Cost-Sensitive Early classification of Time Series

Asma Dachraoui

NNT : 2017SACLA002

Thèse de doctorat de l'Université Paris-Saclay preparée à AgroParisTech

Ecole doctorale n°581
Agriculture, alimentation, BIologie, Environnement, Santé (ABIES)
Spécialité de doctorat: Informatique appliquée
par

Mme. Asma Dachraoui

Cost-Sensitive Early Classification of Time Series

Thèse présentée et soutenue à Paris, le 31 Janvier 2017.

Composition du Jury :

M.	Antoine Cornuéjols	Pr. AgroParisTech	Directeur
M.	Alexis Bondu	Ing. chercheur, EDF Lab, Saclay	Co-Directeur
M.	Jian Pei	Pr. Université Simon Fraser, Canada	Rapporteur
M.	Gilles Stoltz	Chargé de recherche, CNRS-HEC, Paris	Rapporteur
M.	Thomas Guyet	Pr. assistant, AgroCampus-Ouest, Rennes	Examinateur
M.	Jean-Christophe Janodet	Pr. IBISC Lab, Université d'Evry	Président du jury
M.	Fabrice Clérot	Ing. chercheur, Orange Labs, Lannion	Invité

université PARIS-SACLAY
ÉCOLE DOCTORALE
Agriculture, alimentation, biologie, environnement, santé (ABIES)

ABSTRACT

Title : Cost-Sensitive Early Classification of Time Series

Keywords : Early classification, time series, online decision making, adaptive and non-myopic decisions, costly delaying decision.

Early classification of time series is becoming increasingly a valuable task for assisting in decision making process in many application domains. In this setting, information can be gained by waiting for more evidences to arrive, thus helping to make better decisions that incur lower misclassification costs, but, meanwhile, the cost associated with delaying the decision generally increases, rendering the decision less attractive. Making early predictions provided that are accurate requires then to solve an optimization problem combining two types of competing costs.

This thesis introduces a new general framework for time series early classification problem. Unlike classical approaches that implicitly assume that misclassification errors are cost equally and the cost of delaying the decision is constant over time, we cast the the problem as a cost-sensitive online decision making problem when delaying the decision is costly. We then propose a new formal criterion that expresses the trade-off between the gain of information that is expected to incur lower misclassification costs when delaying the decision against the cost of such a delay.

On top of this generic formulation, we propose two different approaches that estimate the optimal decision time for a new incoming yet incomplete time series. In particular, the first approach (i) captures the evolutions of typical complete time series in the training set thanks to a clustering technique that forms meaningful groups, and (ii) leverages

these complete information to estimate the costs for all future time steps where data points still missing. This allows one to forecast what should be the optimal horizon for the classification of the incoming time series. The second approach performs also steps (i) and (ii), but instead of using a clustering technique, it uses a more informed segmentation method that exploits the class labels of the complete time series thanks to the confidence levels computed by a probabilistic classifier.

These approaches are interesting in two ways. First, they estimate, online, the earliest time in the future where a minimization of the criterion can be expected. They thus go beyond the classical approaches that myopically decide at each time step whether to make a decision or to postpone the call one more time step. Second, they are adaptive, in that the properties of the incoming time series are taken into account to decide when is the optimal time to output a prediction.

We conduct extensive experimental studies and make systematic comparisons between both approaches on synthetic and real data sets. The obtained results show that both approaches meet the behaviors expected from early classification systems (i.e. the easier the classification task, the earlier the decision), with a significant superiority of the second approach when the classification of the incomplete time series is difficult.

Acknowledgments

First and foremost, I want to thank my advisor Prof. Antoine Cornuéjols for his support, his patience, his assistance in writing reports, and his immense knowledge and several skills. His valuable guidance helped me to better conduct research, find solutions and write this thesis.

My sincere thanks also go to my advisor Dr. Alexis Bondu, who taught me how to do research and always gave me suggestions when I met problems. It was through his enthusiasm, persistence, understanding and kindness that I completed my masters internship and was encouraged to apply for this thesis.

Thank you Antoine and Alexis for advising my work and for all the research discussions we had and that have always been fruitful and rewarding. I enjoyed working with you and I hope we have other opportunities to work together again.

Besides my advisors, I want to thank my thesis supervisors, Prof. Ahlame Douzal and Engr. Fabrice Clérot for their insightful comments.

I also want to thank my thesis readers, Prof. Jian Pei and Prof. Gilles Stoltz, who kindly accepted to review my work.

I am also grateful to Prof. Thomas Guyet and Prof. Jean-Christophe Janodet who kindly accepted to evaluate my thesis.

This success also belongs to my family as much as it does to me. Both my children, Sandra and Yassine, were born during my PhD study. They have illuminated my life and continue to do so every day. Thank you Sandra and Yassine for being a source of inspiration and comfort during difficult moments. While I regret the time I spent away from you while working towards this degree. Your are my everything.

I also want to thank my parents for the support and care they provided me through my entire life. Thank you Mom, Hedia, for your unconditional love, support and confi-

dence of my abilities even when I wasn't. Thank you Dad, Moncef, for believing in me and I hope I made you proud of me as much as I am proud of you. My dearest sister, Ines, thank you for always being my best friend and always being by my side, for your encouragement and unconditional support. Thank you Achref and Anwer, for being the best brothers anyone could ever hope for. I love you all.

Last but not the least, I want to thank my dearest husband and best friend Baligh. Thank you for your encouragement and love. Thank you for being so understanding and supporting me through the toughest moments of my life. I could never have done this without you. I love you.

Contents

List of Figures

List of Tables

Notations

Symbol	Meaning
$\langle x_1, x_2, \ldots, x_T \rangle$	A sequence of T elements.
$\mathbf{x}_T \in \mathbb{R}^T$	A time series of length T
$\mathbf{x}_t \in \mathbb{R}^t$	An incoming time series of length $t \leq T$
$y \in \mathcal{Y}$	A class label.
$\hat{y}$	A predicted class label.
$\hat{y}^t$	A predicted class label at time t.
$(\mathbf{x}_T^i, y^i) \in mathcal S$	The t^{th} example in the training set $\mathcal{S}$, composed of a time series $\mathbf{x}_T^i$ and its associated class label y^i.
$\mathcal{S} = \{(\mathbf{x}_T^i, y^i)\}_{1 \leq i \leq m}$	A training set of labeled time series of length T.
$h : \underset{\mathbf{x}_T \mapsto \hat{y} = h(\mathbf{x}_T)}{\mathbb{R}^T \to \mathcal{Y}}$	A classifier.
$\mathcal{H} = \{h_t\}_{1 \leq t \leq T}$	A set of classifiers learnt at each time t.
$t^\star$	Optimal time for classifying $\mathbf{x}_t$.
$\mathrm{C}(\hat{y}\|y)$	Misclassification cost function.
$\mathrm{C}(t)$	Delaying cost function.
$f_\tau(\mathbf{x}_t) : \mathbb{R}^t \longmapsto \mathbb{R}$	Expected cost function given $\mathbf{x}_t$ for a future time $\tau \in \{0, \ldots, T-t\}$
$\mathcal{C} = \{\mathfrak{c}_k\}_{1 \leq k \leq K}$	A set of K clusters of time series.
m_{uv}^t	A transition probability between states u and v at time step t.
$\gamma_t = \ell$	The Markov state at time step t is equal to ℓ.
$\langle \gamma_1, \ldots, \gamma_t \rangle$	A sequence of states recoding a time series $\mathbf{x}_t$.
$\overrightarrow{\gamma_t}$	A probability vector.
$\mathbf{M}_t^{t+1}$	A transition matrix from time t to time $t+1$.
$\mathbf{M}_{1,\ldots,t}^t$	A transition matrix from a coded sequence of length t.
$\mathbf{P}_{\langle \gamma_1, \ldots, \gamma_t \rangle}$	A probability vector of observed states.
δ	Markov chain order.

Abbreviations

Abbrev	Meaning
Abbrev	Abbreviation
ECONOMY	**E**arly **C**lassification for **O**ptimized and **NO**n-**MY**opic online decision making.
ECONOMY-K	Extended ECONOMY formulation based on a clustering technique. K is the number of clusters.
ECONOMY-γ	Extended ECONOMY formulation based on Markov chains. γ is a Markov chain state.
KNN	The K-Nearest Neighbors approach.
DFT	Discrete Fourier Transform.
$\overline{\tau}^{\star}_{ETM}$	Estimated decision time mean.
$\overline{\mathrm{C}}_{ECM}$	Estimated cost mean.

The reality might be hard or even impossible to model (and understand) accurately, but there still can be ways to make the good decisions.

Vladimir Vapnik

Chapter 1

Introduction

1.1 On the need to make cost-sensitive early predictions

There exists nowadays an increasing awareness of the importance of learning in support of online decision making. In emergency wards of hospitals, in control rooms of national or international electrical power grids, in government councils assessing emergency situations, in all kinds of contexts, it is essential to make timely decisions in absence of complete knowledge of the true outcome. The issue facing the decision makers is that, usually, the longer the decision is delayed, the clearer is the likely outcome (e.g. should the patient undergo a risky surgical operation), but, also, the higher the cost that will be incurred if only because earlier decisions allow one to be better prepared.

This is a classical optimization problem with a trade-off between the gain of information that can incur lower misclassification costs if one delays the decision, and the rising cost of such a delay. It has historical roots in fields such as sequential decision making, optimal statistical decisions, cost-sensitive learning, etc. but recent technological advancements that have led to a huge amount of data (generated often in fine-grained manner leading to temporal data) and numerous new applications coined in distinctive domains (e.g. medicine, automatic transportation, electric smart grids, internet and financial systems and so on) give a new impetus to research works in this area.

Recently, making early predictions on time series has attracted a considerable attention since it comes as a straightforward answer for making online decisions based on incoming time series with data points still missing. This is a very common task in many areas. For example, earlier diagnosis based on abnormal heart beat of preterm infants

who are developing sepsis during their hospitalization in neonatal intensive care units may help to initiate treatment before the clinical symptoms appear [54]. On the basis of heart beat time series data, the ability to achieve classifications as early as possible can permit early diagnosis and thus preventative and adjunctive therapies.

In this situation and many others, it is desirable, and, often, essential to achieve early predictions. However, this commonly entails two contradictory objectives: the earliness and the quality of predictions. Indeed, the earliest the decision, the more rewarding it can be. Yet, often, gathering more information allows one to get a better decision. This is a trade-off between time vs quality of predictions that must be optimized and generally solved online.

In the growing literature on early classification of time series problem, several works that are openly motivated by making early predictions turn out to be concerned with the problem of classifying from incomplete time series, rather than with the problem of optimizing a trade-off between the quality of the prediction and the time it is performed. And even if the earliness of the decision is mentioned as a motivation in these works, the decision procedures themselves do not take it explicitly into account. They instead evaluate the confidence or the reliability of the current prediction in order to decide if the time is ripe for prediction, or if it seems better to wait one more data point. In addition, the procedures are myopic in that they do not look further than the current time to decide if it a prediction should be made.

As it is defined in the literature, the problem of early classification of time series was not placed in its general context where it is crucial to explicitly take into account the cost of delaying the decision in the optimization criterion. We thus argue that, as soon as the earliness is involved within the decision making process, the cost of delaying the decision seems to be, intuitively, a crucial factor that should not be ignored and should be explicitly accounted for, along with the misclassification cost, in the decision optimization procedure (Turney provides in [98] a complete survey on different types of cost and emphasizes the crucial role that they play in real-world applications).

To propose a general framework for the early classification of time series problem and to solve an optimization problem combining two competing costs are our main contributions in this thesis.

1.2 Challenges and main objectives of this thesis

The problem of making early classification is challenging for many reasons. Two substantial requirements for achieving early decisions are the quality and the earliness of predictions. These requirements are contradictory: the earliest the decision, usually, the more erroneous the prediction it can be. To address both of these challenges, an early classification system should solve different kinds of tasks:

- First, an early classification system should be able to label incomplete time series, and broadly time series of different lengths.
- Second, an early classification system should optimize the earliness vs quality of the prediction trade-off.
- Finally, an early classification system should be able to make online decisions.

In addition to these challenging tasks for making early classification, we consider the task of explicitly taking into account two different types of costs when making a prediction: (i) the cost incurred by misclassifying time series, and (ii) the cost incurred by delaying the decision before making a prediction. To the best of our knowledge, we are the first to explicitly consider the costs of delaying the decision when making early classifications. Until our work [33], state-of-the-art early classification approaches implicitly assume that all misclassification errors are cost equally and the cost of delaying the decision is constant over time.

The design goals we wish to achieve when making cost-sensitive decisions include the following requirements:

1. **Take into account the misclassification cost.**
2. **Take into account the delaying cost of decision.**
3. **Optimize the time vs quality of prediction trade-off:** the system should involve both contradictory requirements, the earliness and the quality, in the optimization criterion.
4. **Ability to decide online:** the system should be able to decide, online, when to stop considering additional information and output a prediction.
5. **Ability to make non-myopic decisions:** the system should give an estimate of when the optimal prediction time is likely to occur.

6. **Ability to adapt the prediction to the incoming time series:** the output prediction should depend on the individual input time series, i.e. the total cost is re-estimated with each new arriving data point.

7. **Few parameters to set.**

1.3 Contributions

In this present work, we are concerned with the problem of early classification of time series when delaying the decision is costly. First of all, we started by defining the problem and dividing it into two independent tasks: (i) labeling incomplete time series, and (ii) estimating, online, the optimal decision time. Then, in order to explicitly take into account the misclassification cost and the cost of delaying the decision, we cast the problem to a cost-sensitive online decision making problem, and proposed a new optimization criterion, along with two approaches that satisfy the above mentioned objectives.

As major contributions of this thesis, we particularly cite the following:

1. An early classification of time series framework has been proposed in which early classification systems are explicitly endowed with a decision function that decides when to stop considering additional information and make a prediction.

2. We proposed **ECONOMY** (**E**arly **C**lassification for **O**ptimized and **NO**n-**MY**opic online decision making), a new generic[1] optimization criterion that explicitly takes into account the cost of misclassification and the cost of delaying the decision.

3. We developed two different approaches, ECONOMY-K and ECONOMY-γ, that implement the generic formulation, ECONOMY, using two different segmentation methods. The objective behind segmenting the training set is to build meaningful groups, that differ as widely as possible, in order to capture typical evolutions of the training time series. This will be useful to estimate, for an incoming yet incomplete time series, the costs for future time steps where the corresponding data points are still missing. Specifically:

    - The segmentation in ECONOMY-K is performed by using a clustering technique over the training set. The result is a set of meaningful groups of time series built according to a specific similarity policy.

[1] By generic, we mean that the optimization criterion does not depend on a specific type of classifier, any classifier can be used.

- The segmentation in ECONOMY-γ is achieved based on special type of Markov chains. The objective is to segment time series while tacking into account information about their class label.

In addition of satisfying the objectives we mentioned above, both methods offer interesting properties as will be detailed later in this work.

4. We performed extensive experiments on real data sets and synthetic data sets for which we described the generation procedure for the sake of reproducibility. Then, we presented and interpreted most important results.

1.4 Organization of the thesis

This thesis is organized as follows.

Chapter 2 presents the background of conventional machine learning and time series classification problems and introduces the challenges of making early classifications. We propose a generic framework of early classification that will provide the basis to review the state-of-the-art methods.

Chapter 3 suggests a categorization of the strategies used for instantiating early classifiers and describes their techniques for handling incomplete time series for the purpose of making classification. One of the proposed strategies will be used to instantiate early classifiers in our experimental study.

Chapter 4 examines the state-of-the-art of early classification methods according to the general framework of early classification of time series that we proposed in Chapter 2.

Chapter 5 starts with a formal analysis of the quality against the earliness of prediction trade-off while introducing the notion of costs. This analysis provides a base for a new cost-sensitive online decision approach, called ECONOMY (Early Classification for Optimized and NOn-MYopic online decision making), that trades off the gain of information that is expected to incur lower misclassification costs when delaying the decision against the cost of such a delay.

In **Chapter** 6, we conduct extensive experimental studies on synthetic and real data sets, and show that both approaches, ECONOMY-K and ECONOMY-γ, that imple-

ment the generic optimization criterion ECONOMY using two different segmentation techniques, meet the behaviors expected from early classification systems.

In **Chapter** 7, as conclusion, we summarize our main contributions and discuss possible directions for future works.

Chapter 2

Background

Introduction

This chapter consists of two parts. In the first, we provide some basic notions on supervised learning with a focus on classification problems. This allows us to introduce our notations and draw a direction of this thesis. We then discuss the challenges of the classification on time series and give a brief synthesis on representations and similarity measures adapted to time series. In the second part, we introduce the problem of early classification on time series. We define the problem and make a series of comparisons with closely related problems as offline, online and anytime classification problems. We finally propose a generic framework that is able to describe early classification methods. This exposes the peculiarities of our setting and serves as a basis for our contributions throughout this thesis.

2.1 Supervised classification of time series

In our thesis, we study the classification of time series. Time series can be seen as vectors belonging to $\mathbb{R}^t$ where $t \in \{1, \ldots, T\}$. Because T can be very large, it does not seem appropriate to use generative models [75] to perform classification, since these models are prone to the curse of dimensionality [11]. This is why in this thesis, we have considered mainly discriminative models [75] for the classification task since the dimension of the input space is generally not a parameter that impacts the theoretical guarantees on learning and generalization. The following sections aim at giving but a flavor of basic concepts on supervised discriminative classification.

2.2 Some basic notions on supervised classification

Supervised learning aims at learning a function that maps between variables in an *input space* $\mathcal{X}$ and a variable in an *output space* $\mathcal{Y}$ and applying this mapping function to predict the outputs of unseen data. Depending on the type of $\mathcal{Y}$, two different learning problems are distinguished. *Classification* problems try to map input data to a *qualitative* label (or a class), in this case $\mathcal{Y}$ is a finite and discrete space (e.g. colors, correct/incorrect, alphabets, etc.), while *Regression* problems try to learn a *quantity* in $\mathcal{Y}$ which is a continuous space (e.g. stock market price, weight, etc.). In this thesis, we will focus on classification problems and, more precisely, binary classification where $\mathcal{Y} = \{-1, +1\}$.

2.2.1 Training set and classes

Let $\mathcal{D}$ be an unknown distribution probability on $\mathcal{X} \times \mathcal{Y}$. A training sample is a finite set of examples, where, for each element $\mathbf{x} \in \mathcal{X}$ is assigned a target value $y \in \mathcal{Y}$. These examples are drawn (usually identically and independently distributed) according to the distribution $\mathcal{D}$. Therefore, only a partial knowledge about $\mathcal{D}$ is given by this training set. We note $\mathcal{S} = \{(\mathbf{x}^1, y^1) \ldots, (\mathbf{x}^m, y^m)\}$ a training sample of m data, with $\mathbf{x}^i$ is the description of the example i and y^i is its class label.
From the training sample $\mathcal{S}$, the objective of learning is to infer a function $h : \mathcal{X} \longrightarrow \mathcal{Y}$ that maps $\mathcal{X}$ to $\mathcal{Y}$, i.e. a function that best describes the relation between inputs and classes (in presence of partial knowledge). The *true* relation, which is unique for each problem, is the *target function* to be inferred since it is generally unknown. The function h is then one of the possible functions in some space, usually called *hypothesis space*, that try to approach the *target function* and the crux challenge of learning is to find the best hypothesis. We note the hypothesis space $\mathcal{H} \in \mathcal{Y}^{\mathcal{X}}$, where $\mathcal{Y}^{\mathcal{X}}$ represents the set of all possible functions that map $\mathcal{X}$ to $\mathcal{Y}$.
Different hypothesis classes have been proposed for different machine learning problems. In classification problems, the class of *linear classifiers* has almost the most popular methods used in machine learning. A classifier is linear if its decision boundary on $\mathcal{X}$ (that is usually $\mathbb{R}^t$, where $t \in \{1, \ldots, T\}$) is a linear function. In case of binary linear classifiers, positive and negative examples are separated by a *hyperplane* which is a generalization of a *line* to dimensions larger than 2 (a plane is of dimensions 3).
A hyperplane in $\mathcal{X}$ is a sub-space of $\mathcal{X}$ that can be described by a linear equation of the form:

$$\langle \mathbf{w}, \mathbf{x} \rangle + b = 0 \tag{2.1}$$

where $\langle \mathbf{w}, \mathbf{x} \rangle = \sum_{i=1}^{t} w^i \mathbf{x}^i$, $\mathbf{w} = (w_1, \ldots, w_t) \in \mathcal{X}$ is a normal vector of the hyperplane that can be seen as the vector of coefficients weighting each variable in $\mathcal{X}$ (each dimension corresponds to one variable), and $b \in \mathbb{R}$ is a bias. If $b = 0$, the hyperplane passes through the origin.

Definition 2.2.1 (Linear classifier) *A linear classifier of normal vector* $\mathbf{w} \in \mathcal{X}$ *and a bias* b *is a function* $h_{\mathbf{w},b} \in \mathcal{Y}^{\mathcal{X}}$ *such that for any example* $\mathbf{x} \in \mathcal{X}$*, it is defined as:*

$$h_{\mathbf{w},b}(\mathbf{x}) = sign(\langle \mathbf{w}, \mathbf{x} \rangle + b) = \begin{cases} +1 & if \quad \langle \mathbf{w}, \mathbf{x} \rangle + b > 0 \\ -1 & if \quad \langle \mathbf{w}, \mathbf{x} \rangle + b < 0 \end{cases} \tag{2.2}$$

and $sign(\langle \mathbf{w}, \mathbf{x} \rangle + b = 0)$ *is arbitrarily set to* $+1$*.*

Although linear classifiers are simple to implement, the difficulty is in determining the parameters $\mathbf{w}$ and b based on the training set with the objective of a good generalization on new data.

Another important concept in linear classification is the *margin*. Given $\mathbf{x} \in \mathcal{X}$ and a function $h_{\mathbf{w},b} \in \mathcal{H}$, it is possible to measure the confidence of predicting the class $\hat{y}$, where $\hat{y} = h_{\mathbf{w},b}(\mathbf{x})$ of $\mathbf{x}$ through computing its margin. The margin of $\mathbf{x}$ is defined as its distance from the hyperplane that is represented by $(\mathbf{w}, b)$.

Definition 2.2.2 (Margin of an example) *Given a function* $h_{\mathbf{w},b} \in \mathcal{H}$ *and an example* $\mathbf{x}^i \in \mathcal{X}$*, we define the margin* $m^{\mathbf{x}^i}_{h_{\mathbf{w},b}}$ *of* $h_{\mathbf{w},b}$ *on* $\mathbf{x}^i$ *as:*

$$m^{\mathbf{x}^i}_{\mathbf{w},b} = \frac{\mathbf{w}\mathbf{x}^i + b}{\| \mathbf{w} \|} \tag{2.3}$$

When $y^i m^{\mathbf{x}^i}_{\mathbf{w},b} < 0$, then, $\mathbf{x}^i$ is misclassified, while $y^i m^{\mathbf{x}^i}_{\mathbf{w},b} \geq 0$ indicates that $\mathbf{x}^i$ is correctly classified and $m^{\mathbf{x}^i}_{\mathbf{w},b}$ represents the confidence of the class prediction, i.e. the largest the margin, the more confident the prediction. We can now define the margin on the training set $\mathcal{S}$.

Definition 2.2.3 (Margin of a set) *Given a training set* $\mathcal{S}$ *and a linear classifier function* $h_{\mathbf{w},b} \in \mathcal{H}$*, we define the margin* $m^{\mathcal{S}}_{\mathbf{w},b}$ *of* $h_{\mathbf{w},b}$ *on* $\mathcal{S}$ *as:*

$$m^{\mathcal{S}}_{\mathbf{w},b} = \min_{\mathbf{x}^i \in \mathcal{S}} m^{\mathbf{x}^i}_{\mathbf{w},b} \tag{2.4}$$

2.2.2 Loss function and risks

The objective of learning in classification is to find the function h^* (for simplicity we note h^* instead of $h_{\mathbf{w}^*,b^*}$) that best fits the training sample $\mathcal{S}$. We may first start by

introducing the concept of a *loss* function $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_+$ that measures how good a function h is on a particular training example $(\mathbf{x}^i, y^i)$.

Definition 2.2.4 (Loss function) *Given an example $(\mathbf{x}^i, y^i)$ and a function $h_{\mathbf{w},b}$, the loss of predicting the class of $\mathbf{x}^i$ is:*

$$\ell(h_{\mathbf{w},b}(\mathbf{x}^i), y^i) \tag{2.5}$$

For binary classification, the most popular losses are:

$$\begin{aligned}
\textbf{0-1 loss:} \quad & \ell_{0-1}(h_{\mathbf{w},b}(\mathbf{x}^i), y^i) = \mathbb{I}[h_{\mathbf{w},b}(\mathbf{x}^i) \neq y^i] \\
\textbf{squared loss:} \quad & \ell_2(h_{\mathbf{w},b}(\mathbf{x}^i), y^i) = (h_{\mathbf{w},b}(\mathbf{x}^i) - y^i)^2 \\
\textbf{hinge loss:} \quad & \ell_{hinge}(h_{\mathbf{w},b}(\mathbf{x}^i), y^i) = \max(0, 1 - y^i h_{\mathbf{w},b}(\mathbf{x}^i)) \\
\textbf{log loss:} \quad & \ell_{log}(h_{\mathbf{w},b}(\mathbf{x}^i), y^i) = log(1 + \exp^{-h_{\mathbf{w},b}(\mathbf{x}^i) y^i}) \\
\textbf{exponential loss:} \quad & \ell_{exp}(h_{\mathbf{w},b}(\mathbf{x}^i), y^i) = \exp^{-h_{\mathbf{w},b}(\mathbf{x}^i) y^i}
\end{aligned} \tag{2.6}$$

Now, let us generalize the concept of *loss* to examples from the distribution $\mathcal{D}$. Such generalization is called (true) *risk* since it measures the average loss of the function $h_{\mathbf{w},b}$ on $\mathcal{D}$.

Definition 2.2.5 ((True) risk) *Given a function $h_{\mathbf{w},b} \in \mathcal{H}$ and a distribution $\mathcal{D}$, the (true) risk is defined as the expected loss of $h_{\mathbf{w},b}$ on $\mathcal{D}$:*

$$R^{\ell}_{\mathcal{D}}(h) = \mathbb{E}_{\mathcal{D}}[\ell(h(\mathbf{x}), y)] \tag{2.7}$$

The goal would be to minimize $R^{\ell}_{\mathcal{D}}$, however, since $\mathcal{D}$ is unknown, the true risk can not be calculated but can be estimated from a finite sample of data drawn from $\mathcal{D}$ such as $\mathcal{S} = \{(\mathbf{x}^1, y^1), \ldots, (\mathbf{x}^m, y^m)\}$:

$$\mathbb{E}_{\mathcal{D}}[\ell(h(\mathbf{x}), y)] \approx \hat{\mathbb{E}}_{\mathcal{S}}[\ell(h(\mathbf{x}), y)] \tag{2.8}$$

$\hat{\mathbb{E}}_{\mathcal{S}}[\ell(h(\mathbf{x}), y)]$ is called empirical risk of $h_{\mathbf{w},b}$ on $\mathcal{S}$ and is defined as:

$$R^{\ell}_{\mathcal{S}}(h) = \frac{1}{m} \sum_{\mathcal{S}} \ell(h(\mathbf{x}^i), y^i) \tag{2.9}$$

Finding the function h^* that minimizes $R^{\ell}_{\mathcal{S}}$ is known as empirical risk minimization (ERM),

$$h^* = \underset{h \in \mathcal{H}}{\operatorname{ArgMin}}\, R^{\ell}_{\mathcal{S}} \tag{2.10}$$

and the hope is to minimize the true risk trough the minimization of the empirical risk. Without going into detail, in practice and for a fixed, convenient, loss function ℓ, the approximation of $R^{\ell}_{\mathcal{D}}$ mainly depends on (i) the size of the training sample, as it is expected that $R^{\ell}_{\mathcal{D}} \longrightarrow R^{\ell}_{\mathcal{S}}$ if more data are available, and (ii) the class of functions $\mathcal{H}$. Short of providing a comprehensive overview of the statistical learning theory that aims at providing bounds on the difference between the true risk and the empirical risk associated with any function $h_{\mathbf{w},b} \in \mathcal{H}$, we now provide some results for the Perceptron learning rule.

