runge. «Causal network reconstruction from time series: From theoretical assumptions to practical estimation». In: *Chaos* 075307 (May 2018), p. 28 (cit. on p. 52).