2.2.3 The Perceptron algorithm

Inspired by the functioning of biological neurons, the Perceptron, introduced by Rosenblatt [85], is one of the most popular machine learning methods. The Perceptron is a linear classifier composed only of one neuron. It can be extended to nonlinear classification using more neurons organized in many layers. This is called Multilayer Perceptron [88], a particular type of neural networks.

The Perceptron is an iterative algorithm (see Algorithm 1) that aims at learning a linear function in a simple way. Given a sample $\mathcal{S} = \{(\mathbf{x}^1, y^1), \ldots, (\mathbf{x}^m, y^m)\}$, the Perceptron algorithm considers one example at a time and updates the weight vector $\mathbf{w}$ and the bias b when an example is misclassified. The interesting property of the Perceptron is its

Algorithm 1 Perceptron algorithm

Input:

- A sample $\mathcal{S} = \{(\mathbf{x}^1, y^1), \ldots, (\mathbf{x}^m, y^m)\}$

Output:

- A hyperplane $(\mathbf{w}, b)$

$\mathbf{w}^0 \leftarrow 0$, $b \leftarrow 0$
for all $\mathbf{x}^i : i \in [1, N]$ **do**
  if $(y^i \neq h_{\mathbf{w}^k, b^k}(\mathbf{x}^i))$ **then**
    $\mathbf{w}^{k+1} \leftarrow \mathbf{w}^k + y^i \mathbf{x}^i$
    $b^{k+1} \leftarrow b^k + y^i$
    $k \leftarrow k + 1$
  end if
end for
return $(\mathbf{w}^k, b^k)$

guarantee to converge in a finite number of steps if the problem is linearly separable.

Theorem 2.2.6 (Convergence of Perceptron [15, 76]) *If the training data* $\mathcal{S} \in \mathcal{X} \times \mathcal{Y}$ *is linearly separable, with a margin* $m^{\mathcal{S}}_{\mathbf{w}^*,b^*}$*, by a hyperplane* $(\mathbf{w}^*, b^*)$ *of norm* $\| \mathbf{w}^* \|$*, then the Perceptron algorithm converges after*

$$\frac{R^2}{m^{\mathcal{S}}_{\mathbf{w}^*,b^*}} \tag{2.11}$$

updates during learning, assuming $\| \mathbf{x}^i \| < R$, $i \in \{1, \dots m\}$ *(*$R = \max_i(\| \mathbf{x}^i \|)$*)*

From Theorem 2.2.6, the remarkable property of the Perceptron is that it convergences after only a finite number of steps independently of the size of the training set m, on the underlying distribution $\mathcal{D}$, and almost independently of the dimension of the input space (the dependence is indirect through R).

2.3 Time series classification

Time series classification has raised great interest over the last few decades not only within the data mining and machine learning communities but also within numerous practical fields such as medicine [43, 57, 77, 82], geology [63], astronomy [83], telecommunication, meteorology, energy, financial market, etc., where, in almost every application, data are measured over time leading to a vast amount of temporal data [2, 71].

Numeric time series are a particular type of temporal data with real values ordered in time. Often, these values, referred as data points or measurements, are correlated over time. This makes time series distinctive from typical data, used commonly by conventional machine learning algorithms, where each value is measured by an independent variable. Therefore, conventional classification methods can not directly support the specific properties of time series. For these reasons effective approaches have been proposed and classical ones have been adapted to take into account the time dependent data [65].

Time series classification is a type of supervised learning where the training data, being composed of labeled time series, is commonly used to build a classifier and the objective is to use the learnt classifier to predict the labels of unseen time series. Classification of time series can be formally stated as the following.

Let $\mathbf{x}_T$ be a time series composed of T time ordered real values $\langle x_1, x_2, \dots, x_T \rangle$ where $\forall t,\ 1 \leq t \leq T,\ x_t \in \mathbb{R}$ is the t^{th} component of the time series $\mathbf{x}_T$, hence $\mathbf{x}_T \in \mathbb{R}^T$. Let $\mathcal{S}$ denotes a set of m training examples with each training example being a couple

$(\mathbf{x}_T^i, y^i) \in \mathbb{R}^T \times \mathcal{Y}$, meaning that each time series $\mathbf{x}_T^i$ is provided together with its associated label $y^i \in \mathcal{Y}$, where $\mathcal{Y}$ is a finite set of classes. For example, in the case of patient's health monitoring, one $(\mathbf{x}_T^i, y^i)$ might consist of the electrocardiograph signal as $\mathbf{x}_T^i$ and the status of the patient's health as y^i. A classifier is then a function: $h_T : \underset{\mathbf{x}_T \mapsto \hat{y}=h_T(\mathbf{x}_T)}{\mathbb{R}^T \to \mathcal{Y}}$ that maps time series of length T to their class labels.

Furthermore, time series classification is challenging especially when very large and massive data set should be handled. Indeed, classification methods have shown some difficulties in scaling up and finding meaningful forms of similarity when time series data sets are massive [38, 65]. Faced with these difficulties, that continue to evolve due to the fast development of digital sources of information leading thus to more important amount of data, a possible solution is to use representations that reduce time series dimentionality while retaining their essential characteristics [103].

2.4 Time series representation

As commonly one data point of time series is considered as one dimension, time series are typically of high dimensionality. This introduces additional complexity when applying machine learning and data mining algorithms. Indeed, considering time series in their raw representation (i.e. their time-domain forms) may not be convenient and may become challenging and costly to visualize, process, store, query, etc. time series of high dimensions. Therefore, a recommended practice to make a good use of high dimension time series is to change the level of their representation.

In the literature, various and several representations have been suggested including the Discrete Fourier Transform (DFT) [3], Discrete Wavelet Transform (DWT) [23], Principal Component Analysis (PCA) [58, 79], Singular Value Decomposition (SVD) [64], Piecewise Linear Approximation (PLA) [93], Piecewise Aggregate Approximation (PAA) [61, 111], Shape Definition Language (SDL) [4], Symbolic Aggregate approXimation (SAX) [67], and many others (see [103] for a well structured review). Some methods quite often used as representations in the literature (e.g. DFT) are not primarily designed to reduce the dimentionality of time series, but they naturally have the ability to achieve it. We propose, in Figure 2.4, a taxonomy of the main representations for numeric time series.

Figure 2.4 shows three main families of representations. The first family includes

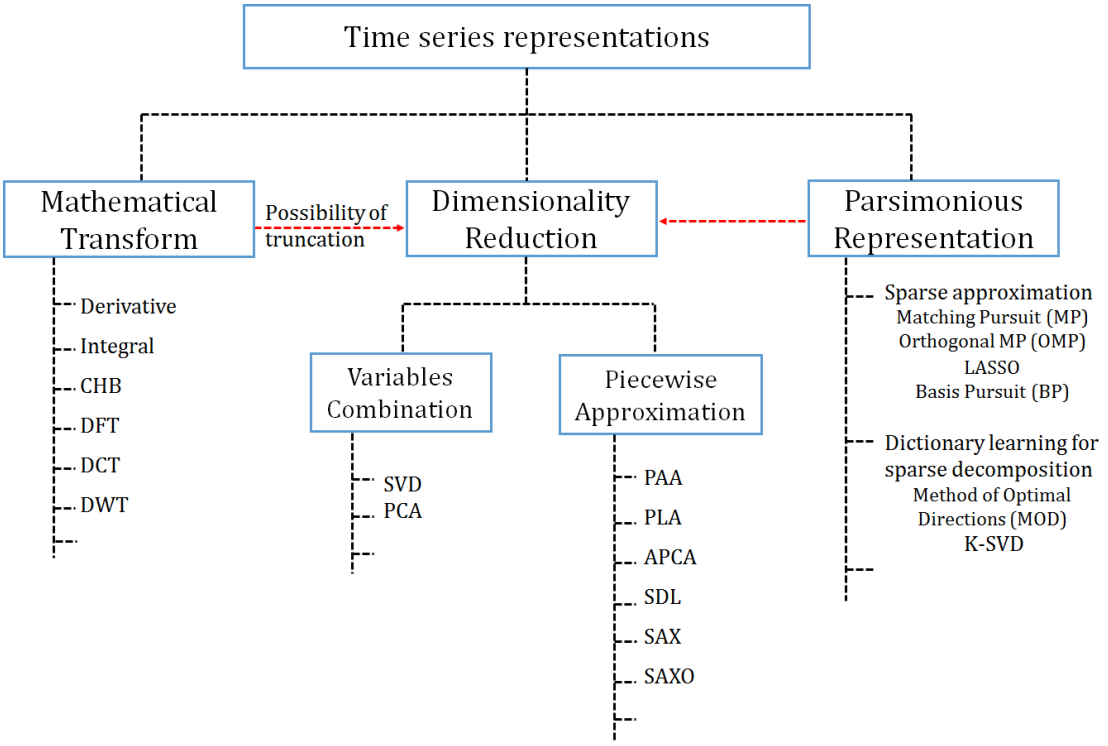


Figure 2.1: A possible taxonomy of time series representations. The representations are mainly divided into three different families.

mathematical transforms such as DFT or DWT. These transformations do not directly reduce dimensionality, but allow to better extract relevant characteristics of time series together with the possibility to reduce their dimension. The second family consists of methods that allow for dimensionality reduction by mean, for example, of compression (e.g. PAA), symbolization (e.g. SAX), etc. Finally, the third family groups parsimonious representations that aim to preserve the principal characteristics of the data while making them as sparse as possible.

Furthermore, in addition to the ability of allowing dimentionality reduction, some properties should be taken into account when selecting a time series representation such as: information gain, ability knowledge extraction, ability for local processing, computational complexity reduction, ability for search acceleration, ability to adapt to the data, user parameter-free, ability to make an incremental updating, etc.

2.5 Time series similarity measures

In many time series learning and data mining tasks, it is required to use a similarity measure in order to quantify the degree of the dissimilarity or similarity between time series. Ding et. al [37] suggested that different similarity measures extract different aspects of similarity when applied to the same problem. It is crucial then to select the appropriate measure that best captures the relevant information to better solve the addressed problem. Furthermore, some properties are desired when selecting a similarity measure such as (i) robustness to noise, outlier, temporal and spatial distortions, (ii) yielding low computational complexity, and (iii) not implying user parameters, etc.

In recent years, and due to the growing interest in using time series that have particular characteristics compared with traditionally used data, a large number of similarity measures were proposed. Most prominent ones include Euclidean distance, a simple and effective distance that implies no user parameter, but is not robust against noise and different forms of distortions (e.g. temporal or spatial distorsions). Dynamic Time Warping (DTW) [10, 14, 73, 92] is more robust against temporal distortions but is computationally expensive (faster variants, such as [91], were proposed). The Longest Common SubSequence (LCSS) [3, 17, 74, 94, 101] is robust against noise and outlier but implies to set a threshold in order to assess the similarity (this parameter should be set with care since it defines the similarity between data). The Threshold Query Execution for LargeSets of Time Series (TQuEST) [8] measures the similarity after coding time series, but provides good results only over some specific data sets. Spatial Assembling Distance (SpADe) [27], based on feature extraction, this measure is robust against the temporal and spatial distortions, noise and outlier but is difficult to scale up. Some distances such as Edit distance with Real Penalty (ERP) [24], Edit Distance on Real sequence (EDR) [25], Extended Edit Distance (EED) [41] extend the Edit Distance (ED) in order to deal with different natures of applications. Many other distances and similarity measures (see [37] for a well structured review on time series similarity measures) have been recently proposed in the literature to respond to the different requirements implied by time series learning problems.

In Figure 2.2, we propose to categorize time series similarity measures according to the properties of a metric space which are summarized in four axioms: identity, separation, symmetry and triangle inequality (see Definition 2.5.1). We call a function that respects the properties of the metric space, distance. Otherwise, it is called a similarity measure (see Definition 2.5.2).

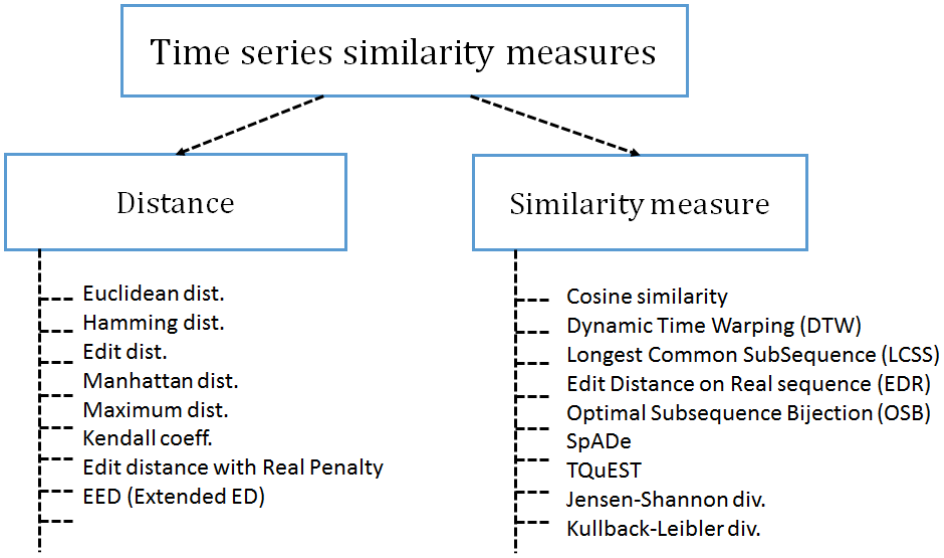


Figure 2.2: A possible taxonomy of time series similarity measures.

Definition 2.5.1 *A metric space is a pair* (E, d) *where* E *is a non-empty set and* $d : \mathrm{E} \times \mathrm{E} \longrightarrow \mathbb{R}$ *such that for any* $x^1, x^2, x^3 \in \mathrm{E}$*, the following holds:*

a1. $d(x^1, x^2) \geq 0$ *(Non-negativity or separation axiom)*

a2. $d(x^1, x^2) = 0 \Leftrightarrow x^1 = x^2$ *(Identity)*

a3. $d(x^1, x^2) = d(x^2, x^1)$ *(Symmetry)*

a4. $d(x^1, x^3) \leq d(x^1, x^2) + d(x^2, x^3)$ *(Triangle inequality)*

Definition 2.5.2 *A similarity measure is a real-valued function that quantifies the similarity between two time series* $\mathbf{x}^1, \mathbf{x}^2 \in \mathbb{R}^T$*:*

$$d(\mathbf{x}^1, \mathbf{x}^2) : \mathbb{R}^T \times \mathbb{R}^T \longrightarrow \mathbb{R}$$

The interest behind making such a categorization is to assist in selecting the appropriate similarity measure that (i) better suits time series in their new representation domain, and (ii) captures the relevant aspects of similarity in the data. For example, this taxonomy can be useful to make a choice about the similarity measure when the

goal is to perform exact or bounded similarity querying. Therefore, the choice of distance/similarity measures depends on the requirements of the problem at hand, time series types and the used representation [37, 39, 50].

2.6 Data streams vs batch data

For many years, conventional learning systems have assumed that the training data were independent and identically distributed (i.i.d.) and they were successfully used to build classifiers over static batches of data. With the first significant development in the area of digital information technology leading to more data, conventional learning systems have succeeded to optimize and improve the performance of their algorithms leveraging the increasing availability of data. The challenge was then to develop storage capacity and optimize data access. However, despite the efforts made to adapt to the rapid growth of digital information, conventional learning systems have shown their limits faced with the growing need to make online decisions and work with data that evolve over time: data streams [1, 9, 42]. In the following, we examine the main properties of static and data streams:

- Static data are immediately available when they are entirely stored as in traditional databases. They can be easily accessible in memory and scanned multiple times (when they are not very large). Problems usually faced with static data include missing values, incorrect values, outliers, noise, etc. This commonly requires some preparation and cleansing of the data before they can be processed.
- Data streams arrive sequentially and are changing continuously in time [42]. Problems arise with data streams when the data flow is rapid and storage, access and multiple scans are not easily realizable.

2.7 Time series early classification

2.7.1 Practical challenges

In many domains, it is natural to acquire the description of an object incrementally, with new observations arriving sequentially over time. This is the case in medicine, when a patient keeps undergoing successive examinations until it is determined that enough evidence has been acquired to decide with sufficient certainty the disease he/she is suffering from. Sometimes, the observations are not controlled and just arrive over

time, as when the behavior of a consumer on a web site is monitored online in order to predict what add to put on his/her screen.

In such situations, the interest is in making a prediction as early as possible because either each example is costly or it is critical to act quickly in order to yield higher returns. However, this generally induces a trade-off as less measurements commonly entail more prediction errors. This is a problem where the quality of the prediction is traded off against the earliness of the prediction.

To further illustrate the growing need to make early predictions, we give the following real application example.

Motivating example One of the growing requirements in today's applications such as in healthcare surveillance is the need to automatically make accurate and quick decisions, without waiting for additional information. A possible application of early prediction in healthcare surveillance is in medical emergency detection systems where there is a special need to detect emergencies, early enough, without making mistakes, before patient's condition deteriorates. Such a targeted supervision can enhance the management of healthcare, improve human and material resources allocation and save unnecessary economical costs (e.g. costs induced by the number of health professionals devoted for surveillance).

In concrete terms, this can be done by automatizing the medical emergency detection systems so that situations that are expected to rapidly deteriorate are accurately and quickly put forward. This will certainly help health professionals to focus on emergencies and take appropriate actions before things get worsen.

2.7.2 Scenario

Early classification of time series can be viewed as a particular case of the conventional time series classification, where, in addition to maximize the quality of predictions, it has the added property of minimizing the time for making a prediction. However, making early predictions means that less measurements are considered which consequently entails more prediction errors. This leads to a trade-off between maximizing the quality of the prediction which is better if more measurements are used and minimizing the time to make a prediction which is better if less measurements are awaited.

First, let us introduce the notations we use throughout this manuscript. Let $\mathcal{S}$ be a

training set composed of complete labeled time series $\{(\mathbf{x}_T^i, y^i)\}_{1\leq i\leq m}$ where $\mathbf{x}_T^i \in \mathbb{R}^T$ is a time series of length T and $y^i \in \mathcal{Y}$ is its associated label.

Definition 2.7.1 (Complete time series) *A complete time series* $\mathbf{x}_T = \langle x_1, x_2, \ldots, x_T \rangle$ *of length* T *is a time series that contains the full information that describe an object over time.*

A typical example would be the position of an object that is moving between a source and a destination. The position, quantified by the horizontal and vertical coordinates of the object, is recorded during consecutive times resulting a sequence of numeric values.

Actually, as mentioned above, since early classification is a particular case of conventional classification, we will make, whenever necessary, systematic comparisons between both classification problems.

When a conventional classifier[1] h_T is learnt given the training set $\mathcal{S}$, the goal is to use h_T in order to predict the class labels of new unlabeled time series $\mathbf{x}_T$ also of length T, $h_T : \underset{\mathbf{x}_T \mapsto \hat{y}=h(\mathbf{x}_T)}{\mathbb{R}^T \to \mathcal{Y}}$. Whereas, given $\mathcal{S}$, an early classifier should arrange to learn a decision function(s) $h_t : \underset{\mathbf{x}_t \mapsto \hat{y}=h(\mathbf{x}_t)}{\mathbb{R}^t \to \mathcal{Y}}$ from the complete time series so that it will be able to predict the label of a new unlabeled and incomplete time series $\mathbf{x}_t$ with variable length t, where $\forall t,\ t \leq T$.

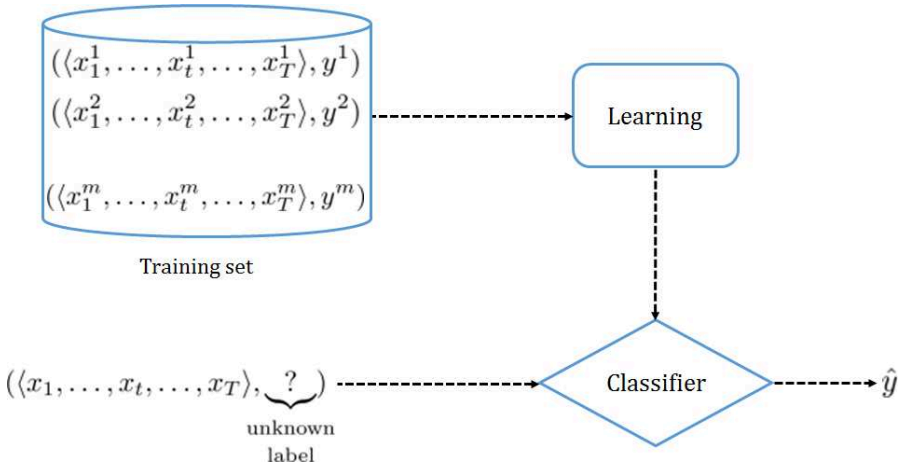
Definition 2.7.2 (Incomplete time series) *An incomplete time series* $\mathbf{x}_t = \langle x_1, \ldots, x_t \rangle$ *is a sequence of the* t *first real values describing an object until time* t.

The learning and prediction processes of conventional and early classification systems are shown in Figure 2.3.

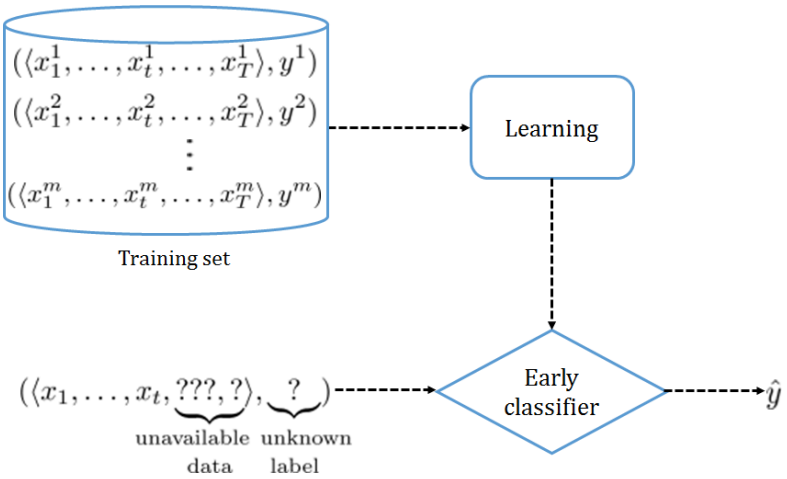
Specifically, an early classifier separately performs the learning and the predictive phases:

- **Learning phase:** an early classifier has access to the training data set $\mathcal{S} = \{(\mathbf{x}_T^i, y^i)\}_{1\leq i\leq m}$ composed of m labeled time series $\mathbf{x}_T^i$ of length T provided with their class labels $y^i \in \mathcal{Y}$. The challenge here is to use the available complete data in $\mathcal{S}$ in order to learn a decision function(s) $h_t : \mathbf{x}_t \mapsto y$, where $\forall t,\ t \leq T$.

[1] Whenever necessary for disambiguity, throughout this thesis, a conventional classifier is simply referred to as classifier.



(a) Conventional classification workflow



(b) Early classification workflow

Figure 2.3: Comparison between the conventional classification and the early classification systems. Both systems have access to a training data set composed of complete time series of length T. In Figure(a), while a classifier is learnt over the complete training time series and then is used to predict the label of a new complete time series of length T, an early classifier, as illustrated in Figure (b), should manage to learn from the complete training time series in a manner that it will be able to predict the label of an incomplete time series of length $t < T$.

One possible approach to do this would be to learn a set of classifiers $\mathcal{H} = \{h_t\}_{1 \leq t \leq T}$, where h_t is an independent decision function induced out of the training time series trimmed to their t first components. As such, the t first components are considered as the explanatory input variables (see Figure 2.4).

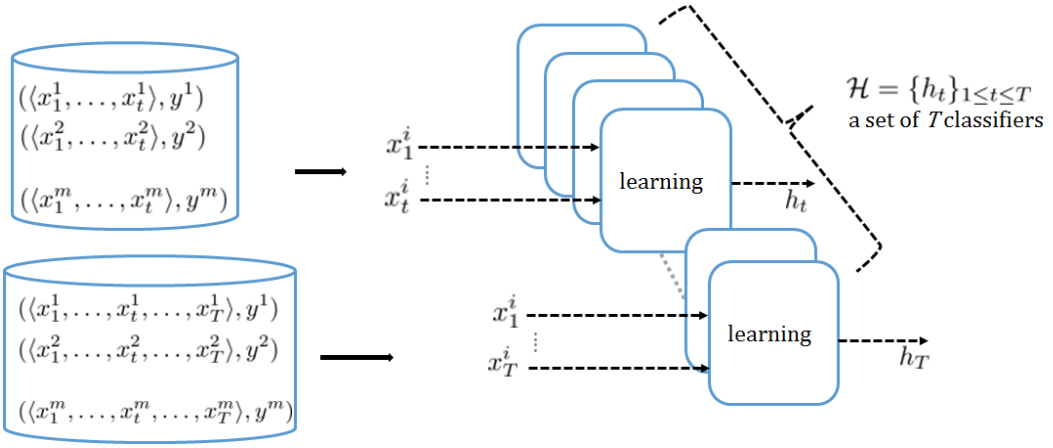


Figure 2.4: A possible implementation of an early classifier is made using a series of classifiers trained over sub-spaces of time series with different lengths.

The approach proposed so far, as shown in Figure 2.4, presents a possible solution for implementing an early classifier. Other possible strategies could be used for implementation. For example, assume there is a function *Transform(.)* that is able to output a fixed-length vector for any variable length input time series data. In this case, only one classifier is learnt, but all the available time series data to be given in the classifier input should be transformed using the *Transform(.)* function. Possible strategies for implementing early classifiers will be discussed in detail in Chapter 3. Later in this thesis, we choose to adopt the approach described in Figure 2.4[1].

- **Predictive phase:** Now, once the early classifier is learnt (regardless of how it is implemented), and for a new incoming time series $\mathbf{x}_t$ of any length t where $1 \leq t \leq T$, the classifier should be able to predict the label of $\mathbf{x}_t$. This can easily be done if one asks, in advance, to have a prediction at a particular future time step(s). In this case, after receiving all measurements until the decided time for prediction, the classifier will output a prediction. However, in the one hand, when the quality of the prediction is what matters, it is obvious that the classifier will wait until the last time to make a prediction as more measurements commonly improve the quality of the prediction. In the other hand, when the interest is in making a prediction as early as possible, the classifier will immediately give a prediction without further delay. Apart of these two extreme cases, the interest is in

[1] This choice does not affect the general understanding and operation of the early prediction system.

making optimal decisions that trade off between the quality and the earliness of predictions. The principal question is therefore how to decide online that now is the optimal time to make a prediction?

To answer this question, we consider the early classification problem as a problem of deciding when enough information has been gathered to make a reliable decision. Therefore, when an incomplete time series $\mathbf{x}_t$ arrives, there should be a decision function that decides whether it is time or not to output the prediction $h_t(\mathbf{x}_t)$ on the class label of $\mathbf{x}_t$. In Figure 2.5, we describe the process of making early predictions as we define it in this work.

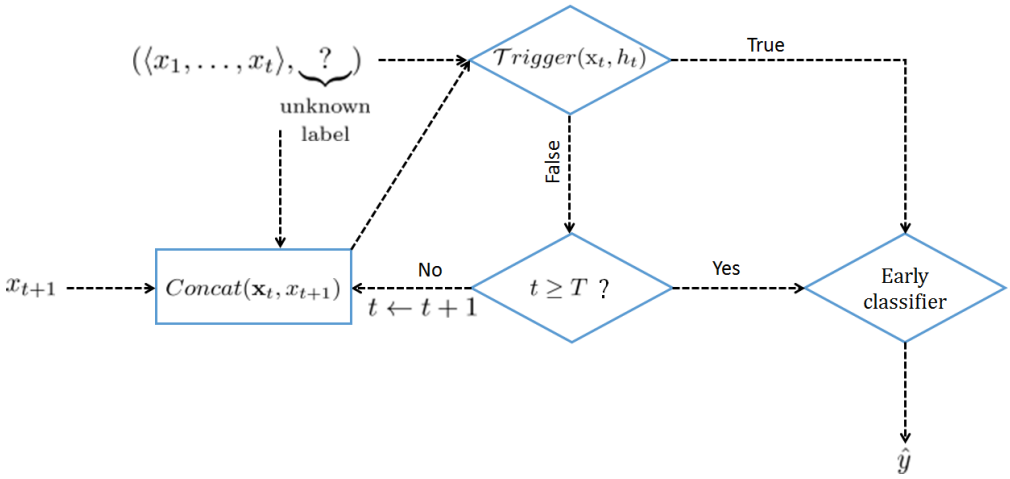


Figure 2.5: Description of the early prediction process: given an incoming time series $\mathbf{x}_t$, the $\mathcal{T}rigger$ function decides to either stop measuring new data and output a prediction on the class label of $\mathbf{x}_t$, or check if new data are available and extend $\mathbf{x}_t$ by adding a new measurement x_t using the *Concat()* function, until time T.

As shown in Figure 2.5, the system looks into the incoming t measurements of the time series $\mathbf{x}_t$. When the $\mathcal{T}rigger$ function decides to make a prediction, $\mathbf{x}_t$ is labeled based on h_t and on the t available measurements. Otherwise, an additional measurement is awaited and the process is repeated until time T. At time T, the prediction is done anyway (leading back to solve the classic classification problem).

In the following sections, we formally introduce the early classification of time series problem and propose a generic framework.

2.7.3 Generic framework

In this section, we propose a generic framework to describe the process of making early classification of time series.

Suppose that the training set $\mathcal{S} = \{(\mathbf{x}_T^i, y^i)\}_{1 \leq i \leq m}$ has been used in order to learn a series of functions $\mathcal{H} = \{h_t\}_{1 \leq t \leq T}$, each function h_t being able to classify examples of length t: $\mathbf{x}_t = \langle x_1, x_2, \ldots, x_t \rangle$ (see Figure 2.4).

Moreover, assume there is a function $\mathcal{T}rigger$ that decides when to stop measuring additional information and output a prediction using h_t for the class label of an incoming incomplete time series.

Let $\mathbf{x}_t$ be an incoming time series at instant t, where $\forall t$, $t < T$. When waiting for an additional instant, a real value x_t is added to the end of $\mathbf{x}_t$ using the function $Concat()$. For instance, the training set $\mathcal{S}$ could be a set of time series generated by measuring the arterial tension of a monitored patients in an hospital.

After learning from $\mathcal{S}$ has taken place, the goal is to choose the earliest time t^* at which a new incoming and still incomplete time series $\mathbf{x}_{t^*} = \langle x_1, x_2, \ldots, x_{t^*} \rangle$ (e.g. sequence of arterial tensions on a period of time about a new patient) can be optimally labeled.

We propose in Algorithm 2 a generic description of time series early classification process.

2.7.4 Required properties

A number of existing early classification approaches suggest different definitions and thus produce different algorithms to make early predictions (see Chapter 4.1 for a detailed state of the art). Some authors like Xing et al. in [106] considered the problem of making early predictions on time series as the trade-off between the earliness and the quality of a classifier. However, others as Parrish et al. view the problem as a problem of classification with confidence from incomplete time series. These and more approaches will be discussed in detail in Chapter 3 and Chapter 4.

By gaining insight into these approaches, we suggest that an early classifier is faced with two interconnected questions:

- The first is which function is able to label an incoming yet incomplete time series. This question brings us to study, in Chapter 3, the different approaches used for implementing early classifiers.

Algorithm 2 Time series early classification framework.

Input:

- An incomplete time series $\mathbf{x}_t$ with $1 \leq t \leq T$;
- $\mathcal{H} = \{h_t\}_{1 \leq t \leq T} : \mathbb{R}^t \longrightarrow \mathcal{Y}$, a set of predictive functions h_t learned from the training set;
- $x_t \in \mathbb{R}$, a new incoming real measurement;
- $\mathcal{T}rigger : \mathbb{R}^t \times h_t \longrightarrow \mathcal{B}$, where $1 \leq t \leq T$ and $\mathcal{B} \in \{\text{True}, \text{False}\}$, a boolean decision function that decides whether it is time or not to output the prediction $h_t(\mathbf{x}_t)$ on the class of $\mathbf{x}_t$;

$\mathbf{x}_t \longleftarrow \varnothing$
$t \longleftarrow 0$
while $(\neg \mathcal{T}rigger(\mathbf{x}_t, h_t))$ **do** /* wait for an additional measurement */
  $\mathbf{x}_t \longleftarrow Concat(\mathbf{x}_t, x_t)$ /* a new measurement is added at the end of $\mathbf{x}_t$ */
  $t \longleftarrow t + 1$
  if $(\mathcal{T}rigger(\mathbf{x}_t, h_t) \,||\, t = T)$ **then**
    $\hat{y} \longleftarrow h_t(\mathbf{x}_t)$ /* predict the class of $\mathbf{x}_t$ and exit the loop */
  end if
end while

- The second is how to decide online that the current instant is the one that yields the optimal prediction. In Chapter 4, we review the main state-of-the-art early classification methods and in Chapter 5 we propose a solution for this question.

In this thesis, and whenever necessary, the methodological choices that determine the outcome of our ideas are mentioned in each chapter. We begin by defining the following:

- The used data are time series of finite length T.
- The time series are considered as they are generated (i.e. in their time-domain forms).
- Each measurement in the time series is considered as an independent input variable within the classifier.
- One class label y is associated to the entire time series.
- There is no possibility to obtain a feedback about the class label unless the entire time series is available.
- The used data are static or changing over time but in more stable manner compared with data streams.

2.8 Comparison with other learning systems

In this section, we compare the early classification system, as defined in this thesis (see Section 2.7), with three closely related learning systems: the offline, online and anytime learning systems. With the aim to clarify the distinction between the different learning systems, some important points of comparison should be considered:

1. Types of input data they deal with
2. The distinction between the learning and the prediction phases
3. Continuous access to a feedback to update the induced decision function
4. The computational time involved in each algorithm

2.8.1 Offline learning

The offline learning is often called *batch* learning from the fact that all the training examples are first collected, then they are given as one batch to the learning system to be build. The offline learning is carried out in two separate phases as a batch process. In the learning phase, a classifier is built without any limits in accessing the training time series data set $\mathcal{S} = \{(\mathbf{x}_T^i, y^i)\}_{1 \leq i \leq m}$. Then, in the predictive phase, the previously learnt classifier is used to predict the labels of new unseen time series $\mathbf{x}_T$ of length T. Figure 2.6 shows the offline learning framework.

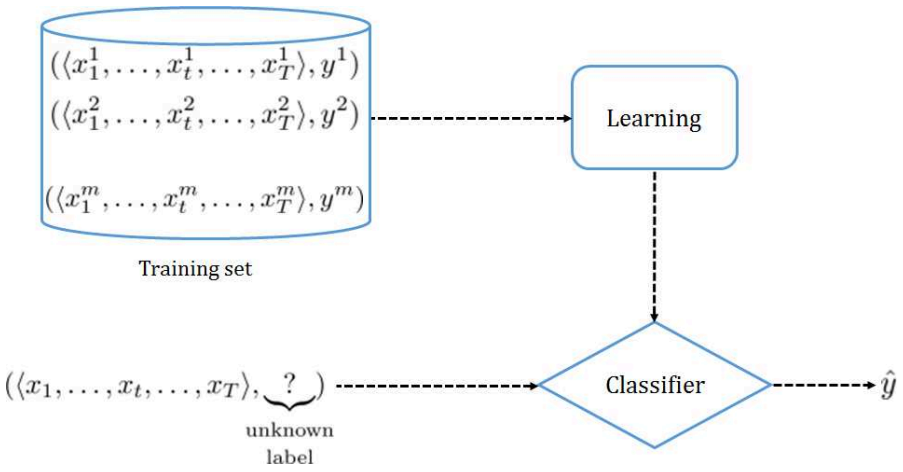


Figure 2.6: Offline learning framework

Major differences between an early classifier compared with an offline classifier are summarized as follows:

- **During the learning process**, an early classifier, as well as an offline classifier, has an unlimited access to the training complete time series $\mathcal{S}$. However, while in the offline case, a function h_T is induced from $\mathcal{S}$ with the objective to map complete time series $\mathbf{x}_T$ to their target class labels y, an early classifier should manage to learn from $\mathcal{S}$ a decision function(s) that is able to map incomplete time series $\mathbf{x}_t$ to their target class labels y.
- **During prediction**, an offline classifier should be able to predict the label of any complete time series of length T. However, an early classifier should be able to predict the class labels of incoming yet incomplete time series of any length t, where $\forall t$, $t \leq T$.

2.8.2 Online learning

In online learning, the training examples are not collected before learning, instead they arrive continuously in time during the learning process. At the same time, with each new measurement, the online learning system is always able to output a prediction. Therefore, all along predicting, the model keeps learning as soon as new data are available to improve its performance and to not become obsolete. This is one of the aspects that characterizes online learning where there is no separate phases for learning and prediction. Online learning process is illustrated in Figure 2.7.

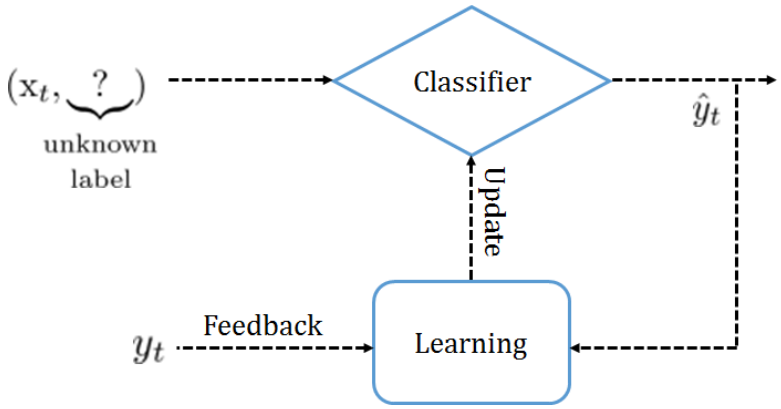


Figure 2.7: Online learning framework

By contrast to early classification systems where their decision functions are learnt offline (regardless of how early classifiers are implemented), online learning systems will be able to adjust their decision functions by keeping learning throughout their use since they have continuous access to the true label of each processed time series. The initial

training data do not need to be as consistent and diversified since the online systems will adapt, during use, to changing conditions [31, 113]. Consequently, while online systems try to improve the quality of their predictions by improving their decision functions when learning from new examples, early classification systems try to improve their predictions online based on each new measurement and past experiences.

2.8.3 Anytime learning

Anytime learning, as introduced by Grefenstette in [53], concerns learning systems that have the properties of anytime algorithms [35]. The basic characteristics of anytime algorithms are: (i) the algorithm has the possibility to be stopped and resumed, at any time, during the learning process, (ii) the algorithm outputs a prediction if terminated at any time, and (iii) the quality of the prediction improves over time.

Most prominent anytime algorithms are interruptable [89]. An interruptible anytime algorithm must produce a valid prediction if interrupted. The framework of anytime learning is shown in Figure 2.8.

In Figure 2.8, the $\mathcal{T}rigger$ function is a part of an external module that manages the resources and can interrupt the algorithm at any time.

Anytime learning systems differ from early classification systems in three major points:

1. Although anytime systems share with the early classification systems the full access to the complete training time series, they are actually constrained by time so that, during the learning process, they can be interrupted before processing all the training examples, thing that affects the prediction quality of the anytime systems.

2. Anytime learning systems may not be able to predict the label of incomplete time series if learnt over complete time series. However, early classifiers should be able to make a prediction whatever the input time series lengths.

3. Anytime learning systems are interrupted when they receive a user external order asking for a prediction. However, early classification systems decide by themselves when to output a prediction.

Specifically, early classification systems, as studied in this work, endow anytime learning systems with the added capacity of deciding by themselves when to stop their intake of new data and make a prediction. In some way, early classification systems could be named as autonomous anytime classification systems. Expect that they do not really

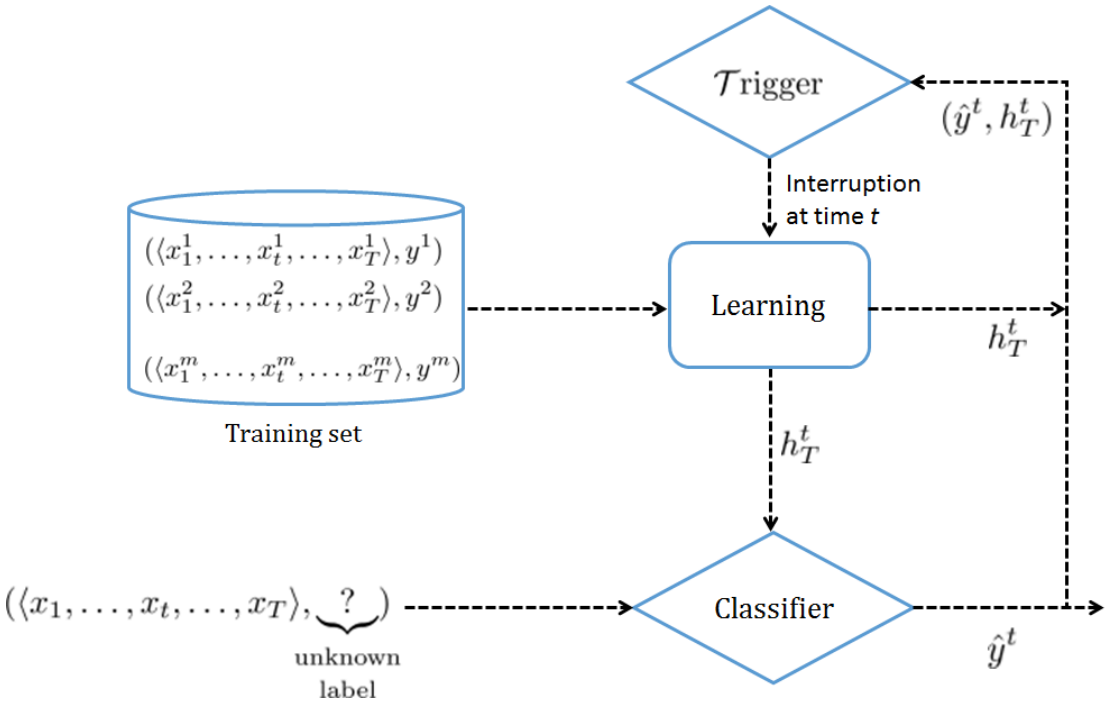


Figure 2.8: At time t, the algorithm, being learnt only on some training time series (i.e. processing all training time series data requires a duration D, where $D > t$), is interrupted and asked to output a prediction on the class label of a new unlabeled time series $\mathbf{x}_T$. h_T^t is the decision function learnt at t using some complete time series. And, $\hat{y}^t$ is the predicted class label of $\mathbf{x}_T$ at time t. The quality of this prediction is expected to improve for times $t + i$, where $i > 0$.

learn at each time step, but really estimate the expected improvement of their prediction in order to decide when it is best to stop.

Summary

In this chapter we have introduced the problem of early classification of time series and defined its main requirements:

- the ability to predict the label of incoming yet incomplete time series, and
- the online optimization of the earliness vs quality trade-off during the prediction

Specifically, an early classifier separately performs the learning and prediction phases. During the learning process, the early classifier has unlimited access to the training data

which are composed of complete time series and the goal is to learn a decision function(s) that maps incomplete time series to their target class labels. Here, we proposed a simple implementation of early classifiers based on using a series of classifiers, each is learnt at each time step t, where $1 \leq t \leq T$, over time series trimmed to their t first measurements. Then, during the prediction, the early classifier is confronted with two tasks. Firstly, time series to be classified are coming in a sequential manner and the early classifier should be able, when decided, to classify incoming time series of any lengths. Secondly, making early prediction usually entails a tension between, on one hand, maximizing the quality of the prediction by requiring additional data, and, on the other hand, minimizing the used data, so as to allow for the earliest possible prediction. This leads to solve, online, an optimization problem where the quality of the prediction is traded off against the earliness of the prediction.

Note that this is a version of the learning with privileged information paradigm introduced by Vapnik and Vashist in 2009 [99].

Finally, we proposed an early classification generic framework that involves a decision function that decides when to stop measuring additional information and output a prediction. This brings early classifiers closer to (i) anytime learning systems, with the added capacity of being autonomous for deciding when to predict, (ii) online learning systems where the prediction is estimated online, and (iii) offline learning systems where the early classifiers are learnt offline.

In the next chapter, we give an overview of the state-of-the-art methods designed to make prediction over incomplete time series.

Chapter 3

Time series early classification: Classifier instantiation strategies

Introduction

The previous chapter provided the background on the time series classification task and introduced the early classification of time series problem. Specifically, early classification of time series aims at predicting as early as possible, but accurately, the class label of an incoming yet incomplete time series. Two indispensable requirements for making early classifications have been identified in Section 2.7:

- the first is the ability to predict the label of an incomplete time series of any length, and
- the second is to optimize online the decision making problem giving the incoming time series.

In this chapter, we deal with the first requirement for making early classifications which consists in designing early classifiers in order to be able to predict the class labels of incomplete time series, and broadly the labels of any-length time series. Some directions have been proposed to adapt with the incompleteness of data when classifying. We go through a number of these systems and suggest other directions (for which we have not found references, such as changing data representation, but think they are interesting strategies that can be used to solve the problem). We propose then four main categories:

1. **Adapting to missing values**: the first category includes methods that do not use all the information contained in the complete training time series. These meth-

ods deal directly with an incoming time series and predict its class label without carrying out any operation to complete it.

2. **Implicit imputation of missing values**: the second category includes methods that implicitly use all the information contained in the complete training time series. They leverage the complete information for making a prediction on the class label of the incomplete time series.

3. **Explicit imputation of missing values**: the third category includes methods that explicitly use all the information contained in the complete time series in order to impute the missing values and then make a prediction on the class label of the imputed time series.

4. **Changing the representation**: the last category, we propose, can include methods that implicitly use all the information contained in the complete time series. These methods change the representation in another time-invariant domain in order to obtain *complete data*. The prediction on the class label is done on the transformed data.

Figure 3.1 shows the different strategies we suggest to handle missing values for incomplete time series classification.

In the following sections, we suggest to examine each of the proposed strategies for implementing early classifiers. The list of the presented methods is non-exhaustive. However, it provides an insight on how early classifiers can be implemented and how incomplete time series are handled when making classifications. In addition to presenting the state-of-the-art, we propose new strategies that were not discussed in the literature for early classification problems (see Section 3.3). Furthermore, since the issue of labeling time series of variable length is not a key focus in this thesis, we only provide ideas, choose one strategy to use later in the thesis, but without performing comparisons between methods. In addition, early classification algorithms are not easily comparable when the task is to optimize contradictory objectives (no Free Lunch Theorem [105]).

3.1 Adapting to missing values

This category includes methods that do not leverage the complete information contained in the training set which is composed of complete time series. Rather, they arrange to directly deal with incomplete time series.

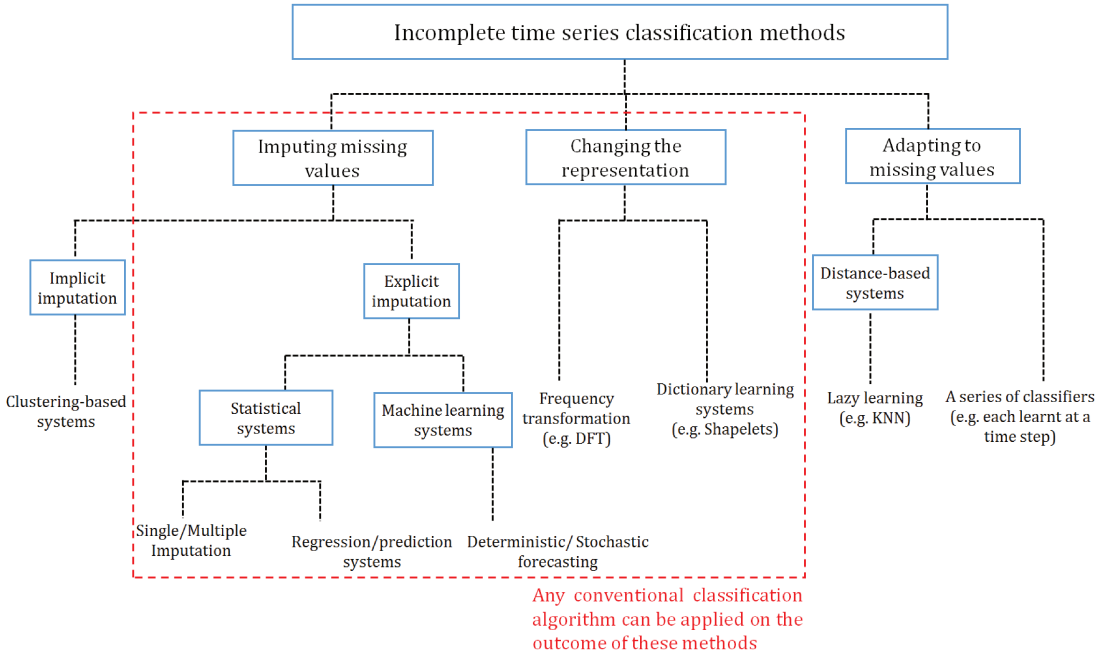


Figure 3.1: Taxonomy of the different strategies suggested to handle missing values in order to make a prediction on the class labels of incomplete time series.

We present here two systems that are able to label incomplete time series without using the complete information contained in the training data set: (i) the first is the distance-based system that uses a distance or similarity measure in order to determine similar cases w.r.t. to the incoming time series and then decides on its class label. (ii) the second system is a series of classifiers, each is learnt at a time step t, where $1 \leq t \leq T$, using training time series trimmed to their t first components.

3.1.1 Distance-based systems

A first simple example would use the K-Nearest Neighbors (KNN) algorithm to label incomplete time series. The KNN algorithm is a non parametric lazy learning system where no explicit learning is done from the training data set. All the training data are used during the prediction phase.

Using the KNN algorithm, one can estimate the label of an incoming time series $\mathbf{x}_t$ based on the class label with the highest frequency from the K most similar time series. The conditional probability $P(y|\mathbf{x}_t)$ of the class given the incoming time series $\mathbf{x}_t$ can

be computed as the normalized frequency of time series that belong to each class in the set of the K most similar time series for the incoming time series $\mathbf{x}_t$. For example, in a binary classification problem ($y \in \{-1, +1\}$), the conditional probability of the class label $+1$ is estimated using:

$$P(y = +1|\mathbf{x}_t) = \frac{\#(y = +1)}{\#(y = +1) + \#(y = -1)} \tag{3.1}$$

When $K = 1$, $\mathbf{x}_t$ is simply classified by the class label of the nearest time series.

Furthermore, to determine the K nearest neighbors for the incoming time series $\mathbf{x}_t$, where, $t < T$, a distance or similarity measure (see Section 2.5) has to be used between $\mathbf{x}_t$ and the training time series trimmed to their first t values.

Commonly, the choice of the best distance or similarity measure is based on the properties of the data and the problem at hand. Otherwise, empirical experiments using different distances and similarity measures (with a fixed K) can be performed on the training data. Then, the distance or similarity measure that yields the most accurate results can be used. Similarly, there is no winning rule for choosing the number of the nearest neighbors to consider. This depends on the problem at hand and differs from a data set to another. Each fixed K value should be evaluated through a set of training-test evaluations.

An example related to making early classification using lazy learning is the Early Classification on Time Series (ECTS) approach.

ECTS

[107] is the first work that formally defined the early classification problem. The approach is motivated by the concept of the so-called *Minimum Prediction Length* (MPL) used to extend the 1-Nearest Neighbor (1NN) classification method with the Euclidean distance to achieve early prediction. Since the 1NN classifiers are *lazy*, the authors added a training step to determine the MPLs. Let $\mathbf{x}_T \in \mathbb{R}^T$ be a complete time series composed of T measurements. Furthermore, let RNN($\mathbf{x}_T$) be the reverse nearest neighbors of $\mathbf{x}_T$ meaning the set of training time series that take $\mathbf{x}_T$ their nearest neighbor. Specifically, the MLP of $\mathbf{x}_T$ is the smallest length k at which the RNN($\mathbf{x}_k$) does not change when revealing the $T - k$ remaining measurements. When a new yet incomplete time series $\mathbf{x}_t$ arrives and if its nearest neighbor $\mathbf{x}_T^i$ has a MLP less than time t, $\mathbf{x}_t$ is classified. This procedure being too conservative, a hierarchic clustering is used and the MLP of each time series in the training set is learnt depending on its cluster membership.

3.1.2 A series of classifiers

The second system that can be used to make predictions on incomplete time series would train a series of classifiers $\mathcal{H} = \{h_t\}_{1 \leq t \leq T}$, where each classifier is a function h_t induced out of the training time series trimmed to their t first components. As such, the t first measurements are considered as explanatory input variables.

A simplest training process would be to independently train the classifiers. Then, during the prediction phase, when a time series $\mathbf{x}_t$ arrives at an arbitrary time t, the prediction of its class label is obtained from the classifier h_t that was learnt on time series of length t. Figure 3.2 illustrates this process.

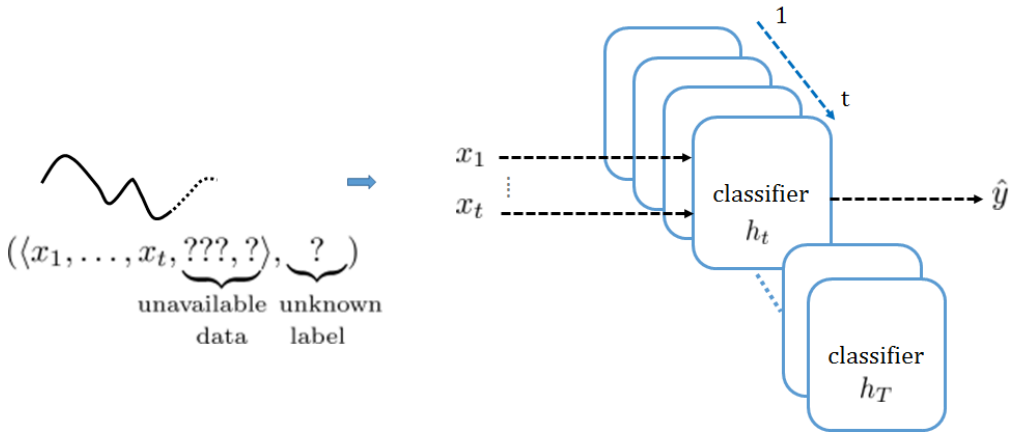


Figure 3.2: Early prediction is made using a series of classifiers trained over sub-spaces of time series with different lengths.

This approach is simple and permits to handle directly incomplete time series without the need to impute missing values. However, it may show some limits in practice when time series are of high dimensions, since the number of classifiers increases exponentially as the dimension of time series increases.

Another alternative to reduce the number of considered classifiers consists in using sliding *window* methods that construct a set of window classifiers $\{h_{w \times k}\}_{1 \leq k \leq K}$, with $w \times 1$ is the length of the first time window. $w \times 2$ is the length of the second window, etc. In this setting, K classifiers are build (instead of T classifiers, where $K < T$) where each covers time series of different lengths (having the same start point). Specifically, each window classifier $h_{w \times k}$ maps an input time series of length $w \times k$ into its class label

y. A new incoming time series $\mathbf{x}_t$ is classified by the window classifier that covers the t time series components.

Discussion

The obvious advantage of these methods is that they permit any conventional classification algorithm to be applied. In that, incomplete time series of any length t are handled directly since a series of classifiers are build at each time step. Examples of methods using a series of classifiers for making early predictions are [32], [72], [55]. Although these methods can be implemented in practice and give adequate performances, they can become rapidly intractable when the dimension of time series are very high. Furthermore, if there are correlations among the data, they will not be captured when assuming the independence hypothesis.

3.2 Imputation-based systems

In this thesis, we consider that during the prediction process, time series measurements are sequentially occurring. As such, the yet unobserved measurements can be considered as missing values occurring at the end of the incoming time series. However, if one waits long enough, measurements will become available.

Conventional classification algorithms are not designed to directly handle incomplete time series, even less incrementally received time series. Faced with this issue, missing data are commonly resolved by estimating and filling in the missing values through applying different imputation techniques including single, multiple imputation, regression and forecasting imputation methods (see [69, 80] for well-structured surveys on missing data imputation techniques).

Globally, these methods try to impute the missing values then make a prediction on the class label of the imputed time series. In this case, a two stage system is performed: (i) first, a prediction model would be learnt using the complete training time series, then it would be used to estimate the missing values in the incoming time series. Once the incoming time series is imputed, (ii) a classifier that was also learnt over the complete training time series is used to predict the class label of the imputed time series (see Figure 3.3).

Other methods try to handle incomplete time series and make a prediction by lever-

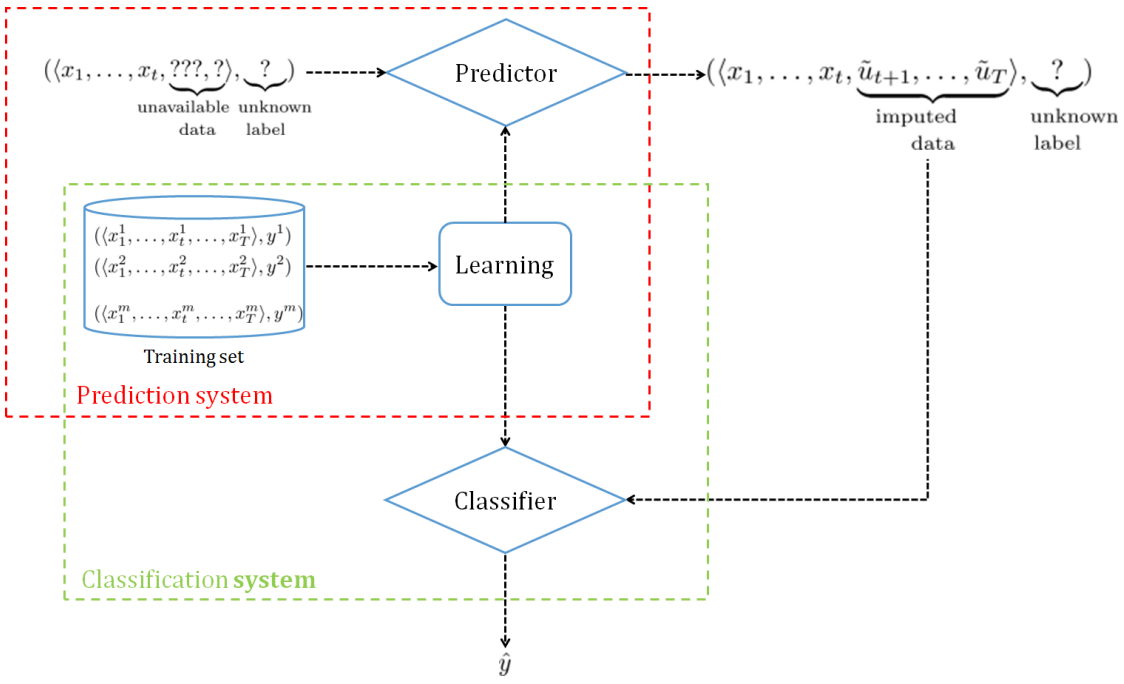


Figure 3.3: The class label of an incoming time series $\mathbf{x}_t$ is predicted after imputing the missing values in $\mathbf{x}_t$. Two-stage process is performed: (i) first a predictor is learnt over the training complete time series with the objective of estimating the missing values, then (ii) a classifier is also learnt over the training complete time series, to be subsequently applied on the imputed time series.

aging information from the training data that contain complete time series, without explicitly imputing the missing values. In this case, during the learning phase, the classifier should arrange to map incomplete time series to their class labels based on complete time series.

In the following, we examine the most prominent imputation techniques and divide them, as discussed above, into two categories depending on whether or not they explicitly use the complete information contained in the complete training time series.

3.2.1 Explicit imputation

This category includes the methods that explicitly use the complete information contained in the training data set in order to complete the missing measurements before predicting the label of the imputed time series. The problem of missing data imputa-

tion is a broad topic, but according to the proportion and the type of the missing data, different solutions can be used. The most common used methods are:

- **Single imputation techniques:** they commonly impute the missing values with fixed values such as zeros, (conditional) mean values, the Last Observation Carried Forward (LOCF) and many others [5]. Using such a technique over time series may be inefficient due, for example, to ignoring the inherent uncertainty in the imputed values which can result strong correlation and often lead to biased parameter estimation (e.g. the estimation of the variance).

- **Forecasting-based imputation techniques:** they are commonly divided into two families: deterministic or stochastic predictions. Deterministic approaches assume that time series data are generated by an unknown function and the objective is to identify the most appropriate function that better fits the data. Once the function is fit, it can be used to estimate the missing values. Examples include Least Squares Approximation [5, 21, 52], Maximum Likelihood (ML) algorithm [45], etc. Stochastic forecasting approaches use generally Box-Jenkins' Autoregressive Integrated Moving Average (ARIMA) models [19] to find the best fit of the data and then estimate the missing values. These methods may involve unsubstantiated hypotheses of the underlying distribution and often have high computational complexity.

- **Machine Learning imputation techniques:** include the K-Nearest Neighbors algorithm [104], a variant of the SVM algorithm [95] and many others, see [90]. Generally, linear regression is used for numeric time series. Its advantage is that it uses the training data to fit the predictive function, however this commonly overestimates the correlations and underestimates the variances.

- **Multiple imputation techniques:** developed mainly by [87], they can remedy the above mentioned disadvantages. They replace each missing value by two or more values which introduce random variations into the imputation process. The result is two or more data sets, each with different imputed values. Then, each data set is analyzed using the same desired method. The analyzes are then combined in order to reflect the additional variability and uncertainty due to the missing data. Although multiple imputation techniques have the advantage of being consistent and asymptotically efficient, they also have the disadvantage of requiring considerable data processing and calculation of estimates provided some assumptions are met. This may not be convenient for online decision making problems.

In the following paragraph, we give examples using explicit imputation techniques for making early predictions.

Examples of forecasting-based early classification approaches

As an example of classifying with incomplete time series, the approach in [6, 78] proposes an incomplete decision rule that imputes the distribution over the missing values in order to make a prediction on the class label given the available time series measurements. In this approach, a complete time series is modeled as a random variable whose distribution is dependent on the current incomplete time series and the training data. In order to make this approach tractable, the authors propose to explore only a subspace restricted to a small number of measures of the input space, instead of the whole space that contains all the possible continuations of $\mathbf{x}_t$. Furthermore, compared to the use of standard imputation techniques with the aim to solve a classification task, this approach similarly tries to estimate the distribution of the data given an incomplete time series in order to efficiently predict its class label.

A second example is [28] where the idea is to combine functional prediction with probabilistic functional classification in order to predict the daily traffic flow from partially observed trajectories. Daily traffic flow trajectories are represented as a realization of a mixture of stochastic processes. First, the *K-centers functional clustering* (k-CFC) [29] method is used to identify distinct daily traffic and organize them into different groups. Then, given a partial daily traffic, forecasting the missing part is performed based on trajectories in the nearest group.

Discussion

First, we recall that the explicit imputation process should be done online with each new observed measurement. The objective is to complete the incoming time series so that, the conventional classifier that was learnt using complete time series will be used to predict the class label of the imputed time series.

Using explicit imputation techniques for predicting the missing values is a natural and straightforward strategy to complete the incoming time series in order to predict its label using conventional classification approaches. However, the task of predicting the missing values becomes problematic mainly because of the following reasons:

- When the considered time series are of high dimensions (i.e. the training time series are composed of a large number of measurements) and only few measurements are

observed, predicting the missing values (which are consecutive) becomes a source of errors due to the accumulated estimation errors when using single imputation and regression techniques.

- Using the multiple imputation approach may be possible and yield better results if there is no time constraints.

3.2.2 Implicit imputation

The second category of imputation-based approaches includes methods that implicitly leverage the information included in complete data to predict the class label of an incoming incomplete time series.

To better present this strategy, we give examples of how implicit imputation is used for making early predictions.

Clustering-based approach

The idea is to identify meaningful subsets of complete time series and try to leverage the complete information contained in these subsets for predicting the class label of the incoming yet incomplete time series. This approach is performed in three steps as shown in Figure 3.4:

1. Complete training time series $\{(\mathbf{x}_T^i, y^i)\}_{1\leq i\leq m}$ are clustered into K meaningful subsets $\{C_k\}_{1\leq k\leq K}$ using a clustering technique,

2. Then, the most similar cluster is determined using an appropriate distance or similarity measure d_k, where $d_k = dist(\mathbf{x}_t, \bar{\mathbf{c}}_k)$ is the distance between the incoming time series $\mathbf{x}_t$ and the average time series $\bar{\mathbf{c}}_k = \langle c_1, c_2, \ldots, c_T\rangle$ representing each cluster (here $\bar{\mathbf{c}}_k$ is the average time series of the cluster C_k, it could be the centroid of each cluster, the median, etc.). The distance between $\mathbf{x}_t$ and $\bar{\mathbf{c}}_k$ is computed using a distance between the first t components of the two series.

3. Finally, the new incoming time series $\mathbf{x}_t$ is classified by a majority vote of time series that belong to the nearest cluster. $\mathbf{x}_t$ is thus assigned to the most common class in the nearest cluster.

This approach is conceptually close to the KNN-based approach discussed earlier. The resemblance lies in the crucial role that play distances and similarity measures for

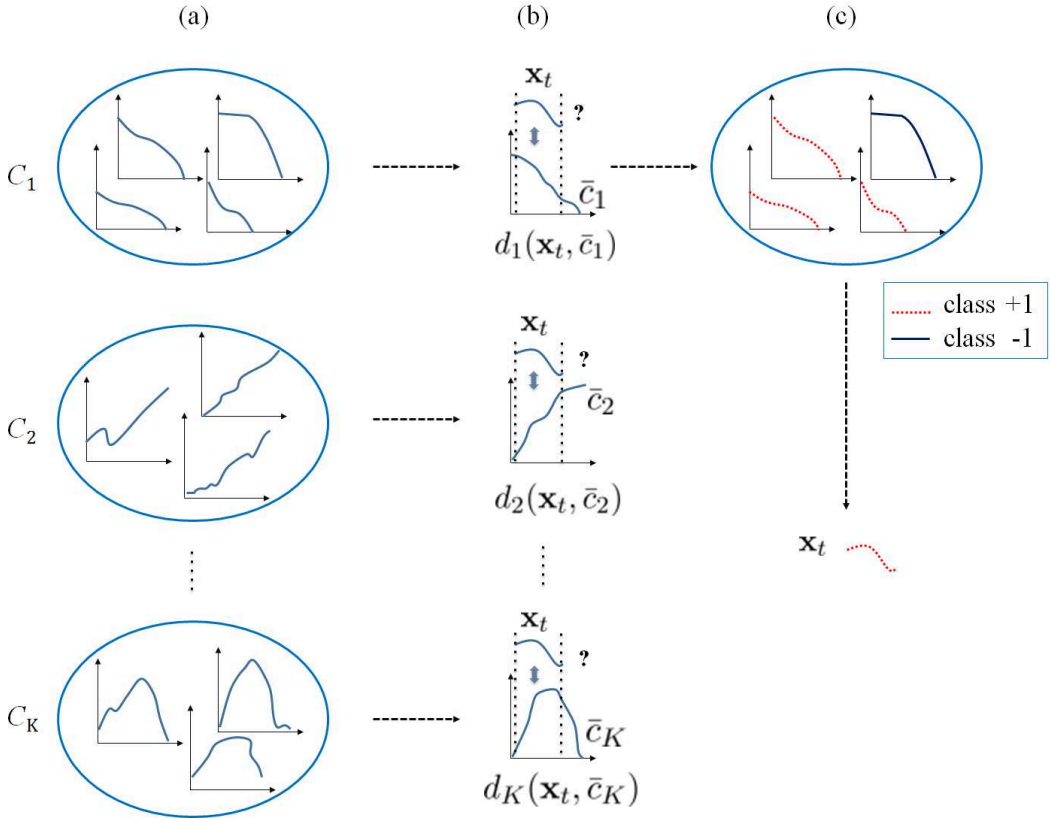


Figure 3.4: The clustering-based approach is performed in three steps: **(a)** identify meaningful subsets of complete time series in the training set: C_k. **(b)** Find the most similar cluster using an appropriate distance or similarity measure d_k, where $d_k = dist(\mathbf{x}_t, \bar{\mathbf{c}}_k)$ is the distance between the incoming time series $\mathbf{x}_t$ and the complete time series representing the cluster C_k (here $\bar{c}_k$ is the average time series of the cluster C_k). **(c)** $\mathbf{x}_t$ is assigned to the most common class (in this example, class +1) in the most near cluster (here C_1) by a majority vote of time series among the same cluster.

identifying the nearest time series and thus assign the incoming time series to the most common class decided by the nearest time series. However, while the clustering-based approach makes a selection of time series among the training data set depending on the incoming time series in order to leverage their possible continuations, which implicitly permits to impute the incomplete time series and then predict its label, the KNN does not use complete information contained in the training data.

Closely related examples for early classification methods using these techniques include the Early Classification on Time Series (ECTS) approach [107] and the Early Fault

Classification (EFC) method [20] using distances to guide early classifications.

EFC

The Case-based Reasoning method for fault detection [20] proposed to perform as early as possible failure diagnosis of a simulated dynamic system in a laboratory plant. The aim of failure diagnosis is to detect faults of interest and their causes quickly enough to avoid the failure of the overall system. The faults are described by time series. A K-nearest neighbors method (KNN) is applied as a retrieving algorithm using different distances such as the Euclidean distance, the Dynamic Time Warping (DTW) [14] distance and Manhattan distance, in order to label an incoming time series $\mathbf{x}_t$ based on a set of similar cases.

Discussion

In this category, the imputation of missing values in the incoming time series is implicitly done through leveraging the complete information contained in the nearest time series or the nearest cluster (i.e. in term of distance). The quality of the prediction mainly depends on the ability of the used distance or similarity measure to identify meaningful subset(s) of time series where there is inherent information about the common class label (see the example illustrated in Figure 3.4).

Furthermore, in such setting, the learning is generally lazy and multiple choices have to be set such as the clustering algorithm to use, the distance or similarity measure, the number of clusters to consider, etc.

3.3 Representation-based systems

When missing values are occurring in the incoming time series during the predictive phase (here, we recall that the missing values are consecutive, and concern unobserved measurements that will be available after some time steps), a time series $\mathbf{x}_t$, where $t < T$ (T is the length of complete time series in the training set), is considered incomplete without taking into account the domain in which it is represented. However, if one changes its representation, the time series may become *complete*.

Indeed, the definition of the incompleteness of time series is closely dependent on the domain in which it is represented. For instance, if raw time series are incomplete in the time domain, they can become *complete* (under some conditions) in the frequency

domain (see Definition 3.3.1). In such a setting, a key property of the representation method to use is to be invariant against the length of the time series in the time domain.

Definition 3.3.1 *In our context, we define a representation method that is invariant with respect to the length of a time series* $\mathbf{x}_t \in \mathbb{R}^t$ *in the time domain, as a function, denoted by* $\phi(\mathbf{x}_t) \in \mathbb{R}^K$*, that maps* $\mathbf{x}_t$ *to its representation* $\phi(\mathbf{x}_t)$*, such that,* $\forall t$*,* K *remains constant.*

According to Definition 3.3.1, we consider that, in some situations where some conditions are met, the incomplete time series in the time domain can become complete in another domain through the function ϕ.

Such a transformation function may imply two types of imprecision: (i) the first imprecision is due to the incompleteness of the data, and (ii) the second imprecision concerns the changing from one domain to another. Both imprecision resources are indiscernible when attempting to recover the original time series. In the following, we examine each imprecision resource.

Imprecision due to the incompleteness of the data

Let $\mathbf{x}_T$ be a time series of length T and $\mathbf{x}_t$ be the same time series trimmed to its t first measurements where $t \leq T$.

Let $\phi : \mathbb{R}^t \longrightarrow \mathbb{R}^K$, $\forall t \leq T$, be a representation function:

$$\text{if } \phi(\mathbf{x}_T) \in \mathbb{R}^K \text{ and } \phi(\mathbf{x}_t) \in \mathbb{R}^K, \text{ then } \mid \phi(\mathbf{x}_T) - \phi(\mathbf{x}_t) \mid \geq 0.$$

From this, we would refer to that the incompleteness of the time series $\mathbf{x}_t$ may lead to some imprecision(s) when applying the transformation $\phi(\cdot)$.

Imprecision due to changing the domain

Changing the representation of time series from one domain to another commonly leads to information loss (even if the time series are complete in the temporal domain). Indeed, when the original data can not be reconstructed perfectly, there was certainly a loss of information during conversion (e.g. aliasing, leakage).

The choice of the representation is crucial since the obtained data should remain as informative as possible in order to be useful in practice.

Numerous time series representation methods have been suggested in the literature (see Section 2.4). Among these representations we are interested only in the ones that

respect Definition 3.3.1.

In the following, we suggest to examine two strategies that permit to circumvent the incompleteness of the incoming time series by changing its domain in order to make a prediction on its class label. We illustrate the first strategy by giving the example of the frequency transformation technique. The second strategy is conceptually similar to the frequency transformation but provides a richer data decomposition which results in an adaptive and data-driven dictionary.

3.3.1 Frequency transformation

In this example, the representation of a time series $\mathbf{x}_t$ is changed into its representation in the frequency domain using the Discrete Fourier Transform (DFT) [23].

When using the DFT, the time series $\mathbf{x}_T$ which is composed of discrete data points over time, is decomposed into a sum of sine terms of different frequencies, each of which represents a frequency component.

DFT thus maps $\mathbf{x}_T = \langle x_1, \ldots, x_T \rangle$ to T coefficients $c_1^F, \ldots, c_T^F$ where each coefficient c_j^F is defined as:

$$c_k^F = \sum_{i=1}^{T} x_i W_T^{ik} \tag{3.2}$$

where $W_T = \exp^{-j2\pi/T}$.

The Discrete Fourier Transform of $\mathbf{x}_T$ is defined as the sequence of its Fourier coefficients c_k^F:

$$DFT(\mathbf{x}_T) = \langle c_1^F(\mathbf{x}_T), \ldots, c_T^F(\mathbf{x}_T) \rangle \tag{3.3}$$

If the reconstruction of $\mathbf{x}_T$ is not affected when certain coefficients are ignored, one can truncate its DFT sequence, retaining only a limited number K[1] of the relevant coefficients:

$$DFT_K(\mathbf{x}_T) = \langle c_1^F(\mathbf{x}_T), \ldots, c_K^F(\mathbf{x}_T) \rangle \tag{3.4}$$

Now, when a new incomplete time series $\mathbf{x}_t$ arrives, its Discrete Fourier Transform is determined and then truncated, retaining only its K first coefficients.

$$DFT_K(\mathbf{x}_t) = \langle c_1^F(\mathbf{x}_t), \ldots, c_K^F(\mathbf{x}_t) \rangle \tag{3.5}$$

[1] The number K of retained coefficients can be easily determined from the training data set.

Once the incoming time series and the training data set are transformed through the same policy, any conventional classification algorithm, learnt over the transformed training time series, can be thus applied to predict the class label of the $DFT_K(\mathbf{x}_t)$.

Discussion

This example of frequency transformation clearly shows that it is possible to extract rather complete information from incomplete time series when some conditions are met.

It should be pointed out, that such transformation does not directly lead to transformed data of a same length, some pre-processing step should be achieved. For example, the truncation of the DFT sequence of a time series can be done retaining only K coefficients. This parameter should be carefully estimated from the training data to avoid loss of information.

3.3.2 Dictionary-based systems

Considering all measurements of time series (including sometimes noise and aberrant ones), in a classification task, may not always be the best way to make accurate predictions. Instead, extracting relevant and valuable features[1] that are effective for the classification problem can considerably increase the predictive performances [110]. The set of these extracted features that manifest the target class labels in the training set is commonly known as a dictionary. In this context, early predictions can be achieved by exploiting such a dictionary. Figure 3.5 describes this idea.

Dictionary-based approaches go along these following steps:

- In the learning phase, a dictionary of features which are characterizing as much as possible the target classes is built. One of the key principles of dictionary learning is that the dictionary has to be inferred from the training data.
- In the predictive phase, when a match (e.g. based on a distance) between a dictionary feature and a sub-sequence in the new incoming time series x_t is found, x_t can be labeled before all measurements are available.

Examples of early classification methods using the so-called dictionary based learning are [108] and [48]. In these approaches, the elements of the dictionary are called *shapelets*. A shapelet as introduced in [109] represents a portion/sub-sequence of a time series characterizing the target class label. Specifically, a shapelet is defined by a triplet $f =$

[1] We define a feature as a sub-sequence that characterizes the target class label in a time series.

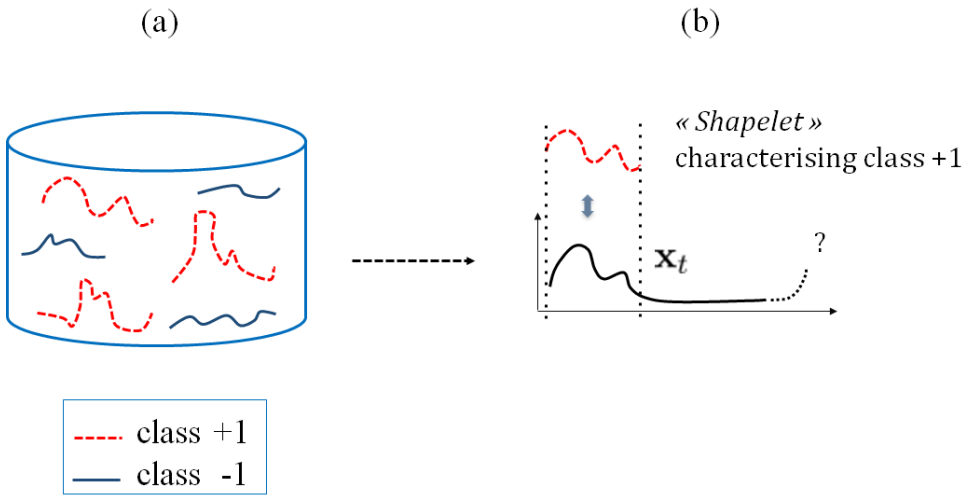


Figure 3.5: The dictionary-based approach is performed in two steps: (a) a dictionary of features manifesting the class labels in time series is inferred from the training data set, (b) when a new time series $\mathbf{x}_t$ arrives, its distances from the dictionary elements are computed. Early prediction is made when a (strong) match is found between a feature from the dictionary and $\mathbf{x}_t$.

(s, δ, y) where s is a distinctive[1] sub-sequence or feature, a distance threshold δ allows measuring the closeness/similarity of sub-sequences to s and the class y.

In the training phase, a set of shapelets is extracted and their associated distance thresholds are learned. In the prediction phase, a new incoming time series $\mathbf{x}_t$ is labeled when a match with a shapelet is found.

Discussion

This category of approaches may be useful only in some domains where the time series data have typical shapes that are common for a specific class label. For example, in medical informatics, some diseases are rapidly identified thanks to abnormalities included in a patient signal since it has a typical behavior. However, when time series are very volatile and no distinctive features representing each class, such as in power load or financial market data, there will be no interest in applying dictionary-based algorithms.

Moreover, building the dictionary and making a prediction on the class label of an incoming time series that involves all of the dictionary elements may be computationally very expensive.

[1] A shapelet is considered distinctive if all time series matching f have a high probability to belong to class y.

Summary

In this chapter, the main focus was on the problem of how one can instantiate early classifiers so that they will be able to make predictions on the class labels of incomplete time series.

We proposed a categorization of the possible strategies for early classifier implementations. While some systems were designed to impute the missing data before making a prediction, other systems have managed to directly adapt with incomplete data. We provided a list of strategies depending on explicit or implicit use of the complete information contained in the training data and propose to change the representation of time series. This list is non-exhaustive but it gives valuable insights into incomplete time series classification issue.

In our context, the implementation of early classifiers is not the crux task for making early predictions. We have shown that for labeling incomplete time series, conventional classification systems can hence be used after some pre-processing steps. From this, only one of the presented strategies will be used later in this thesis.

In the following chapter, we review the state-of-the-art of early classification methods with a focus on online decision making issue.

Chapter 4

Time series early classification: Online decision making

Introduction

The emergence of the early classification of time series problem is manifestly a recent trend that is built each year and is being developed gradually mainly due to the growth of its scope and the increasing need to make early predictions. This is confirmed by numerous references in the literature where the problem is narrowly defined as a problem of classification with confidence from incomplete time series.

In Chapter 2, we have defined the problem as a problem of online decision making and identified two tasks for making early classifications on time series:

1. the first task concerns the implementation of early classifiers so that they will be able to make predictions on the class labels of incomplete time series. In Chapter 3, we have suggested a number of different strategies to endow classifiers with the ability to label incomplete time series.

2. the second task pertains on when to decide online on the optimal time for making a prediction.

In this chapter, our focus is on the second requirement. To make online decisions, we argue that early classification systems should be explicitly endowed with a decision function that decides when enough information has been gathered to make a reliable decision. We suggest thus to examine state-of-the-art early classification methods according to the early classification framework we presented in Chapter 2, and, repeated below:

Algorithm 1 Time series early classification framework.

Input:

- An incomplete time series $\mathbf{x}_t$ with $1 \leq t \leq T$;
- $\mathcal{H} = \{h_t\}_{t \in \{1,\dots,T\}} : \mathbb{R}^t \longrightarrow \mathcal{Y}$, a set of predictive functions h_t learned from the training set;
- $x_t \in \mathbb{R}$, a new incoming real measurement;
- $\mathcal{T}rigger : \mathbb{R}^t \times h_t \longrightarrow \mathcal{B}, t \in \{1, \dots, T\}, \mathcal{B} \in \{\text{True}, \text{False}\}$, a boolean decision function that decides whether it is time or not to output the prediction $h_t(\mathbf{x}_t)$ on the class of $\mathbf{x}_t$;

$\mathbf{x}_t \longleftarrow \varnothing$
$t \longleftarrow 0$
while $(\neg \mathcal{T}rigger(\mathbf{x}_t, h_t))$ **do** /* wait for an additional measurement */
  $\mathbf{x}_t \longleftarrow Concat(\mathbf{x}_t, x_t)$ /* a new measurement is added at the end of $\mathbf{x}_t$ */
  $t \longleftarrow t + 1$
  if $(\mathcal{T}rigger(\mathbf{x}_t, h_t) \,||\, t = T)$ **then**
    $\hat{y} \longleftarrow h_t(\mathbf{x}_t)$ /* predict the class of $\mathbf{x}_t$ and exit the loop */
  end if
end while

From the above presented framework, one can see that the $\mathcal{T}rigger$ function plays a crux role for making an optimal prediction when the decision should be made online.

In the rest of this chapter, we review the main early classification approaches suggested in the literature and critically examine them according to the $\mathcal{T}rigger$ function they propose.

4.1 Trigger function: State-of-the-art and discussion

In this state-of-the-art, seminal works and closely related core works for the early classification of time series problem are critically examined according to the flowing requirements:

- The ***Trigger* function** (see Algorithm 2), that decides when to stop measuring additional information and output a prediction,
- The **optimization** of the time vs quality trade-off which guaranties to find the optimal balance between the earliness and the quality of the prediction,
- The **adaptability** with respect to the incoming time series $\mathbf{x}_t$. An early classification approach is considered adaptive when it takes into account $\mathbf{x}_t$ in the decision

making process,

- The **non-myopia** characteristic: a non-myopic algorithm in our context does not only decide if the current instant is the optimal time to make a decision, but it also estimates when the optimal time is likely to happen. In words, even it is decided to delay the prediction, the algorithm is able to provide an estimate of when the optimal prediction time is likely to occur.

Indeed, when making early predictions, there must be a procedure that controls the time (here, it is called $\mathcal{T}rigger$) so that it is able to stop measuring additional information and output a prediction. Otherwise, there will be no control of the earliness of the prediction and results would be obtained once all information are gathered. In a way, this function reflects how one considers the balance between the time and the quality of the prediction. In fact, when fixing either one, the earliness or the quality, the optimization of the trade-off becomes limited either on optimizing the earliness at the expense of the quality of the prediction or vice versa. Although, generally, the aim is to simultaneously optimize both objectives.

In addition to involve the earliness and the quality of the prediction in the decision making process, it is highly desirable to make adaptive and non-myopic decisions. Adaptability ensures that the optimal time of decision depends on the incoming time series. The property of making non-myopic decisions offers valuable opportunities compared to myopic sequential decisions. For instance, when the prediction is about the breakdown of an equipment or about the possible failure of an organ in a patient, this forecast capacity allows one to make preparations for thwarting as best as possible the breakdown or failure, rather than reacting in haste at the last moment.

These are goals we would like to achieve when suggesting a new approach for making early classifications. Bearing this in mind, we first start by examining state-of-the-art early classification methods according to these requirements. In the following paragraphs, the most prominent methods are presented chronologically.

4.1.1 BIBL (2004)

In 2004, Rodríguez and Alonso [84] were the first to bring up the problem of making early classification (although it was not the main objective of the problem they addressed, but a natural finding).

The principal goal of the Boosting Interval-Based Literals (BIBL) [84] method was to design a system able to classify time series of variable lengths. Consequently, an interesting result is obtained which is the capacity to label partial sequences.

In this approach, each time series is divided into a number of time intervals where each is described by a literal[1]: (i) relative literals that describe the trend of the time series such as *increases* or *stays*, and (ii) region literals that consider the presence of the time series over some intervals such as *always* or *sometimes*. Each literal is considered as a base (weak) classifier that is able to output a weight associated to each class. The ensemble of base classifiers, using Adaboost [40] technique, was exploited to make predictions on available measurements which are prefixes of the complete time series. Since the classifier is a linear combination of literals, when suffixes of the complete time series are missing and their weights are set to zero. In this way, when a new time series $\mathbf{x}_t$ comes, it is assigned to the class label with the greater weight after computing weights for each class.

Discussion

- **Trigger function:** There is no explicit function that decides when to stop measuring additional information and make a decision. The only condition is when, at least, all measurements in the first interval that enables constructing one literal are available. This condition is required in order to obtain a result. The other literals whose intervals are still unavailable are simply omitted from the ensemble classifiers.

- **Optimization of the time vs quality trade-off:** Although this approach has introduced the concept of early classification and served as a baseline method, it does not address how to find the optimal time to make a reliable prediction on incomplete time series. Hence, this approach does not optimize the earliness and does not guarantee the quality of the prediction.

- **Adaptability:** The fact that time intervals are described by literals and thus summarized (which often entails information loss) makes it harder to provide adaptive predictions with respect to the incoming time series.

- **Myopia:** The decision making process is considered myopic since there is no procedure able to estimate the optimal time for making a prediction in the future.

[1] When discussing the (formal) language of propositional logic, a literal can be referred to as a basic statement or sentence which is a type of syntactic formula that outputs true or false. For example, increases(Observation, Variable, Beginning, End, Value) is a literal that outputs true if the difference between the values of the *Variable* for *End* and *Beginning* is greater or equal than *Value*.

BIBL [84] only focuses on making predictions based on incomplete time series without addressing the issue of estimating the optimal time for making a reliable prediction. In that, the trade-off between the time and the quality of prediction is not optimized. As it is, BIBL only satisfies the first requirement which is the ability to label incomplete time series, and does not respect the second requirement for making online early predictions.

4.1.2 EFC (2006)

So soon after, Bregón et al. [20] (in collaboration with Rodríguez and Alonso [84]) have suggested a method for early classification of faults (EFC) in a dynamic system based on a Case-Based Reasoning[1] (CBR) algorithm. A case base is composed of time series measured by sensors and describing a fault which is their associated class label. The advantage of using a CBR method is twofold. First, when new cases are available, they can be added to the training set without the need to update the classifier. Second, CBR methods are able to predict the class labels of time series of different lengths, without the need to build a classifier for each length.

In this approach, a K-Nearest-Neighbors (KNN) method is used together with three different similarity measures: the Euclidean distance, the Manhattan distance and the Dynamic Time Warping distance (DTW). Then, when a fault is detected, its label should be predicted before a user-specified time threshold is met (this corresponds to the maximum time allowed to identify a fault). The experimental study on simulated data showed that the most important increase, regarding the KNN (K is ranged from [1,3,5]) accuracy, occurs when going through thirty percent to fifty percent of the complete time series length.

Discussion

- **$\mathcal{T}$rigger function:** The *$\mathcal{T}$rigger* function outputs *true* when a match (based on distances) is found between the incoming time series and the training time series. Otherwise, the prediction is postponed until a user-specified time threshold is achieved.

- **Optimization of the time vs quality trade-off:** This work tends to consider only the earliness of the prediction at the expense of the quality of the prediction.

[1] In contrast to the *eager* machine learning algorithms which try to construct a general function out of the training observations, Case-Based-Reasoning (CBR) is considered as a *lazy* learning algorithm that is limited to store the training observations and map the descriptions of the retrieved similar cases to the target problem. The CBR method is a general case of the Instance-Based learning methods.

- **Adaptability:** This approach is adaptive since the prediction is made based on the majority class label associated to the most similar cases given an incoming fault.

- **Myopia:** The decision making process is myopic in this approach since at each time step, the system decides to make a prediction if a match is found, or to delay the prediction for the next time step.

In this particular application, the earliness is obtained thanks to measuring the similarity between faults that have typical shapes in a factory system. In addition, the space of all possible faults is finite which leads to easily perform early predictions. However, this does not necessarily guarantee reliable predictions when similar shaped time series correspond to different faults.

4.1.3 TCF (2006)

Another example showing the need to make early predictions is in traffic detection for network security and traffic engineering. In [13], a technique for Traffic Classification on the Fly (TCF) is suggested. The goal in this approach is to identify the application associated with a traffic flow as early as possible in order to detect intrusion or malicious attacks, forbidden or new applications, etc. A flow describing a specific application is composed of N packets[1]. Since mining the data in each flow packet is manifestly not feasible (because of the privacy issues, complex encryption techniques, the high speed of links and transfer, the high storage and computational complexities, etc.), a flow is represented by the sizes of its first P packets (where P is a specified parameter). The TCF method is performed in two steps: the offline and the online. In the offline step, a clustering using the K-means algorithm together with the Euclidean distance is performed to construct clusters containing each similar applications. Then, each cluster is analyzed to identify the set of the applications with which it is composed. In the online step, when a new flow is being acquired, P packets should be gathered before making any prediction. Later, when the P packets are received, the incoming flow is represented by the sizes of its P packets and is assigned to the most near cluster (the cluster giving the minimum distance of its centroid against the incoming flow). Finally, a flow is labeled with the application that is the most common in the cluster.

[1] In traffic flow, network flow or packet switching networks, a packet is the unit of data that is sent from a particular source to a particular destination on the Internet. The Transmission Control Protocol (TCP) layer of TCP/IP is in charge of dividing the message (e.g. a file) to be sent into *chunks*. Over each chunk, a numbered packet is built including the Internet addresses of the source and destination in addition to information about the connection settings)

Discussion

- $\mathcal{T}$**rigger function:** The prediction is triggered once P packets of the incoming flow are received. In other words, when a user-specified time threshold is achieved.
- **Optimization of the time vs quality trade-off:** the empirical study conducted in this work, showed that an application is accurately identified by observing only the five first packets of a flow. However, this does not guarantee and generalize by no means the reliability of the prediction.
- **Adaptability:** This approach is not adaptive since the number of packets is fixed the same for all applications. Furthermore, it still looks odd to represent the flows by the sizes of its packets although it is understandable that data of packets are not accessible (mainly because confidentiality).
- **Myopia:** An incoming flow is immediately associated to an application once the required number of packets is met. This makes the decision process myopic since, at each time step, the prediction is postponed without no knowledge about the time one should wait before making a prediction.

In the TCF approach [51], the main objective is to predict the label of an incoming yet incomplete traffic flow. The proposed approach allows one to make early, but not reliable predictions. In fact, the authors use the number of packets to describe the data which may be not an appropriate representation of the traffic flow. In addition, there is no insight of how the shortest number of packets is determined in order to obtain reliable predictions.

4.1.4 SCR/GSDT (2008)

Until the work of Xing et al. [106], the early classification problem has not been explicitly defined. Instead, heuristics have been used to label the incoming yet incomplete time series by fixing either the time or the quality of the prediction without putting forward that there is two contradictory objectives. In [106], Xing et al. are the first to define the early classification problem as the challenge of finding the best compromise between the accuracy and the time of the prediction. This is accomplished by predicting the label of incomplete sequences as early as possible while maintaining an expected accuracy. Here, early classification is applied over temporal symbolic sequences.

Two methods are suggested to make early predictions: a Sequential Classification Rule (SCR) method and a Generalized Sequential Decision Tree (GSDT) method. Based on a set of features extracted from the training data set, association rules and decision trees are built. Afterwards, when a sequence s_t is arriving, a search in all branches and against all rules is conducted until a match[1] is found. Thereafter, s_t is immediately labeled once the expected accuracy is achieved.

Discussion

- $\mathcal{T}$**rigger function:** The prediction is triggered once a user expected accuracy is achieved.
- **Optimization of the time vs quality trade-off:** This work represents a worthwhile contribution for the early classification problem since it is the first to define the problem as a trade-off between the time and the quality of the prediction. However, early prediction is conditioned on a user expected accuracy even though feature selection step may emphasizes the need to find early and distinctive features. Globally, it is visible that there is an effort to consider both time and accuracy when deciding. But this still tends to consider only the quality of the prediction and is just as unconcerned with the question of optimizing the time vs quality of the prediction trade-off.
- **Adaptability:** The two suggested approaches provide adaptive predictions since the incoming sequence s_t is paired off with all rules (respectively a search in all branches) before making a prediction.
- **Myopia:** The decision making is myopic since an imminent decision is made once a user-specified accuracy threshold is met.

In this work, the authors mainly focused on providing efficient and interpretable features selection for early classification of temporal symbolic sequences. The proposed approaches do not address the issue of finding the optimal trade-off between the time and quality of prediction. Rather, they make a myopic decision when a user-specified accuracy threshold is achieved. Conditioning achieving early prediction by setting a user accuracy parameter may reduces this trade-off to a single objective which consists on controlling the quality of the prediction regardless earliest is the decision or not. In other words, the

[1] Particularly, an incoming sequence s_t is considered to match a sequential classification rule R if the features in s_t appear in the same order as in R.

obtained time to make a decision is not optimized which does not guarantee the earliness of predictions when a user accuracy is expected.

4.1.5 ECTS (2009)

So soon after the introduction and the attempt to define the early classification problem and since symbolic approaches in [106] were not appropriate for numeric time series, Xing et al. [107] suggested a new method called Early Classification on Time Series (ECTS). In this approach, the so-called Minimum Prediction Length (MPL) is introduced, and is estimated using a 1-nearest neighbor (1-NN) classifier. Specifically, the MPL is the smallest length of an incoming time series for which i) its nearest neighbors remain the same after acquiring new measurements and ii) the quality of the prediction remains the same as that obtained with full length time series.

Furthermore, as the nearest neighbors approaches are *lazy*, the authors added a training step to determine the MPLs. A hierarchic clustering is performed, then the MPLs are computed for each time series according to their cluster membership.

The idea is to explore the stability of the relationship between the nearest neighbors explored in the *complete* space (the space of complete time series) and the nearest neighbors formed in the subspace containing the incoming time series $\mathbf{x}_t$. Once both stability and a user expected accuracy are achieved, $\mathbf{x}_t$ is labeled based on the majority class in the nearest cluster.

Discussion

- **$\mathcal{T}$rigger function:** The $\mathcal{T}rigger$ function is activated depending on the estimation of the earliest time at which the prediction $h_t(\mathbf{x}_t)$ is equal to the one that would been made if the complete time series $\mathbf{x}_T$ was known. This is done by using the nearest training time series.

- **Optimization of the time vs quality trade-off:** There is no formal optimization of the trade-off between the earliness and the quality of the prediction. The optimal time for making a prediction is strongly based on finding the nearest neighbor(s).

- **Adaptability:** The decision made given $\mathbf{x}_t$ is specific to $\mathbf{x}_t$ since it is specific to the cluster containing $\mathbf{x}_t$.

- **Myopia:** The decision making process is myopic in this approach.

The ECTS approach does not automatically optimize the early classification trade-off, but, it determines the optimal time for classifying an incoming time series based on the MLP(s) of the nearest neighbor(s).

4.1.6 EDSC (2011)

Aware of how important this issue is, Xing et al. continue to take the lead in bringing forward new solutions to the early classification problem.

They suggested the Early Distinctive Shapelet Classification (EDSC) [108] method in order to provide efficient and interpretable features selection to guide early classification. They consider that some locally distinctive features that exclusively represent a subset of time series in one class (defined as shapelets) may be important to make early prediction besides their interpretability. In the training step, EDSC extracts a set of shapelets $\{f_i = (s, \delta, y),\ i \in \{1, N\}\}$ and learns their associated distance thresholds δ (see Section 3.3.2 for a detailed description of the shapelet). Then, it ranks all the shapelets in utility considering both their earliness and accuracy and takes those of the highest utilities that cover all the training data set. In the prediction step, an incoming time series $\mathbf{x}_t$ is labeled when a match with a shapelet is found. $\mathbf{x}_t$ is considered to match a shapelet f if $d(s, \mathbf{x}_t) \leq \delta$, where d is the distance between the shapelet sequence s and the incoming time series $\mathbf{x}_t$.

Besides, it is noteworthy that such approach is only valuable in some domains where time series have typical shapes which are common for one class such as in medical and health informatics. Yet, when time series are very volatile and there is no distinctive features representing each class, there is no interest in applying such a method.

Discussion

- **Trigger function:** The $\mathcal{T}rigger$ function outputs *true* when the incoming time series $\mathbf{x}_t$ matches a shapelet $f = (s, \delta, y)$.
- **Optimization of the time vs quality trade-off:** There is no optimization of the earliness vs quality trade-off. Th optimal time for making a prediction is based on finding a match with a shapelet. This strongly relies on the definition of the training shapelets and the used distance.
- **Adaptability:** This approach is semi-adaptive because of the consideration of the incoming time series in the decision making, but at the same time it is heavily dependent on the used distance. Otherwise, the distances thresholds which are

specific for each shapelet control the degree of adaptability. For instance, a large distance threshold δ may make the same match with a shapelet f for totally two different sequences.

- **Myopia:** at each time step, the system decides either to make or to postpone the prediction. This makes the decision making process myopic in the EDSC approach.

The EDSC approach does not explicitly optimize the trade-off between the earliness and the accuracy of the prediction. Furthermore, it myopically decides at each time step to make a prediction or to wait for an additional measurement.

4.1.7 CCII (2012)

Different from previously presented works, a method based on probabailistic forcasting is suggested in [6, 78] for Classifying with Confidence from Incomplete Information (CCII).

In this approach, the early classification task is considered as a classification from incomplete data. Besides, the objective is to make reliable prediction.

The concept of the reliability stands for that the probability of assigning a label to a given incomplete data would be the same as the one of assigning a label to a given complete data. Therefore, the idea is to label incomplete data only when a user reliability threshold is met.

This is achieved by suggesting an incomplete decision rule able to label an incomplete time series $\mathbf{x}_t$ by imputing a distribution over the missing data conditioned on the incoming time series $\mathbf{x}_t$ and the training data. Here, the complete time series are modeled as random variables whose distributions are dependent to the incoming yet incomplete time series and assumed to be *i.i.d* with the training data set. Based on these distributions, the reliability bound on the classifier's prediction is estimated at each time step. Thereafter, an incoming time series $\mathbf{x}_t$ is labeled when the estimated reliability exceeds a specified threshold.

Discussion

- **$\mathcal{T}$rigger function:** The $\mathcal{T}rigger$ function outputs *true* when, with a probability at least equal to a user specified threshold, the assigned label obtained using the incomplete time series will match the one that would be assigned using the complete time series.

- **Optimization of the time vs quality trade-off:** In this approach, there is no explicit penalty for predicting too late. Indeed, the authors focus on making

reliable prediction instead of searching a good balance between the earliness and the quality of the prediction.

- **Adaptability:** The estimation of the reliability bound conditioned on the incoming time series makes this approach adaptive.
- **Myopia:** The prediction is made when a reliability-specified threshold is achieved. This condition needs to be tested at each time step rendering the decision making process myopic in this approach.

In this work, the authors propose a probabilistic forecasting approach in order to maximize the reliability of the prediction: the probability that the early prediction using incomplete time series leads to the same classification result as the classification of the complete time series. This approach does not take the earliness explicitly into account. It instead evaluates the reliability of the current prediction in order to make a decision. In addition, this procedure is myopic in that it does not look further than the current time to decide if it a prediction should be made.

4.1.8 ECRO (2013)

The objective of ensemble methods is to combine the decisions of individual (weak) classifiers in order to obtain better predictive performances than could be obtained from any of the individual classifiers. In the context of early classification, the idea is the following: for a given incoming time series $\mathbf{x}_t$, an ensemble of independent classifiers which are trained on the same time intervals[1], estimate the class label of $\mathbf{x}_t$.

The *agreement* of the classifiers about the prediction given $\mathbf{x}_t$ is then translated into a confidence value whereby a decision can be made (see Figure 4.1). In this case, an agreement rule should be finely defined in order to make reliable predictions (e.g. weighting the classifiers and assign greater weights to those which err the least).

The proposed approach [55] (ECRO) is a particular case of the general description we gave above, since: i) the number of the classifiers over each interval is fixed to 2 and ii) the agreement rule considers that the classifiers agree to make a decision if they output the same label. For instance, each classifier makes a decision about the label of $\mathbf{x}_t$ based on the t observed measurements in the first time interval. If both classifiers

[1] The training time series are divided into k time intervals. A set of classifiers are trained independently on each interval. For instance, n classifiers are trained over the first interval. Then, n' classifiers which are independent from the first ones are trained over the second interval, etc.

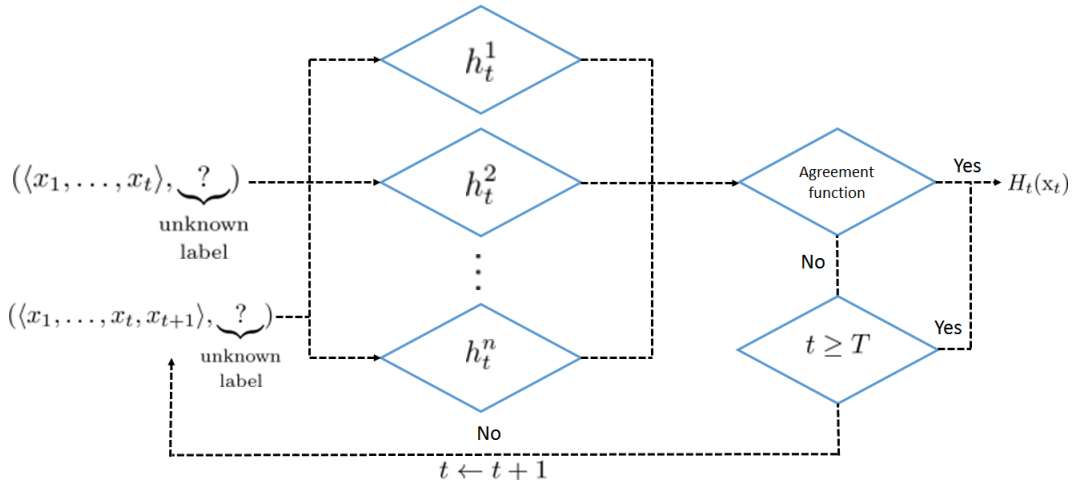


Figure 4.1: The incoming time series $\mathbf{x}_t$ is labeled once the ensemble classifiers are in agreement. The function $H_t(\mathbf{x}_t)$ could be a weighted sum of the individual classifiers outputs. It could correspond also to the function of the best classifier (the one that gives correct predictions most often).

are enough confident about their predictions (if they output the same label[1]), $\mathbf{x}_t$ is then labeled, otherwise the task is passed to the two next classifiers with new available time interval. This idea of triggering or postponing the prediction at each step is close to the notion of the reject option [30] where a user-specific threshold should be tuned in order to reject/accept a label.

Discussion

- **Trigger function:** The *Trigger* function outputs *true* if the classifiers confirm their agreement on the output label. The agreement function outputs a confidence score whereby a decision could be made.
- **Optimization of the time vs quality trade-off:** This approach does not explicitly take into account the earliness when making a decision. It does not trade off between the earliness and the quality of prediction.
- **Adaptability:** In this approach, the incoming time series $\mathbf{x}_t$ is taken into account when making a decision. This is done through the classifiers predictions which are specific for the given input .

[1] The drawback in this approach is that when the classes are imbalanced in a classification task, it may happen that the classifiers tend to output the majority class with a high accuracy when few measurements are observed.

- **Myopia:** When an agreement of the classifiers about the label of $\mathbf{x}_t$ is met, $\mathbf{x}_t$ is labeled. This condition is tested at each time step making the decision procedure myopic.

The interesting property in this method is that without requiring a user expected accuracy, the classifiers are able to stop acquiring additional information and provide early predictions. However, on the other hand, there is no penalty for reporting the rejection which in turn shows that there is no explicit consideration of the time in the decision making process. Furthermore, the voting procedure considered in this approach may bring into question the reliability of the prediction when the classes are imbalanced.

4.1.9 MEDSC-U (2014)

Recently, a Modified EDSC with Uncertainty estimates (MEDSC-U) method has been suggested. MEDSC-U extends the Early Distinctive Shapelet Classification (EDSC) method [108] to estimate the temporal uncertainty associated with the early prediction while yielding interpretable results. In this approach, it is highlighted that the concept of confidence is relevant when making early classification because incomplete information often entail some uncertainty.

By contrast with EDSC that stops measuring additional information and labels an incoming time series $\mathbf{x}_t$ when a match is found against a shapelet f, MEDSC-U relies instead on the estimation of uncertainty for each class at each time step to decide when to stop or postpone the prediction. Thus, an incoming time series $\mathbf{x}_t$ is labeled, at each time step, with the class that has the minimum uncertainty at that time. The prediction is triggered once a user-specified uncertainty threshold is met.

Discussion

- **$\mathcal{T}$rigger function:** The $\mathcal{T}rigger$ function outputs *true* when the system is confident enough about the prediction it provides. The confidence threshold is a user-specified parameter.

- **Optimization of the time vs quality trade-off:** There is no optimization of the trade-off between the time and the quality of the prediction. The requirement of uncertainty constraint may even tend to delay the time of the prediction against the EDSC method [108].

- **Adaptability:** This approach is adaptive since the distances measured against all the shapelets and the uncertainty measure involve the incoming time series $\mathbf{x}_t$.
- **Myopia:** When a user-specified confidence threshold is met, a prediction is immediately made without looking further than the current time to make a decision.

In this work, the authors provide an estimation of the temporal uncertainty associated with the decision. Early results are obtained by evaluating the estimated uncertainty against a user-specified confidence threshold. From this, there is no optimization of the time vs quality trade-off. Instead, depending on the requirements on uncertainty, the earliness of prediction is not explicitly taken into account when making a decision.

4.1.10 Other related works

Beyond these seminal works, other approaches have been suggested for handling the early classification problem. These methods are doubtlessly interesting regarding the raised problems (e.g. the use of shapelets is relevant in making early predictions in the realm of biomedical data), but do not bring any hypothetical solution or definition for the early classification problem. Rather, they addressed the problem under other constraints. For example, in [46, 47, 49], the focus is on making early classification on multivariate numeric time series. Commonly, the idea of these methods consists in extracting multivariate shapelets from all dimensions of the time series. Afterwards, these latter are used together with distance thresholds and utility scores to make early predictions. In [59], a new boosting algorithm based on weight propagation and the standard exponential loss technique is suggested for handling the multi-class early classification problem. Dainotti et al. [34], interested in the early classification of network traffic, they applied ensemble classifiers to improve the prediction accuracy through experimenting a multiple strategies for combining the classifiers. He et al. [56] raised the problem of making early classification on imbalanced multivariate time series. They suggested to use a set of classifiers over multiple subsets. These subsets are obtained after applying an under-sampling [66] method over the imbalanced training data set. Lin et al. [68] are interested in making reliable early classification on multivariate time series with numerical and categorical attributes. Their early prediction method being based on shapelets, they mainly focused on designing a technique for feature extraction based on the concept of equivalence classes mining. In [7], Antonucci et al. leverage the potency of the so-called credal classifiers[1], and imprecise Hidden Markov models (iHMMs) [70] to perform early

[1]The notion of credal classifier relays on the new idea of classification based on the possibility to suspend the prediction. Credal classifiers are different from the traditional classifiers with a rejection

classification.

4.2 Discussion

Properties	Methods/References								
	BIBL [84]	EFC [20]	TCF [13]	SCR/GSDT [106]	ECTS [107]	EDSC [108]	CCII [78]	ECRO [55]	MEDSC-U [48]
$\mathcal{T}rigger$ function	✗	✓	✓	✓	✓	✓	✓	✓	✓
trade-off optimization	✗	✗	✗	✗	✗	✗	✗	✗	✗
Adaptability	✗	✓	✗	✓	✓	✓	✓	✓	✓
Decision making myopia	✗	✗	✗	✗	✗	✗	✗	✗	✗

Table 4.1: Properties of early classification of time series methods. This list of methods is non exhaustive and shows only methods discussed in the state-of-the-art presented in Section 4.1.

Table 4.1 summarizes the above discussed early classification methods according to the different properties. To the best of our knowledge, there is not yet any early classification method that succeeds to simultaneously provide these four requirements.

Globally, it is remarkable that even if the earliness of the decision is mentioned as a motivation in the main state-of-the-art works, the decision procedures themselves do not take it explicitly into account. They instead evaluate the confidence or reliability of the current prediction(s) in order to decide if the time is ripe for prediction, or if it seems better to wait one more time step. In addition, the procedures are myopic in that they do not look further than the current time to decide if it a prediction should be made.

Another important aspect when making online decisions is when delaying the decision is costly. Indeed, information can be gained by waiting for more evidences to arrive, thus

option that decide to suspend the prediction according to a threshold, more details can be found in [112].

helping to make better decisions that incur lower misclassification costs, but, meanwhile, the cost associated with delaying the decision generally increases, rendering the decision less attractive. Early classification requires then to solve an optimization problem combining two types of competing costs:

- the gain of information that incurs lower misclassification costs that can be expected if one delays the decision,
- the raising cost of such a delay

While it is straightforward to involve costs associated to delaying decisions, which is a crucial factor in many applications, most of the early classification methods do not explicitly take it into account. Instead, they focus on making reliable prediction from incomplete time series.

Summary

In this bibliographical chapter, we have discussed the main state-of-the-art early classification methods according to four requirements: (i) the $\mathcal{T}rigger$ function that decides when to output a prediction, (ii) the optimization criterion that balances the quality vs earliness trade-off (iii) the adaptability to data by taking into account the incoming time series, and finally, (iv) the non-myopic characteristic.

A conclusion one can draw on the basis of this study is that there is no early classification method that efficiently optimizes the quality vs earliness trade-off. Rather, most systems balance the quality vs the earliness trade-off by either thresholding one of the two objectives or making some conditions on the prediction triggering such as the confidence on the prediction.

In this thesis, we suggest a novel early classification method that defines the problem as a cost-sensitive decision making problem. Our method optimizes the quality vs earliness trade-off by combining two types of costs: (i) the cost of misclassification that controls the quality of the prediction, and (ii) the cost of delaying the decision which controls the earliness of the prediction. We present our new formalization in Chapter 5 followed by two meta-learning mechanisms that achieve adaptive and non-myopic decisions. The robustness of these two approaches is tested on multiple synthetic and real data sets in Chapter 5. We finally conclude our main contributions and present future research directions in Chapter 7.

Chapter 5

Time series early classification: Cost-sensitive online decision making

Introduction

In the previous chapter, we have discussed the main state-of-the-art early classification approaches, and concluded that the proposed decision procedures in those approaches do not explicitly take into account the cost of delaying the decision and do not optimize the trade-off between the gain of information and the cost it is needed to perform it.

However, in some situations, where it is essential to make timely decision in absence of complete knowledge, we argue that, as soon as decision making processes are constrained by time, the cost of delaying the decision seems to be intuitively a crucial factor that should be accounted for in decision making procedures. In this setting, we suggest that making early predictions requires to solve an optimization problem combining two types of competing costs:

1. the gain of information that can be expected if one delays the decision yielding lower misclassification costs, and
2. the raising cost of such a delay.

From this, our objective is to make optimal decisions while meeting time and quality constraints during the prediction process. Our idea is thus to formalize the problem

of early classification of time series as a cost-sensitive online decision making problem involving the two costs: (i) the misclassification cost that is better if one delays the decision, and (ii) the cost of such a delay which is better if decision is made quickly.

As major contribution, we suggest a new optimization criterion that considers both aspects of early predictions, and, furthermore, it allows one to estimate, and update if necessary, the future optimal time step for making a decision. This new online decision strategy provides a base for a new general formalization of the problem, called **ECONOMY** standing for **E**arly **C**lassification for **O**ptimized and **NO**n-**MY**opic online decision making.

The originality of our work is threefold. First, an optimization criterion that automatically balances the expected gain in the classification cost in the future with the cost of delaying the decision. Second, this criterion leads to two meta-algorithms that are adaptive, in that, the properties of the incoming time series are taken into account to decide what is the optimal time to make a prediction. And third, in contrast to the usual sequential decision making techniques, the algorithms presented are non-myopic. In effect, at each time step, they compute what is the optimal expected time for a decision in the future, and it is only if this expected time is the current time that a decision is made. A myopic procedure would only look at the current situation and decide whether it is time to stop asking for more data and make a decision or not.

This chapter is organized as follows. Section 5.1 presents closely related works on cost-sensitive online decision problems. In Section 5.2, we present ECONOMY, a new formalization criterion for making optimal and cost-sensitive classifications from incomplete time series. The following sections then describe two meta-algorithms that implement the generic criterion: ECONOMY-K (Section 5.4) is based on a clustering step. A more elaborated method, ECONOMY-γ, is detailed in Section 5.5. The detailed experimental study is reported in Chapter 6.

5.1 State-of-the-art

The trade-off between the gain of information that is expected to incur lower misclassification costs, and the decision delaying cost is a classical optimization problem that has been known for decades and has historical roots in fields such as sequential decision making, optimal statistical decisions, cost-sensitive learning, time constrained sequential

decision making, etc., but numerous new applications in medicine, electric grid management, automatic transportation, and so on, give a new impetus to research works in this area.

5.1.1 Sequential decision making

In many scenarios involving sequential decision making and optimal statistical decisions [12, 36], one technique especially has gained a wide exposition: Wald's Sequential Probability Ratio Test [102]. The task is to classify a sequence of measurements $\mathbf{x}_t^i$ into one of two possible classes -1 or $+1$. The likelihood ratio:

$$R_t = \frac{P(\langle x_1^i, \ldots, x_t^i \rangle \mid y = -1)}{P(\langle x_1^i, \ldots, x_t^i \rangle \mid y = +1)}$$

is computed and compared with two thresholds set according to the required error of the first kind α (*false positive error*) and error of the second kind β (*false negative error*). One difficulty lies in the estimation of the conditional probabilities $P(\langle x_1^i, \ldots, x_t^i \rangle \mid y)$. And, furthermore, there is no explicit reference to the cost of delaying the decision in the choice of α and β.

Another core of related works on online decision making problem is classification under resource constraints. These resource constraints are different depending on the application domain and strongly impact the decision making procedure.

5.1.2 Classification under resource constraints

In many application domains, including medical diagnosis, IT security, surveillance, etc., the decision making process is often constrained by some resources such as small available memory sizes, restricted budgets or low response time. For example, in medical applications, the cost of additional tests per patient should not exceed some initially fixed budget. In such a situation, the goal is to make a decision that maximizes the performance of the prediction subject to budget constraints.

Particularly, in the extensive literature on the problem of classification under budget constraints, different strategies have been proposed. In works such as [22, 26, 100] the data acquisition costs are incorporated into the decision process through detection cascades, a well known technique in object detection framework for computation cost reduction. The average total cost is reduced because cheap (resp. rapidly accessible)

information are used at first and more costly (resp. slowly) information are used in later stages.

The fundamental difference between the literature on classification under budget constraints and early classification of time series is the order considered when acquiring the data and the optimization of the global cost across time. Typical research works proposed for sequential classification under budget constraints have the possibility to order the input data as a part of the optimization task. This commonly brings a crucial enhancement of the prediction quality while reducing the average cost, as it allows to reasonably use resources. However, in our context, early classification of time series is constrained by an ordered set of measurements and timely decision making.

In this work, we seek to make optimal decisions while meeting time and quality constraints during the prediction process. To do so, we propose to formalize the problem of early classification of time series as a cost-sensitive online decision making problem.

5.2 New formalization of the problem

As motivated before, it is crucial to explicitly take into account the cost associated with delaying the decision which is an essential requirement for decision making in real world system applications.

The question is then to learn a decision procedure in order to determine the earliest time t^* at which a new incoming time series $\mathbf{x}_{t^*} = \langle x_1, x_2, \ldots, x_{t^*} \rangle$ can be optimally labeled. To do so, we associate a cost with the prediction quality of the decision procedure and a cost with the time step when the prediction is finally made:

- We assume that a ***misclassification cost function*** $\mathrm{C}_t(\hat{y}|y) : \mathcal{Y} \times \mathcal{Y} \longrightarrow \mathbb{R}$ is given, providing the cost at time t of predicting $\hat{y}$ when the true class is y.

- Each time step t is associated with a real valued ***time cost function*** $\mathrm{C}(t)$ which is non decreasing over time, which means that it is always more costly to wait for making a prediction. Note that, in contrast to most other approaches, this function can be different from a linear one, reflecting the peculiarities of the domain. For instance, if the task is to decide if an electrical power plant must be started or not, the waiting cost rises sharply as the last possible time approaches.

We can now define a cost function f associated with the decision problem.

$$f(\mathbf{x}_t) \;=\; \sum_{y \in \mathcal{Y}} P(y|\mathbf{x}_t) \sum_{\hat{y} \in \mathcal{Y}} P(\hat{y}|y, \mathbf{x}_t)\, \mathrm{C}_t(\hat{y}|y) \;+\; \mathrm{C}(t) \tag{5.1}$$

This equation corresponds to the expectation of the cost of misclassification after t measurements have been made, added to the cost of having delaying the decision until time t. The optimal time t^* for the decision problem is then defined as :

$$t^* \;=\; \underset{t \in \{1,\ldots,T\}}{\mathrm{ArgMin}}\; f(\mathbf{x}_t) \tag{5.2}$$

However, this formulation of the decision problem requires that one be able to compute the conditional probabilities $P(y|\mathbf{x}_t)$ and $P(\hat{y}|y, \mathbf{x}_t)$. The first one is unknown, otherwise there would be no learning problem in the first place. The second one is associated with a given classifier, and is equally difficult to estimate.

Short of being able to estimate these terms, one can fall back on the expectation of the cost for *any time series* (hence the function now denoted $f(t)$):

$$f(t) \;=\; \sum_{y \in \mathcal{Y}} P(y) \sum_{\hat{y} \in \mathcal{Y}} P(\hat{y}|y)\, \mathrm{C}_t(\hat{y}|y) \;+\; \mathrm{C}(t) \tag{5.3}$$

From the training set $\mathcal{S}$, it is indeed easy to compute the a priori probabilities $P(y)$ and the conditional probabilities $P(\hat{y}|y)$ which are nothing else than the confusion matrix associated with the considered classifier. One gets then the optimal time for prediction as:

$$t^* \;=\; \underset{t \in \{1,\ldots,T\}}{\mathrm{ArgMin}}\; f(t)$$

This can be computed before any new incoming time series, and, indeed, t^* is independent on the input time series. Of course, this is intuitively unsatisfactory as one could feel, regarding a new time series, very confident (resp. not confident) in his/her prediction way before (resp. after) the prescribed time t^*. If such is the case, it seems foolish to make the prediction exactly at time t^*.

Now, based on the general formulation defined in Equation 5.1, that we call **ECONOMY**, standing for **E**arly **C**lassification for **O**ptimized and **NO**n-**MY**opic online decision making[1], our objectives in the next sections are to extend ECONOMY in order to

[1] It is called such because it gives the ground to obtain computable algorithms that give online,

make **online**, **adaptive** and **non-myopic** decisions.

In the following, we start by presenting our methodology to achieve these objectives. Then, we propose two different methods that extend the global formulation ECONOMY following the proposed methodology.

5.3 Our methodology

The general formalization, proposed in Section 5.2, makes it possible to express the expected cost of a decision after t time steps as shown in Equation 5.1, repeated below:

$$f(\mathbf{x}_t) \;=\; \sum_{y \in \mathcal{Y}} P(y|\mathbf{x}_t) \sum_{\hat{y} \in \mathcal{Y}} P(\hat{y}|y, \mathbf{x}_t)\, \mathrm{C}_t(\hat{y}|y) \;+\; \mathrm{C}(t)$$

However, this formulation of the problem does not readily yield a method for finding, online, the optimal decision time. First, it would require to compute all the decision costs until time T before knowing what is $t^\star$, clearly defeating the purpose of the approach. Second, the terms $P(y|\mathbf{x}_t)$ and $P(\hat{y}|y, \mathbf{x}_t)$ are difficult to estimate on a single time series. This requires that some generalization over the space of possible time series takes place.

To overcome the difficulties of computing $P(y|\mathbf{x}_t)$ and $P(\hat{y}|y, \mathbf{x}_t)$, our idea is to capture typical evolutions in the complete training time series using some technique accounting for the incoming time series. To do so, we first propose to (i) segment the complete training time series into coherent groups that we note $\{G_k\}_{1 \leq k \leq K}$. Based on these groups, the idea is to substitute the term $P(\hat{y}|y, \mathbf{x}_t)$ by the computable term $P(\hat{y}|y, G_k)$ and add another term to take into account the incoming time series $\mathbf{x}_t$. Then, (ii) based on this information, it is possible to define a cost function that should be able to estimate the expected cost for each future time step given the incoming yet incomplete time series.

5.3.1 Segmentation

The idea behind segmenting the complete time series is to leverage the complete information and the different behaviors identified in the training data set to make adaptive estimate of the future costs given an incoming time series $\mathbf{x}_t$.
Specifically, the complete training time series should be segmented into coherent groups $\{G_k\}_{1 \leq k \leq K}$ using a specific segmentation method. These groups will be used later to compute the terms $P(\hat{y}|y, G_k)$ which are simply confusion matrices computed over each

adaptive and non-myopic decisions.

group. However, the construction of these groups should obey two constraints as well as possible:

1. Different groups should correspond to different confusion matrices.

2. Groups should contain similar time series, and be dissimilar to other groups.

In constraint (1), a confusion matrix $P(\hat{y}|y, G_k)$ is computed over time series in a group G_k using an already learned classifier. The confusion matrices computed over each group should be different as much as possible in order to discriminate the cost between groups. To achieve this, the objective is to form different groups that are different as much as possible. In words, the segmentation should help to achieve a supervised task.
In constraint (2), similar time series, grouped using a specific similarity function, should belong to the same group and should be different from time series in other groups. This way, an incoming time series will generally be assigned markedly to one of the groups. As such, the segmentation should help to make the cost function adapted to the incoming time series.

5.3.2 Estimate the expected costs for future time steps

The second idea we propose in order to overcome the necessity to compute $f(\mathbf{x}_t)$ for all $t \in \{1, \ldots, T\}$ is to compute in advance, at time t, the expected costs of decision for all future time steps. This is possible through computing confusion matrices $P_t(\hat{y}|y, G_k)$ on each group G_k and at each time step t, then estimating a membership between an incoming time series $\mathbf{x}_t$ and each group, which provides information about potential futures.

Specifically, given that, at time t, $T - t$ data points are still missing in the incoming time series $\mathbf{x}_t$, it is possible to compute the expected cost of classifying $\mathbf{x}_t$ at each future time step $\tau \in \{0, \ldots, T - t\}$.
Now, assume that there is a function f_τ that estimates the expected cost for each of the remaining τ time steps using the complete information obtained based on the formed groups. This allows one to forecast the optimal horizon $t^\star$ for the classification of the input time series $\mathbf{x}_t$:

$$t^* = t + \underset{\tau \in \{0,\ldots,T-t\}}{\text{ArgMin}} \; f_\tau(\mathbf{x}_t) \tag{5.4}$$

Of course, these expected costs given by $f_\tau(\mathbf{x}_t)$, where $\tau \in \{0, \ldots, T - t\}$ and the estimated optimal horizon $t^* = t + \tau^\star$, where $\tau^\star = \text{ArgMin}_{\tau \in \{0,\ldots,T-t\}} f_\tau(\mathbf{x}_t)$, can be

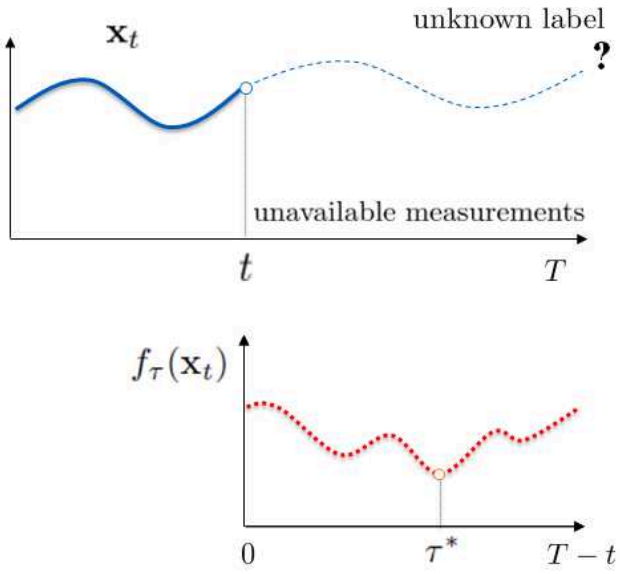


Figure 5.1: The first curve represents an incoming time series $\mathbf{x}_t$. The second curve represents the expected cost $f_\tau(\mathbf{x}_t)$ given $\mathbf{x}_t$, $\forall \tau \in \{0, \ldots, T-t\}$. It shows the balance between the gain in the expected precision of the prediction and the cost of waiting before deciding. The minimum of this trade-off is expected to occur at time $t+\tau^\star$. New measurements can modify the curve of the expected cost and the estimated $\tau^\star$.

re-evaluated when a new measurement is made on the incoming time series. Figure 5.1 illustrates this idea.

5.3.3 Decision policy

Our decision policy is defined as the following. At any time step t, if the optimal horizon $\tau^\star = 0$ and for any $\tau > 0$, $f_\tau(\mathbf{x}_t) > f_0(\mathbf{x}_t)$, then the sequential decision process stops and a prediction is made about the class of the input time series $\mathbf{x}_t$ using the classifier h_t: $\hat{y} = h_t(\mathbf{x}_t)$ (here, a set of classifiers $\mathcal{H} = \{h_t\}_{1 \leq t \leq T}$ are used where each classifier h_t is trained over the training time series trimmed to their first t components. Other type of classifiers conceived to predict the class of time series of any length can be used).
Other decision policies can be used such as a less stronger version of the rule we propose for deciding about the optimal time. For instance, in some situations, there may be a very small cost difference $\epsilon > 0$ between two minima that can be distant in time, $f_0(\mathbf{x}_t) = f_{\tau^\star}(\mathbf{x}_t) + \epsilon$. In such a situation, earlier decisions can be made at the expense of small reasonable loss of precision in the cost estimation. Figure 5.2 better illustrates this situation.

Globally, the decision policy is closely related to the application domain and is generally fixed by an expert.

Returning to the general framework outlined for the early classification problem in Section 2.7.3, the proposed function that triggers a prediction for the incoming time series is given in Algorithm 4:

Algorithm 4 Proposed $\mathcal{T}rigger(\mathbf{x}_t, h_t)$ function.

Input:

- $\mathbf{x}_t$, $t \in \{1, \ldots, T\}$, an incoming time series;

1: $\mathcal{T}rigger \longleftarrow false$
2: **for all** $\tau \in \{0, \ldots, T-t\}$ **do**
3: compute $f_\tau(\mathbf{x}_t)$ /* see Equation 5.5*/
4: **end for**
5: $\tau^* = \underset{\tau \in \{0,\ldots,T-t\}}{\text{ArgMin}} \; f_\tau(\mathbf{x}_t)$
6: **if** $(\tau^* = 0)$ **then**
7: $\mathcal{T}rigger \longleftarrow true$
8: **end if**

The proposed algorithm is very simple. Observing an incoming time series $\mathbf{x}_t = \langle x_1, \ldots, x_t \rangle$, the expected cost $f_\tau(\mathbf{x}_t)$ is estimated for each future time step $\tau \in \{0, \ldots T-t\}$. If the estimated optimal decision time $\tau^\star = 0$, the procedure stops and a prediction $h_t(\mathbf{x}_t)$ on the class label of $\mathbf{x}_t$ is made. Otherwise, the algorithm waits for an additional measurement x_{t+1}, unless $t = T$.

5.3.4 Extending ECONOMY

In order to implement our ideas, we propose two different approaches for solving the general formalization ECONOMY following the methodology presented above.

1. First, we present an intuitively alluring approach called **ECONOMY-K**. This approach aims at defining the terms of the cost function f_τ using a segmentation based on a clustering technique.

2. The second approach called **ECONOMY-γ** aims also at defining f_τ. It is more direct and more informed compared to ECONOMY-K because it also uses the information about the class labels when segmenting the complete time series. In addition it involves only one user-parameter.

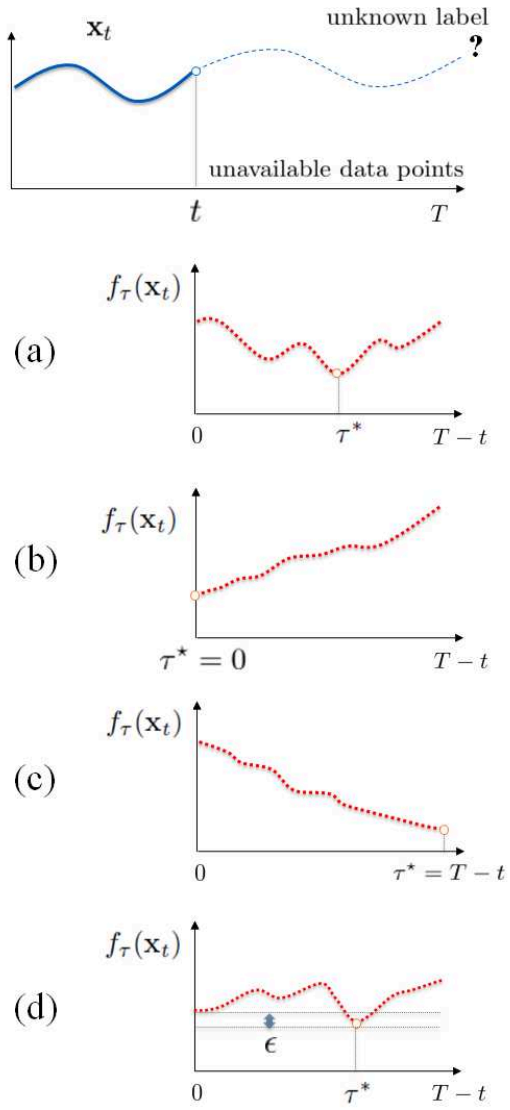


Figure 5.2: An illustrative example of different possible shapes of the estimated costs (impacted by the gain of information and the cost of delaying the decision). In case (a), the cost decreases until $\tau^\star$ since the gain of information incurs lower misclassification costs that compensate the increasing delaying cost. After $\tau^\star$, the increasing delaying cost takes the lead. In case (b), the cost estimation is strongly impacted by a highly increasing delaying cost which leads to an immediate decision, often, at the current time. However, in case (c), the decision is not constrained by a high delaying cost which makes the system tends to wait longer before making a decision, often waiting the last possible time. In case (d), when there is a small cost difference between two distant minima, it is desirable to make an earlier decision.

Before detailing each of the two approaches, we show an example that illustrates an experimental result in order to give insight of the difference between both approaches.

The example in Figure 5.3, illustrated using the synthetic data set (described in detail in Chapter 6), shows the difference in term of the segmentation results between both approaches. We observe that, under the same conditions, the composition of a group of time series obtained by the segmentation used in the ECONOMY-K widely differs from the one obtained by the segmentation used in the ECONOMY-γ approach. Specifically, in Figure 5.3(b), ECONOMY-K uses the K-means clustering technique and the Euclidean distance. Time series are then segmented according to their similarity in shape so that each group contains similar time series and is different from other groups. By contrast, in Figure 5.3(c), ECONOMY-γ uses the information about the class labels to make the segmentation. In this case, time series in each group have different shapes but are described by the same class.

In the following sections, we describe in details each of the proposed approaches.

5.4 ECONOMY-K: Clustering-based early classification approach

5.4.1 Framework

The goal is to estimate the conditional probability $P(\hat{y}|y, \mathbf{x}_t)$ in Equation 5.1, by taking into account the incoming time series $\mathbf{x}_t$, in order to determine the optimal time t^*.

In this approach, the idea is to identify a set $\mathcal{C} = \{\mathbf{c}_k\}_{1 \leq k \leq K}$ of K clusters of time series using a training set so that an (incomplete) input time series $\mathbf{x}_t = \langle x_1, \ldots, x_t \rangle$ can have a membership probability assigned to each of these clusters: $P(\mathbf{c}_k|\mathbf{x}_t)$, and therefore will be recognized as more or less close to each of the prototype time series corresponding to the clusters.

Then, one can compute the confusion matrices $P_t(\hat{y}|y, \mathbf{c}_k)$ associated to each cluster $\mathbf{c}_k$ and each time step t.

The set $\mathcal{C} = \{\mathbf{c}_k\}_{1 \leq k \leq K}$ of clusters should obey two constraints as well as possible.

1. Different clusters should correspond to different confusion matrices. Otherwise, Equation 5.1 will not be able to discriminate the cost between clusters.

2. Clusters should contain similar time series, and be dissimilar to each other, so that an incoming time series will generally be assigned markedly to one of the clusters.

This approach which is called ECONOMY-K is performed into two steps: (i) in the first step, a segmentation of the training time series is achieved through a clustering technique, and (ii) in the second step, the optimal time for making a decision online is estimated following our conceptual methodology described in Section 5.3.

5.4.2 Learning step

During the learning step, the aim is to segment the complete training time series and to compute the confusion matrices $P_t(\hat{y}|y, \mathfrak{c}_k)$ conditionally to each cluster $\mathfrak{c}_k$.

To segment the complete training time series, any clustering technique can be used. Assume that a set $\mathcal{C} = \{\mathfrak{c}_k\}_{1 \leq k \leq K}$ of K clusters is formed using a specific clustering technique and is satisfying the conditions presented above.

Then, for each time step t and each cluster $\mathfrak{c}_k$ (the cluster $\mathfrak{c}_k$ is containing n time series $\mathbf{x}_T^i = \langle x_1^i, \ldots, x_T^i \rangle$, $i \in \{1, \ldots, n\}$, meaning that clusters are built using complete time series), a classifier h_t (already being trained using the training time series trimmed to their t first components) is used to estimate its associated confusion matrix $P_t(\hat{y}|y, \mathfrak{c}_k)$. Note that any supervised learning technique can be used to obtain these classifiers.

In the procedure described above, two choices are made:

1. The first is when performing the clustering directly over complete time series (see Figure 5.4). However, alternatively, one can perform the clustering at each time step t. This gives $K \times T$ different clusters instead of only K clusters if the clustering is done only at time T (Note that the number of clusters may vary from a time step to another, but here for simplicity of notation, we assume that a fixed number of clusters are constructed at each time step t). From this, it is remarkable that the total number of clusters may grow up rapidly when the length of time series are very large. Moreover, clusters built at times $t < T$ may include incomplete information and therefore do not bring novelty for enhancing the prediction. For these reasons, we choose to build K clusters, in all, over complete time series in order to (i) avoid the explosion in the number of clusters when time series are of high dimension, and (ii) ensure maximum information by using complete time series.

2. The second is when only one classifier h_t is learnt at once for all clusters at time step t. Certainly, it is possible that when learning a classifier h_t^k for each cluster c_k^t (if the clustering is performed at each time step) or $\mathfrak{c}_k$ (if the clustering is performed on complete time series at time T), there will be an enhancement of the prediction performances since each classifier is learnt on a specific group of

similar/coherent elements. However, on a numerical level, such setting would cause an exponential explosion in the number of classifiers and handling the problem become computationally intractable.

An approach derived from our work [33] has been proposed by Tavenard and Malinowski [96]. In it, the idea is to remove the clustering step, and all its attendant parameters and choices, by considering each training time series as one cluster. The downside of this approach is its much increased complexity, while it is not obvious that gains in performances are obtained.

Therefore, for the seek of simplicity and caring about making a tractable and effective solution, only K clusters are formed, in all, using the complete training time series, and one classifier h_t is learnt at each time step t over the training time series trimmed to their t first components. Note that clustering the complete time series and learning the classifier h_t are either achieved using a cross-validation procedure if the training set is of small length, or performed over disjoint sets after splitting the training set.

5.4.3 Estimation of the expected cost function f_τ

When a new input time series $\mathbf{x}_t$ of length t is considered, it is compared to each cluster $\mathbf{c}_k$ (of complete time series) and is given a probability membership $P(\mathbf{c}_k|\mathbf{x}_t)$ for each of them as will be detailed in Section 5.4.4. In a way, this compares the input time series to all families of its possible continuations.

Now, given that, at time t, $T-t$ measurements are still missing on the incoming time series, it is possible to compute the expected costs $f_\tau(\mathbf{x}_t)$ of classifying $\mathbf{x}_t$ at each future time step $\tau \in \{0, \dots, T-t\}$, using the set of clusters $\mathcal{C} = \{\mathbf{c}_k\}_{1 \leq k \leq K}$.

The expected cost function f_τ used in the ECONOMY-K approach can then be defined as:

$$f_\tau(\mathbf{x}_t) \;=\; \sum_{\mathbf{c}_k \in \mathcal{C}} P(\mathbf{c}_k|\mathbf{x}_t) \sum_{y \in \mathcal{Y}} P(y|\mathbf{c}_k) \sum_{\hat{y} \in \mathcal{Y}} P_{t+\tau}(\hat{y}|y, \mathbf{c}_k)\, \mathrm{C}(\hat{y}|y) + \mathrm{C}(t+\tau) \tag{5.5}$$

and the optimal decision time for classifying $\mathbf{x}_t$ is:

$$t^* \;=\; t + \underset{\tau \in \{0,\dots,T-t\}}{\mathrm{ArgMin}}\; f_\tau(\mathbf{x}_t)$$

Perhaps not apparent at first, this equation expresses two remarkable properties. First, it is computable, which was not the case of Equation 5.1. Indeed, the terms $P(\mathfrak{c}_k|\mathbf{x}_t)$ and $P(y|\mathfrak{c}_k)$ can now be estimated through frequencies observed in the training data (see Figure 5.4), and the terms $P_{t+\tau}(\hat{y}|y,\mathfrak{c}_k)$ are computed in advance as described in the learning step. Second, the cost depends on the incoming time series because of the use of the probability memberships $P(\mathfrak{c}_k|\mathbf{x}_t)$. It is therefore not computed beforehand, once for all.

5.4.4 Implementation of ECONOMY-K

In order to implement the ECONOMY-K proposed approach, choices have to be made about:

1. The type of used *classifiers* .
2. The *clustering method*, which includes the technique, the distance used, and the number of clusters that are looked for.
3. The method for computing the membership probabilities $P(\mathfrak{c}_k|\mathbf{x}_t)$.

In this thesis, we have chosen to use simple, direct, techniques to implement each of the choices above, so as to clearly single out the properties of the approach through "baseline results". Better results can certainly be obtained with more sophisticated techniques.

Accordingly, (1) we have chosen to use Multi-layer Perceptrons[1] with one hidden layer of $\lfloor t+2/2 \rfloor$ neurons and Naive Bayes classifiers. Then, in order to be able to classify time series of any length t, $1 \leq t \leq T$, where T is the length of the complete time series in the training set, we have chosen to use T binary classifiers, each being learnt over the training time series trimmed to their t first measurements (see Section 3.1.2, for a detailed description of this method). Other methods that are able to deal with input dimension of any length can be used. (2) The clustering over complete time series is performed using K-means with the Euclidean distance. The number K_y of clusters varying for each of the target classes $y=-1$ and $y=+1$ corresponds to the maximum *silhouettes* factor [86]. (3) The membership probabilities $P(\mathfrak{c}_k|\mathbf{x}_t)$ are computed using the following equation:

$$P(\mathfrak{c}_k|\mathbf{x}_t) \;=\; \frac{s_k}{\sum_i^K s_i}, \quad \text{where } s_k \;=\; \frac{1}{1+\exp^{-\lambda\Delta_k}} \tag{5.6}$$

[1] Implemented in WEKA toolkit.

The constant λ used in the sigmoid function s_k is empirically learned from the training set, while $\Delta_k = (\overline{D} - d_k)/\overline{D}$ is the normalized difference between the average of the distances between $\mathbf{x}_t$ and all the clusters, and the distance between $\mathbf{x}_t$ and the cluster $\mathfrak{c}_k$. The distance between an incomplete incoming time series $\mathbf{x}_t^{'} = \langle x_1, \ldots, x_t \rangle$ and a complete one $\mathbf{x}_T^{''} = \langle x_1, \ldots, x_T \rangle$ is done here using the Euclidean distance between the first t components of the two series. Again, other methods for computing the membership probabilities $P(\mathfrak{c}_k|\mathbf{x}_t)$ can be used.

The ECONOMY-K approach can be described by two algorithms. One for the learning phase (see Algorithm 5), and one for making decision (see Algorithm 6).

Algorithm 5 Learning algorithm for early classification of time series

Input:

- A training set $\mathcal{S}$ of m labeled time series $(\mathbf{x}_T^i, y^i) \in \mathbb{R}^T \times \mathcal{Y}$ $(1 \leq i \leq m)$;
- A validation set $\mathcal{S}'$ of m' labeled time series $(\mathbf{x}_T^j, y^j) \in \mathbb{R}^T \times \mathcal{Y}$ $(1 \leq j \leq m')$

1: **for** $t \in \{\min, \ldots, T\}$ **do**
2: Use a learning algorithm that takes as input $\mathcal{S}$ and returns a function $h_t : \mathbb{R}^t \to \mathcal{Y}$
3: **end for**

4: Use a clustering algorithm that takes as input $\mathcal{S}'$ and returns a set $\mathcal{C} = \{\mathfrak{c}_1, \ldots, \mathfrak{c}_K\}$, where each $\mathfrak{c}_k$ is a cluster of similar time series.

5: **for** $k \in \{1, \ldots K\}$ **do**
6: **for** $t \in \{\min, \ldots, T\}$ **do**
7: Compute the confusion matrix $P_t(\hat{y}|y, \mathfrak{c}_k)$ using h_t on the cluster $\mathfrak{c}_k$.
8: **end for**
9: **end for**

5.4.5 Computational complexity

The computational complexity of ECONOMY-K depends on the sequential implementation presented in Algorithms 5 and 6 where the main operations include:

1. Training a set of binary classifiers $\mathcal{H} = \{h_t\}_{min \leq t \leq T}$[1] on m time series of length T in the training set $\mathcal{S}$.

[1] In the text we often note that t is ranging between $[1, T]$ for simplicity. However, it should be noted that in practice $\min \leq t \leq T$ because below min measurements, the classifiers are not effective.

Algorithm 6 Prediction algorithm for early classification of time series

Input:

- An incomplete time series $\mathbf{x}_t$ with $1 \leq t \leq T$
- A set $\mathcal{C} = \{\mathfrak{c}_1, \ldots, c_K\}$ of clusters.

1: **for** $k \in \{1, \ldots, K\}$ **do**
2: Compute the membership probability $P(\mathfrak{c}_k|\mathbf{x}_t)$
3: **end for**

4: **for** $\tau \in \{0, \ldots, T-t\}$ **do**
5: Compute the expected cost $f_\tau(\mathbf{x}_t)$ using Equation (5.5)
6: **end for**
7: **return** $t^\star = t + \underset{\tau \in \{0,\ldots,T-t\}}{\text{ArgMin}} f_\tau(\mathbf{x}_t)$

2. Applying a clustering algorithm on m' time series of length T in the validation set $\mathcal{S}'$.

3. Estimating the confusion matrix on each cluster $\mathfrak{c}_k \in \mathcal{C}$ at each time step t.

4. Computing the membership probability $P(\mathfrak{c}_k|\mathbf{x}_t)$ of a time series $\mathbf{x}_t$ to each cluster $\mathfrak{c}_k$, where $1 \leq k \leq K$.

5. Estimating the cost function $f_\tau(\mathbf{x}_t)$ given a new incoming time series $\mathbf{x}_t$.

In the following, since the computational complexity of ECONOMY-K depends on the type of the used classifier and the used clustering algorithm, we give both **(A)** the complexity in absolute terms[1], and **(B)** the complexity implied by choices we have made. Let **L** be the complexity implied by learning any type of classifier. Let **P** be the complexity of applying this learned classifier on a single time series. Let **C** be the complexity of applying a clustering algorithm. And finally, let **d** be the complexity of the similarity measure used in computing the membership of a time series to a given cluster.

- **Step (1):** Complexity of learning T classifiers. In our case, we have chosen to use T classifiers of the same type where each is learned at each time step t over m time series trimmed to their t first components in the training set $\mathcal{S}$.

[1] We mean by complexity in absolute terms the fact the complexity is computed regardless the specific choices that can be made about the type of classifiers, the type of the clustering algorithm, the similarity measure used in the clustering algorithm, the membership function between a time series and a cluster of time series, etc.

  - **(A):** the complexity of learning T classifiers of any type is $\mathcal{O}(T \times \mathbf{L})$.
  - **(B):** in our case, we have chosen to use Multilayer Perceptron as a classifier implying a complexity of order $\mathbf{L} \simeq \mathcal{O}(m \times T)$, T here appears because it refers to the number of data points in the time series that are considered as variables in the input of a MLP. The complexity of learning T MLPs is then $\mathcal{O}(m \times T^2)$.

- **Step (2):** Complexity of applying a clustering algorithm over the validation set $\mathcal{S}'$.
  - **(A):** the complexity of a clustering algorithm of any type is $\mathcal{O}(\mathbf{C})$.
  - **(B):** in our case, we have chosen to use the K-means clustering algorithm that implies a complexity of order $\mathcal{O}(m' \times K \times T)$ where m' is the number of the time series in the validation set $\mathcal{S}'$, K is the number of clusters and T is length of each time series. Here, T appears since we use the Euclidean distance that computes the distances between each pair of data points in the considered time series of length T.

- **Step (3):** Complexity of computing confusion matrices. The confusion matrices are estimated at each time step t over each cluster $\mathfrak{c}_k$. This needs to apply a learned classifier over time series in each cluster $\mathfrak{c}_k$ at each time step t.
  - **(A):** the complexity of applying T learned classifiers over the K clusters is of order $\mathcal{O}(m' \times T \times \mathbf{P})$.
  - **(B):** in case of applying the learned MLPs over the K clusters at each time step t, the complexity is of order $\mathcal{O}(m' \times T^2)$.

- **Step (4):** Complexity of computing the membership function.
  - **(A):** the complexity of computing the membership of a given time series $\mathbf{x}_t$ to K clusters is $\mathcal{O}(K \times \mathbf{d})$ with $\mathbf{d}$ is the complexity of the function used to compute the membership of $\mathbf{x}_t$ to a cluster $\mathfrak{c}_k$.
  - **(B):** in our case, we have chosen to compute the cluster membership as a probability $P(\mathfrak{c}_k|\mathbf{x}_t)$ to belong to each cluster $\mathfrak{c}_k$ given a time series of length t. This entails a complexity of order $\mathcal{O}(K \times T)$ (see Equation 5.6).

- **Step (5):** Complexity of computing the expected cost function f_τ.

  - **(A):** the complexity of computing the expected cost function $f_\tau(\mathbf{x}_t)$, where $\tau \in \{0, \ldots, T-t\}$ for a given time series $\mathbf{x}_t$ (see Equation 5.5) is of order $\mathcal{O}(K \times \mathbf{d})$. Computing f_τ essentially depends on computing the membership of $\mathbf{x}_t$ to each of the K clusters and this which makes the approach adaptive. The other terms of f_τ are computed in advance before any new time series arrives. It should be noted that, at worse, the complexity of step (5) can be of order $\mathcal{O}(K \times \mathbf{d} \times T)$ when the algorithm waits until the last time to make a prediction.
  - **(B):** the complexity of computing f_τ is, at worse, of order $\mathcal{O}(K \times T^2)$.

- **Overall complexity of ECONOMY-K:** We give, here, the overall complexity of ECONOMY-K in absolute terms and depending on our choices.
  - **(A):** the overall complexity of ECONOMY-K is $\mathcal{O}(TL + \mathbf{C} + m'T\mathbf{P} + K\mathbf{d}T)$
  - **(B):** the complexity of computing ECONOMY-K depending on the specific choices we have made is $\mathcal{O}(mT^2 + m'KT + m'T^2 + KT^2) \simeq \mathcal{O}(mT^2)$, since we consider that $m \simeq m'$ and $K << m'$, the number of clusters K is usually much lower than the number of time series m' in the validation set $\mathcal{S}'$.

It is clear that the obtained complexity of computing ECONOMY-K mainly depends on our choices that concern:

- **The strategy used to predict the class label of a time series of any length t with $1 \leq t \leq T$.** We have chosen to use a simple and direct approach that uses T classifiers, each learned at each time step t (see Section 3.1.2). However, this multiplies T times the complexity of a classifier. One can avoid this additional complexity by using only one classifier. As discussed in Chapter 3, using one classifier can be achieved by imputing the missing values in the incoming time series, changing the representation of the time series to another time-invariant representation, etc.

- **The type of the classifier**. We have chosen to use MLP classifiers where their complexity depends on the size of the training data set and also the number of the input variables since we consider each data point in a time series as an explanatory variable. We can remedy to this by making variable selection, changing time series representation in order to reduce their dimentionality, choosing an other type of classifier that implies less computational complexity, etc.

- **The type of the distance used in the clustering algorithm**. We have used the Euclidean distance that computes the distance between each pair of data points in the considered time series. This, again, depends on the length T of the training time series and their representation in the temporal domain.
- **The membership function**. The function that defines the membership of an incoming time series to a cluster also depends on the choice of the Euclidean distance and thus the length T of time series.

These are choices we have made to implement ECONOMY-K leading to $\mathcal{O}(T^2)$ computations which can be computationally expensive when T increases. But, this does not exclude that making other choices to implement ECONOMY-K can imply less computational complexity.

5.4.6 Discussion

In this section, we proposed the ECONOMY-K approach that (i) uses a clustering technique to segment the complete training time series and computes in advance confusion matrices $P_t(\hat{y}|y, \mathfrak{c}_k)$ conditionally to each cluster and at each time step, and (ii) defines a new cost function f_τ (see Equation 5.5) to optimize the criterion that balances the expected gain in the classification cost in the future with the cost of delaying the decision. Adaptive predictions are obtained through using the membership probabilities $P(\mathfrak{c}_k|\mathbf{x}_t)$.

In this approach, we have sought to determine the baseline properties of our proposed framework. Thus, we have used simple techniques as: (i) clustering of time series using the simple K-means algorithm in order to compare the incoming time series to known shapes from the training set, (ii) a simple formula to estimate the membership probability $P(\mathfrak{c}_k|\mathbf{x}_t)$, and (iii) not optimized classifiers, here a set of classifiers $\mathcal{H} = \{h_t\}_{1 \leq t \leq T}$ learnt at each time step t.

We are aware that in this setting, it is required that the user makes a set of choices that may be baffling. First, the choice of a clustering method with all the attending choices of parameters (e.g. distances, parameters of the method, number of clusters, etc.). Second the choice of a distance between an incomplete time series $\mathbf{x}_t$ and a cluster $\mathfrak{c}_k$ made of complete time series. And, third, the choice of a membership function in order to compute $P(\mathfrak{c}_k|\mathbf{x}_t)$. All these choices can be tricky to make and can entail non negligible variations in the results.

In the following section, while considering the same methodology, i.e. (i) segmentation of the training time series, and (ii) definition of a new expected cost function f_τ for all the future time steps $\tau \in \{0, \dots, T-t\}$ based on the obtained segments, we introduce a competing method which avoids the burdens associated with the clustering segmentation used in ECONOMY-K. In particular, the new method uses a segmentation of the training set which is much more direct and informed since it uses the information about the class labels when segmenting. In addition it implies only one user-parameter.

5.5 ECONOMY-γ: Confidence-based early classification approach

5.5.1 Motivation

Aside the difficulties inherent in the use of a clustering method, specially over time series, there is another aspect that can make the ECONOMY-K approach less than optimal. Indeed, from Equation 5.5, repeated below:

$$f_\tau(\mathbf{x}_t) \;=\; \sum_{\mathfrak{c}_k \in \mathcal{C}} P(\mathfrak{c}_k|\mathbf{x}_t) \sum_{y \in \mathcal{Y}} P(y|\mathfrak{c}_k) \sum_{\hat{y} \in \mathcal{Y}} P_{t+\tau}(\hat{y}|y, \mathfrak{c}_k)\, \mathrm{C}(\hat{y}|y) \;+\; \mathrm{C}(t+\tau)$$

it is apparent that the membership of $\mathbf{x}_t$ to a cluster $\mathfrak{c}_k$ is important only insofar that the associated confusion matrices, given by $P_t(\hat{y}|y, \mathfrak{c}_k)$, are different from one cluster to the other. Otherwise, there is no point in considering $P(\mathfrak{c}_k|\mathbf{x}_t)$, that is to which cluster belongs the incoming time series. In addition, the conditional probabilities $P(y|\mathfrak{c}_k)$ should be as non uniform as possible.

If one, then, is considering an alternative way of segmenting the set of time series, it should better lead to confusion matrices that differ as widely as possible from one category to another. It should also lead to terms alike $P(y|\mathfrak{c}_k)$ as different as possible.

It is not easy to devise directly such a segmentation of the time series, but there exists an approach that naturally favors these properties. This approach is based on two substantial ideas:

1. The first idea consists in changing the representation of time series in a validation set $\mathcal{S}'$. The particularity of such a representation is that it also takes into account the information about the classes of time series. This is specially advantageous for

making a segmentation that differentiates classes of time series. To achieve such a segmentation, we use the output predictions $\{g_t(\mathbf{x}_t^i)\}_{1\leq i\leq|\mathcal{S}'|}$ of a probabilistic model g_t (learned on time series in the training set$\mathcal{S}$) to change the representation of time series, then, the segmentation is done on recoded sequences representing the times series. Compared with the segmentation used in ECONOMY-K which segments only once $\mathcal{S}'$ into K clusters of complete time series, the new approach, we propose here, segments $\mathcal{S}'$ into N groups at time step t with $1 \leq t \leq T$. This results different sets of N groups at each time step t.

2. The second idea is to use Markov chains to represent dependencies between successive observations of recoded sequences over time. This, somehow, leads this new approach to make adaptive predictions with respect to a new incoming time series.

In the following sections, we develop in details these ideas and deploy them to achieve our objectives of: (i) forming different groups in order to obtain confusion matrices that are different as much as possible and (ii) define a cost function f_τ that is able to give online, adaptive and non-myopic decisions.

5.5.2 Framework

We recall our goal which is to overcome the difficulties of computing the conditional probabilities $P(y|\mathbf{x}_t)$ and $P(\hat{y}|y,\mathbf{x}_t)$ in Equation 5.1, repeated below:

$$f(\mathbf{x}_t) \;=\; \sum_{y\in\mathcal{Y}} P(y|\mathbf{x}_t) \sum_{\hat{y}\in\mathcal{Y}} P(\hat{y}|y,\mathbf{x}_t)\, \mathrm{C}_t(\hat{y}|y) \;+\; \mathrm{C}(t)$$

In this section, while following our proposed methodology presented in Section 5.3 (used also in the ECONOMY-K approach), we propose a competing approach called ECONOMY-γ (as will be detailed later, γ refers to confidence intervals or states of the considered Markov chain).
This new approach proposes an intelligent and natural segmentation of the training time series through the use of a Markov chain that captures both typical evolutions of the training time series and possible continuations of any incomplete time series. In addition to capturing all of this relevant information, the states in the Markov chain should be meaningful in order to better capture the transition from one state to another. ECONOMY-γ is performed in two major steps:

1. During the learning phase, (i) Markov chains are specified and their states are carefully determined in order to capture typical evolutions of the training time

series and provide information about the possible continuations of the incoming incomplete time series $\mathbf{x}_t$, then (ii) the segmentation of the training time series is achieved not directly on values of time series but on their recoded sequence obtained using the considered Markov chain. The result is a set of segments having different prediction behaviors as detected by the Markov chain.

2. Estimating the terms of the cost function f_τ in order to determine the optimal time $t^\star$.

5.5.3 Learning step

In this work, the process we describe by a Markov chain is the evolution of time series over time. We will detail in the next section how evolutions of time series are not described by their discrete values but by their predictions estimated by a decision function at each time step t.

5.5.3.1 Specifying the Markov chain

We first start by describing the specific configuration used to construct our Markov chain (see Figure 5.5):

- A set of N states $\langle 1, 2, \ldots, N\rangle$ are decided upon at each time step t, where $1 \leq t \leq T$.
- The states at time step t are not connected to each other. The process starts in one of the states decided at time t and moves to one of the states decided at time $t+1$.
- If the process is in state u, $1 \leq u \leq N$, at time t, it moves to the state v, $1 \leq v \leq N$, at the next step with a probability denoted by m_{uv}^t.
- The probability $m_{uv}^t = p(\gamma_{t+1} = v | \gamma_t = u)$ is called a transition probability from time step t to time step $t+1$ and it only depends on the current state $\gamma_t = u$. This defines the first-order Markov hypothesis as:

$$\overrightarrow{\gamma}_{t+1} | \langle \gamma_1, \ldots, \gamma_{t-1}, \gamma_t \rangle = \overrightarrow{\gamma}_{t+1} | \langle \gamma_t \rangle \tag{5.7}$$

where $\overrightarrow{\gamma}_{t+1} = [p(\gamma_{t+1} = 1), \ldots, p(\gamma_{t+1} = N)]^\top = [p(\gamma_{t+1} = \ell)]_{1 \leq \ell \leq N}^\top$.

- A transition matrix $\mathbf{M}_t^{t+1}$ is constructed at each time step t considering all combinations of the states. Each of the matrix rows sums to 1:

$$\mathbf{M}_t^{t+1} = \begin{array}{c} \\ 1 \\ \vdots \\ \vdots \\ N \end{array} \begin{array}{c} \begin{array}{cccc} 1 & \cdots & \cdots & N \end{array} \\ \begin{pmatrix} m_{11}^t & \cdots & \cdots & m_{1N}^t \\ \vdots & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ m_{N1}^t & \cdots & \cdots & m_{NN}^t \end{pmatrix} \end{array} \tag{5.8}$$

- A Markov chain of order δ, with δ is finite, defines that the state at time step $t+1$ depends on the past δ states as defined by:

$$\overrightarrow{\gamma}_{t+1}|\langle\gamma_1,\ldots,\gamma_{t-1},\gamma_t\rangle = \overrightarrow{\gamma}_{t+1}|\langle\gamma_{t-\delta+1},\ldots,\gamma_{t-1},\gamma_t\rangle \tag{5.9}$$

5.5.3.2 Definition of Markov states

Most binary classifiers, like Neural Networks, Support Vector Machines, Naïve Bayes, decision trees, a.s.o. can easily be made to output a real number $g(\mathbf{x}_t)$ in the range $[0,1]$ such that the output of the classifer $h_t(\mathbf{x}_t) = -1$ if $g(\mathbf{x}_t) \leq 0.5$ and $h_t(\mathbf{x}_t) = +1$ otherwise (the threshold 0.5 depends on the calibration of the function g_t). The value $g(\mathbf{x}_t)$ can be interpreted as expressing a confidence level in the prediction of the class to which belongs $\mathbf{x}_t$, and when some care is taken over the choice of the loss function used to learn g, $g(\mathbf{x}_t)$ can even be interpreted as a probability to belong to class $+1$. (See [81] for instance, that shows how to associate a confidence level to the prediction of a SVM).

What is interesting is that the confusion matrices over examples that are predicted with a confidence level close to 1 and over examples that are predicted with a confidence level close to 0.5 are generally quite different, which is natural if the confidence level $g(\mathbf{x}_t)$ somewhat reflects the probability that the class $+1$ has been predicted for $\mathbf{x}_t$. Hence, the idea to use confidence intervals to define the states of the Markov chain and to differentiate classes of time series.

5.5.3.3 Segmentation

For each time step t, a function g_t is learnt using the training set $\mathcal{S}$ and hence a decision function:

$$h_t(\cdot) = \text{sign}\big(g_t(\cdot) - 0.5\big). \tag{5.10}$$

More specifically, g_t is learnt using the training time series in $\mathcal{S}$ reduced to their first t components $\langle x_1^i, \ldots, x_t^i \rangle$ and their label y^i.

Then, a discretization of the confidence interval $[0, 1]$ for each time step is learnt. Indeed, for each time step t, the associated function g_t induced from the training set $\mathcal{S}$, is applied on time series in a second training set $\mathcal{S}'$ disjoint from $\mathcal{S}$. An ordering of the time series in $\mathcal{S}'$ is then done based on sorting the outputs $\{g_t(\mathbf{x}_t^i)\}_{1 \leq i \leq |\mathcal{S}'|}$ from the time series with the highest confidence value $g_t(\mathbf{x}_t)$, to the time series with the lowest one. As such, it is easy to compute a set of $N-1$ thresholds on $[0, 1]$ for time step t such that each of the induced N sub-intervals is associated with approximately $|\mathcal{S}'|/N$ time series.

In this setting, the N sub-intervals induced at each time t are the states of the Markov chain (see Figures 5.5 and 5.6).

This segmentation method automatically corrects any bias in the calibration of g_t and provides N meaningful subsets $\mathcal{S}^t = \{S_l^t\}_{1 \leq l \leq N}$ of training time series at each time step t.

5.5.3.4 Recoding

Given this discretization scheme, resulting in varying discretization thresholds depending on the time steps $t \in \{1, \ldots, T\}$, each time series $\mathbf{x}_t$ can be recoded as a sequence of t confidence intervals or Markov chain states γ_t where $\gamma_t = \ell$ if $g_t(\mathbf{x}_t)$ is in the sub-interval corresponding to $\ell \in \{1, \ldots, N\}$. (See Figure 5.6).

5.5.3.5 Transition probabilities estimation

This recoding provides a way to compute the likely future outlines of a given incomplete time series $\mathbf{x}_t$. Given the code $\langle \gamma_1, \ldots, \gamma_t \rangle$ of a new incoming time series $\mathbf{x}_t$, the objective is to compute the probability for each future time step $t+s$, $(1 \leq s < T - t)$, that $\gamma_{t+s} = \ell$ with $\ell \in \{1, \ldots, N\}$. Let us note $\overrightarrow{\gamma}_{t+s}$ the vector made of the N corresponding probabilities: $[p(\gamma_{t+s} = 1), \ldots, p(\gamma_{t+s} = N)]^\top$. Then, in all generality, we want to compute $\langle \overrightarrow{\gamma}_{t+1}, \ldots, \overrightarrow{\gamma}_T \rangle | \langle \gamma_1, \ldots, \gamma_t \rangle$: the transition probabilities for all future time steps given $\langle \gamma_1, \ldots, \gamma_t \rangle$, the coded time series $\mathbf{x}_t$.

This would entail learning a set of dependency matrix of $(T-t)\times t$ values, and these dependency matrices should be learned for all possible $t \in \{1, T-1\}$.

The number of possible sequences in the code is N^T, and it is required to estimate the probability of each one of them. With $N = 5$ and $T = 100$, limiting sequences to 100 time steps, this is already approximately 7×10^{72} numbers to estimate.

This becomes computationally very expensive when the number of states N and the dimension of time series T increase. This is why, in the next section and after defining the general expected cost function f_τ in the ECONOMY-γ approach, we give a series of simplifying assumptions based on the Markov conditions in order to make our approach tractable.

5.5.4 Estimation of the expected cost function f_τ

When a new input time series $\mathbf{x}_t$ of length t is considered, it is recoded as a sequence of t confidence intervals $\langle \gamma_1, \ldots, \gamma_t \rangle$, as explained in Section 5.5.3.4.
Given that, at time t, $T-t$ measurements are still missing on the incoming time series, it is possible to compute the expected costs $f_\tau(\mathbf{x}_t)$ of classifying $\mathbf{x}_t$ at each future time step $\tau \in \{1, \ldots, T-t\}$, using the subsets $\{\mathcal{S}^t\}_{1 \leq t \leq T}$ obtained by segmenting the training time series.
The expected cost function $f_\tau(\mathbf{x}_t)$ used in the ECONOMY-γ approach can then be defined as:

$$f_\tau(\mathbf{x}_t) = \sum_{y,\hat{y} \in \mathcal{Y}} \sum_{\ell=1}^{N} (\overrightarrow{\gamma}_{t+\tau} | \langle \gamma_1, \ldots, \gamma_t \rangle) \, P_{t+\tau}(\hat{y} | y, \gamma_{t+\tau} = \ell) \times \mathrm{C}(\hat{y}|y) + \mathrm{C}(t+\tau) \quad (5.11)$$

From Equation 5.11, it is clear that the expected cost for each future time step τ are estimated conditionally to the incoming time series $\mathbf{x}_t$ through the use of the conditional probabilities $\overrightarrow{\gamma}_{t+\tau} | \langle \gamma_1, \ldots, \gamma_t \rangle$. The conditional probabilities $P_{t+\tau}(\hat{y} | y, \gamma_{t+\tau} = \ell)$ are computed in advance over each subset S^t_ℓ, $\ell \in \{1, \ldots, N\}$, and at each time step t using a classifier h_t, where $t \in \{1, \ldots, T\}$.

As detailed above, the complexity of the estimation of the terms $\overrightarrow{\gamma}_{t+\tau} | \langle \gamma_1, \ldots, \gamma_t \rangle$ which are the transition probabilities for the future time steps given the coded sequence of $\mathbf{x}_t$, exponentially increases when T and N increase. In addition, since this estimation should be carried out online with each new incoming measurement, it should be then adapted to situations where the speed of generating measurements is very fast. For these reasons,

we propose three simplifying assumptions based on Markov conditions.

5.5.5 Simplifying assumptions using Markov conditions

First simplifying assumption: Markov condition ahead

The first simplifying assumption concerns the estimation of the terms $\overrightarrow{\gamma}_{t+\tau}|\langle\gamma_1,\ldots,\gamma_t\rangle$. Only $\overrightarrow{\gamma}_{t+1}$ will be computed given as input the coded sequence: $\overrightarrow{\gamma}_{t+1}|\langle\gamma_1,\ldots,\gamma_t\rangle$, and all probability vectors after time $t+1$, i.e. $\overrightarrow{\gamma}_{t+s}$ with $1 \leq s \leq T-t$ are computed using one order dependency: $\overrightarrow{\gamma}_{t+s+1}|\overrightarrow{\gamma}_{t+s}$.

In words, within this hypothesis, only the probability vector at the first future time step $\overrightarrow{\gamma}_{t+1}$ is estimated given the whole past history of $\mathbf{x}_t$ coded as $\langle\gamma_1,\ldots,\gamma_t\rangle$, and thereupon all probability vectors at future time steps are supposed to depend only on the previous one.

Transition matrices estimation

Let us note $\mathbf{M}_t^{t+1}$ the $N \times N$ transition matrix from time step t to time step $t+1$ with elements $m_{u,v}^t = p(\gamma_{t+1} = v|\gamma_t = u)$, where u, v are confidence intervals at time steps t and $t+1$ respectively, that is $u, v \in \{1,\ldots,N\}^2$ (see Figure 5.7). With $N = 5$ different confidence intervals, the transition matrices have each 25 elements.

Let us note $\mathbf{M}_{1,\ldots,t}^t$ the transition matrix from a coded sequence $\langle\gamma_1,\ldots,\gamma_t\rangle$ represented as $t \times N$ probability vector $\mathbf{P}_{\langle\gamma_1,\ldots,\gamma_t\rangle}$ to the probability vector $\overrightarrow{\gamma}_{t+1}$. For illustration, suppose that $N = 5$ and we only look at a length 2 sequence coded as $\langle\gamma_1 = 2, \gamma_2 = 4\rangle$. Then the corresponding probability vector has 10 components: $\mathbf{P}_{\langle\gamma_1,\gamma_2\rangle} = [0,1,0,0,0,0,0,0,1,0]^\top$.

Using these notations, and given an input coded sequence $\langle\gamma_1,\ldots,\gamma_t\rangle$, one can estimate the future probability vector using equation:

$$\overrightarrow{\gamma}_{t+\tau}|\langle\gamma_1,\ldots,\gamma_t\rangle \;=\; \left[\prod_{s=1}^{\tau-1} \mathbf{M}_{t+s}^{t+s+1}\right] \mathbf{M}_{1,\ldots,t}^t \, \mathbf{P}_{\langle\gamma_1,\ldots,\gamma_t\rangle} \tag{5.12}$$

In words, given the past coded history $\langle\gamma_1,\ldots,\gamma_t\rangle$, one computes the next probability vector $\overrightarrow{\gamma}_{t+1}$ using $\mathbf{M}_{1,\ldots,t}^t \mathbf{P}_{\langle\gamma_1,\ldots,\gamma_t\rangle}$, and then, the probability vector at horizon $t+\tau$ is computed thanks to a product of one order transition matrices $\prod_{s=1}^{\tau-1} \mathbf{M}_{t+s}^{t+s+1}$.

However, even this simplified scheme necessitates to learn large transition matrices $\mathbf{M}_{1,\ldots,t}^t$ with N^t elements to be learnt, and this for all possible values of $t \in \{1,\ldots,T-1\}$,

which amounts to $\frac{N^{T+1}-1}{N-1}$ probability values to be estimated, which, for $N = 5$ and $T = 100$ gives approximately 10^{71} values to learn.

Second simplifying assumption: Weak Markov assumption behind

Suppose then that only the last δ codes $\langle \gamma_{t-\delta+1}, \ldots, \gamma_t \rangle$ of an input sequence $\mathbf{x}_t$ be used to compute the future states, we get the following expression:

$$\overrightarrow{\gamma}_{t+\tau} | \langle \gamma_{t-\delta+1}, \ldots, \gamma_t \rangle \; = \left[\prod_{s=1}^{\tau-1} \mathsf{M}_{t+s}^{t+s+1} \right] \mathsf{M}_{t-\delta+1,\ldots,t}^{t} \; \mathsf{P}_{\langle \gamma_{t-\delta+1},\ldots,\gamma_t \rangle} \tag{5.13}$$

where $\mathsf{M}_{t-\delta+1,\ldots,t}^{t}$ which requires $\mathcal{O}(N^{\delta})$ probability values to be learned. Even with $\delta = 2$, and $N = 5$, $5^3 = 125$ probabilities must be estimated, and this for all values of t.

In such setting, we get the following expected decision cost for future time steps $\tau \in \{0, T-t\}$:

$$\begin{aligned} f_\tau(\mathbf{x}_t) \; &= \; \sum_{y,\hat{y} \in \mathcal{Y}} \sum_{\ell=1}^{N} (\overrightarrow{\gamma}_{t+\tau} | \langle \gamma_{t-\delta+1}, \ldots, \gamma_t \rangle) \, P_{t+\tau}(\hat{y} | y, \gamma_{t+\tau} = \ell) \; \times \mathrm{C}(\hat{y}|y) + \mathrm{C}(t+\tau) \\ &= \; \sum_{y,\hat{y} \in \mathcal{Y}} \sum_{\ell=1}^{N} P(\gamma_{t+\tau} = \ell | \langle \gamma_{t-\delta+1}, \ldots, \gamma_t \rangle) \, P_{t+\tau}(\hat{y} | y, \gamma_{t+\tau} = \ell) \; \times \mathrm{C}(\hat{y}|y) + \mathrm{C}(t+\tau) \end{aligned} \tag{5.14}$$

Third simplifying assumption: Strong Markov assumption behind

In the experiments reported in Chapter 6, and because we had only a few thousands training time series, we have radically simplified the approach and used one order memory, yielding the equation:

$$\overrightarrow{\gamma}_{t+\tau} | \langle \gamma_t \rangle \; = \; \left[\prod_{s=0}^{\tau-1} \mathsf{M}_{t+s}^{t+s+1} \right] \mathsf{P}_{\langle \gamma_t \rangle} \tag{5.15}$$

which computes the vector $\overrightarrow{\gamma}_{t+\tau} = [p(\gamma_{t+\tau}) = \ell]_{1 \leq \ell \leq N}^{\top}$.

This requires only the estimation of $T \times N^2$ probability elements using the training set. This first order Markov model provides a baseline with which to assess the minimal capacity of the method.

We can now return to the computation of the expected decision cost for future time

steps $\tau \in \{1, \ldots, T-t\}$:

$$\begin{aligned} f_\tau(\mathbf{x}_t) &= \sum_{y,\hat{y}\in\mathcal{Y}} \sum_{\ell=1}^{N} (\overrightarrow{\gamma}_{t+\tau}|\langle\gamma_t\rangle)\ P_{t+\tau}(\hat{y}|y,\gamma_{t+\tau}=\ell) \times \mathrm{C}(\hat{y}|y) + \mathrm{C}(t+\tau) \\ &= \sum_{y,\hat{y}\in\mathcal{Y}} \sum_{\ell=1}^{N} P(\gamma_{t+\tau}=\ell|\langle\gamma_t\rangle)\ P_{t+\tau}(\hat{y}|y,\gamma_{t+\tau}=\ell)\ \times \mathrm{C}(\hat{y}|y) + \mathrm{C}(t+\tau) \end{aligned} \tag{5.16}$$

Using Equation (5.16), one obtains an estimation of the optimal decision time to come:

$$t^* = t + \underset{\tau\in\{0,\ldots,T-t\}}{\mathrm{ArgMin}}\ f_\tau(\mathbf{x}_t)$$

The whole method can be described by two algorithms. One for learning from a set $\mathcal{S}$ of training sequences (see Algorithm 7), and one for making decision (see Algorithm 8).

Algorithm 7 Learning algorithm for early classification of time series

Input:

- A training set $\mathcal{S}$ of m labeled time series $(\mathbf{x}_T^i, y^i) \in \mathbb{R}^T \times \mathcal{Y}$ $(1 \leq i \leq m)$;
- A validation set $\mathcal{S}'$ of m' labeled time series $(\mathbf{x}_T^j, y^j) \in \mathbb{R}^T \times \mathcal{Y}$ $(1 \leq j \leq m')$;
- a set $\mathcal{G} = \{\mathcal{G}_{\min} \cup \ldots \cup \mathcal{G}_T\}$ where each $\mathcal{G}_t$ is itself a set of scoring functions: $g_t : \mathbb{R}^t \to [0,1]$;

for $t \in \{\min, \ldots, T\}$ **do**
  Use a learning algorithm that takes as input $\mathcal{S}$ and $\mathcal{G}_t$ and returns a function g_t
  Using g_t and $\mathcal{S}$: compute N confidence intervals on $[0,1]$ as explained in Section 5.5 and return subsets S_ℓ^t, where $\ell \in \{1, \ldots, N\}$
end for

for $\ell \in \{1, \ldots N\}$ **do**
  for $t \in \{\min, \ldots, T\}$ **do**
    Compute the confusion matrix $P_t(\hat{y}|y, \gamma_t = \ell)$ using h_t on the subset S_ℓ^t.
  end for
end for

for $t \in \{1, \ldots, T-1\}$ **do**
  Compute the transition matrices $\mathbf{M}_t^{t+1}$
end for

Algorithm 8 Prediction algorithm for early classification of time series

Input:

- An incomplete time series $\mathbf{x}_t$ with $1 \leq t \leq T$
- Subsets of time series S_ℓ^t with $1 \leq t \leq T$ and $1 \leq \ell \leq N$.

1: Compute the sequence $\langle \gamma_1, \ldots, \gamma_t \rangle$ coding for $\mathbf{x}_t$

2: **for** $\tau \in \{0, \ldots, T-t\}$ **do**
3: Compute the expected cost $f_\tau(\mathbf{x}_t)$ using Equation (5.16)
4: **end for**
5: **return** $t^\star = t + \underset{\tau \in \{0,\ldots,T-t\}}{\text{ArgMin}} f_\tau(\mathbf{x}_t)$

5.5.6 Computational complexity

The computational complexity of ECONOMY-γ mainly depends on the multiplication of transition matrices and the type of the used classifiers. The main operations of ECONOMY-γ in Algorithms 7 and 8 include:

1. Training a set of classifier functions $\{g_t\}_{\min \leq t \leq T}$ using the training set $\mathcal{S}$ composed of m training time series and hence a set of T decision function $\mathcal{H} = \{h_t\}_{\min \leq t \leq T}$ can be defined by: $h_t = \text{sign}(g_t(\cdot) - 0.5)$.
2. Segmenting time series in the validation set $\mathcal{S}'$.
3. Estimating confusion matrices on each subset S_ℓ^t at each time step $1 \leq t \leq T$.
4. Estimating $T-1$ transition matrices on m' time series in the validation set $\mathcal{S}'$.
5. Computing the expected cost function $f_\tau(\mathbf{x}_t)$ given a new incoming time series $\mathbf{x}_t$.

We detail for each step, **(A)** the complexity in absolute terms[1], and **(B)** the complexity depending on the specific choices we have made. To allow the comparison against ECONOMY-K, we use the same notation: $\mathbf{L}$ is the complexity of learning a classifier of any type. $\mathbf{P}$ is the complexity of applying this learned classifier on a single time series. $\mathbf{D}$ is the complexity of discretizing the classifier outputs used in the segmentation method, and $\mathbf{d}$ is the complexity of assigning the classifier output to a given segment.

[1] We mean by complexity in absolute terms the fact the complexity is computed regardless the specific choices that can be made about the type of classifiers, the type of the clustering algorithm, the similarity measure used in the clustering algorithm, the membership function between a time series and a cluster of time series, etc.

We recall that the used segmentation method in ECONOMY-γ proceeds as follows. (i) recoding time series needs to apply a learned classifier to predict the score $g_t(\mathbf{x}_t)$ of each time series $\mathbf{x}_t$ in $\mathcal{S}'$ at each time step t. (ii) sorting the obtained $g_t(\mathbf{x}_t)$ for all time series in $\mathcal{S}'$ and then discretize them to obtain N sub-intervals at each time step. This discretization schema is simply using the equal frequency binning procedure which divides the sorted outputs $\{g_t(\mathbf{x}_t^i)\}_{1 \leq i \leq m'}$ into N sub-intervals at each time step such that each interval contains approximately $|\mathcal{S}'|/N$ time series with adjacent values. This meets our goal of making a segmentation that differentiates classes of time series. Other discretization techniques could be used if they satisfy this goal.

- **Step (1):** Complexity of learning T classifiers.
  - **(A):** the complexity of learning T classifiers of any type is $\mathcal{O}(T \times \mathbf{L})$.
  - **(B):** The complexity of learning T MLPs is then $\mathcal{O}(m \times T^2)$.
- **Step (2):** Complexity of segmenting the training time series in $\mathcal{S}'$.
  - **(A):** the complexity of segmenting m' time series is $\mathcal{O}(m' \times \mathbf{T} \times (\mathbf{P} \times \mathbf{d} + \mathbf{D}))$.
  - **(B):** the complexity of the segmentation method using MLP classifiers is $\mathcal{O}(m' \times T \times (T \times \log N + N \log N)) \simeq \mathcal{O}(m' \times T \times T \times \log N)$, since usually $N \ll T$.
- **Step (3):** Complexity of computing confusion matrices. The confusion matrices are estimated at each time step t over each subset S_ℓ^t with $1 \leq \ell \leq N$ and $1 \leq t \leq T$. This needs to apply a learned classifier on each subset S_ℓ^t.
  - **(A):** the complexity of applying T learned classifiers over the N subsets is of order $\mathcal{O}(m' \times T \times \mathbf{P})$.
  - **(B):** the complexity of applying the learned MLPs over the N subsets, at each time step t, is of order $\mathcal{O}(m' \times T^2)$.
- **Step (4):** Complexity of computing $T - 1$ transition matrices, each of size $N \times N$ yields $\mathcal{O}(T \times \log N)$.
- **Step (5):** Complexity of computing the expected cost function f_τ. Computing f_τ in ECONOMY-γ essentially depends on computing the conditional probabilities $\overrightarrow{\gamma}_{t+\tau}|\langle\gamma_t\rangle$ with $\langle\gamma_t\rangle$ is the sequence coding the new incoming time series $\mathbf{x}_t$, and this which makes the approach adaptive. The other terms of f_τ are computed in advance before any new time series arrives. However, it is needed to multiply transition matrices of size $N \times N$ yielding $\mathcal{O}(N^2)$ computations.

  - **(A):** the complexity of computing the expected cost function $f_\tau(\mathbf{x}_t)$, where $\tau \in \{0, \ldots, T-t\}$ for a given time series $\mathbf{x}_t$ (see Equation 5.14) is of order $\mathcal{O}(\mathbf{P} + \mathbf{D} + T \times N^2)$. In all and at worse, the complexity of step (5) can be of order $\mathcal{O}(\mathbf{P} + \mathbf{D} + T^2 \times N^2)$ when the algorithm waits until the last time to make a prediction.
  - **(B):** the complexity of computing f_τ is, at worse, of order $\mathcal{O}(T + \log N + T^2 \times N^2)$.

- **Overall complexity of ECONOMY-K:** We give, here, the overall complexity of ECONOMY-γ in absolute terms and depending on our choices.
  - **(A):** the overall complexity of ECONOMY-γ is:
    $\mathcal{O}((TL) + (m'T\mathbf{PD}) + (m'T\mathbf{P}) + (\mathbf{P} + \mathbf{D} + T^2N^2)) \simeq \mathcal{O}(m'T\mathbf{PD} + T^2N^2)$
  - **(B):** the complexity of computing ECONOMY-γ depending on the specific choices we have made is:
    $\mathcal{O}((mT^2) + (m'T^2 \log N) + (m'T^2) + (T + \log N + T^2N^2) \simeq \mathcal{O}(mT^2)$, since we consider that $m \simeq m'$ and $N << m$ is a user parameter that remains constant.

	ECONOMY-K	ECONOMY-γ
Complexity (A)	$\mathcal{O}(T\mathbf{L} + m'T\mathbf{P} + K\mathbf{d}T)$	$\mathcal{O}(T\mathbf{L} + m'T\mathbf{PD} + T^2N^2)$
Complexity (B)	$\mathcal{O}(mT^2)$	$\mathcal{O}(mT^2)$

Table 5.1: Computational complexities of ECONOMY-K and ECONOMY-γ computed (A) in absolute terms and (B) depending on the specific choices we have made.

In Table 5.1, we show computational complexities implied respectively by ECONOMY-K and ECONOMY-γ.
Regarding these complexities, one can conclude that ECONOMY-γ achieves a computational complexity equal to that implied by ECONOMY-K. Even though both algorithms are conceived and behave differently, they imply the same computational complexity.

5.5.7 Discussion

The method ECONOMY-γ described above has several advantages:

1. Aside the choice of the class of prediction functions g (and hence of decision functions h) that must be made whatever the approach, there are two parameters to

set. The first is N, the number of confidence intervals or Markov chain states one is willing to consider at each time step. Higher values of N may seem preferable because they would yield higher precision. But this is illusory since what matters is the difference in the confusion matrices. In addition, one obtains a better precision on the estimation of these matrices if the number of training sequences used to compute them is large. As will be shown in the experiments (see Chapter 6), a good choice seems to be $N = 5$. The second parameter is the order of the dependency taken into account, similarly to Markov chain models that can depend on the past to various degrees.

2. The confusion matrices that appear in Equation 5.16 tend naturally to differ, leading to better estimates of the future decision costs.

3. The conditional probabilities $P_{t+\tau}(\hat{y}|y, \gamma_{t+\tau} = \ell)$ tend also to differ for different values of the confidence interval ℓ, which favors better predictions.

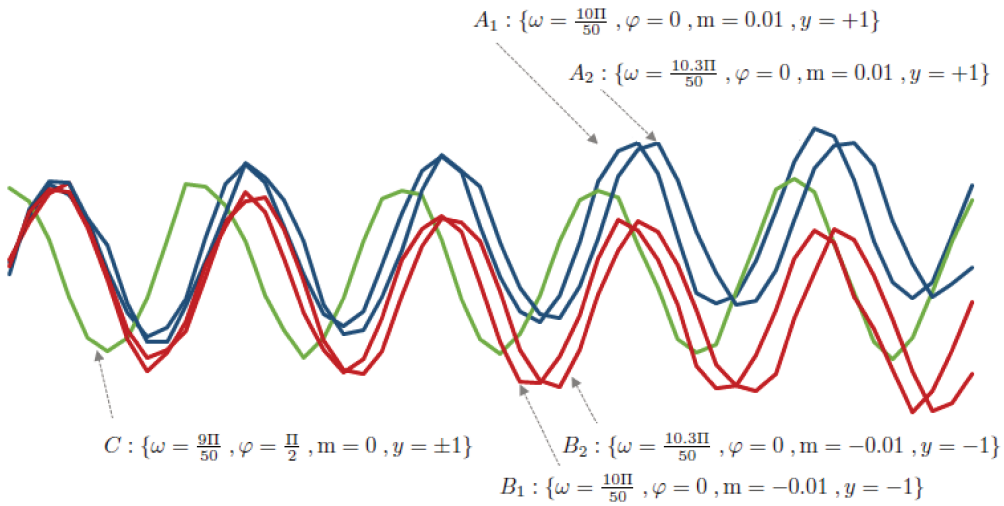
Furthermore, it is expected that using the same algorithm with higher order of time dependencies taken into account would further improve the performances. These richer models should indeed be able to extract the useful information in the training set and new incoming time series, and come near the optimal decision time and optimal cost. However, only very large training sets can allow a learning algorithm to reach this type of performance, by enabling the learning of the large number of conditional dependencies involved in these higher order models.

Summary

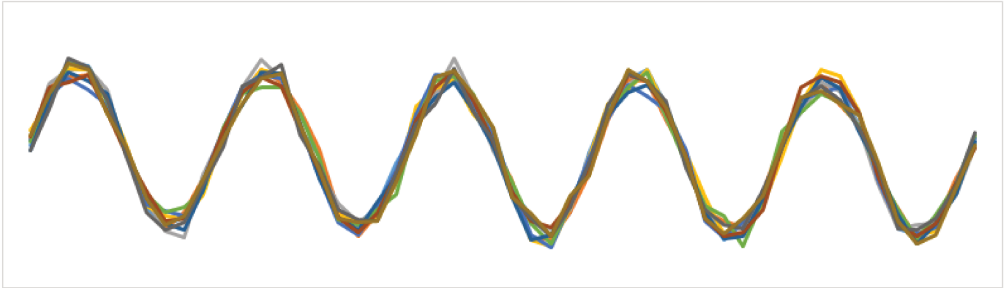
In this chapter, we revisited the problem of early classification of time series when delaying decision incurs a rising cost. We cast the problem to a cost-sensitive online decision making problem and proposed an optimization criterion that balances the expected gain in the classification cost in the future with the cost of delaying the decision.
Within this conceptual framework, we proposed two algorithms that differ in the manner they consider the information contained in the training time series and in the incoming time series. Both approaches are adapted to the peculiarities of the incoming time series, since the cost function is re-estimated with each added information, and provide non-myopic decisions. The first approach called ECONOMY-K is intuitively alluring since it provides a simple solution to the problem that captures typical evolutions of the time series using a clustering technique. The second method is called ECONOMY-γ. It is

more direct and informed since it provides a natural schema using Markov chain concept to capture the generalized patterns in the training time series. It captures typical behaviors of the training time series with a precision that depends on N, the number of confidence intervals considered, and on the order of dependency taken into account. The advantage of ECONOMY-γ against ECONOMY-K, beyond implying less user parameters and competitive computational complexity, is the use of a segmentation method that also takes into account the information about the class labels of the training time series. It succeeds thus to group time series described by the same class in spite of their dissimilarity in shape. This makes the segmentation method in ECONOMY-γ more informed than the clustering used in the ECONOMY-K approach that segments the training data only according to their shape.

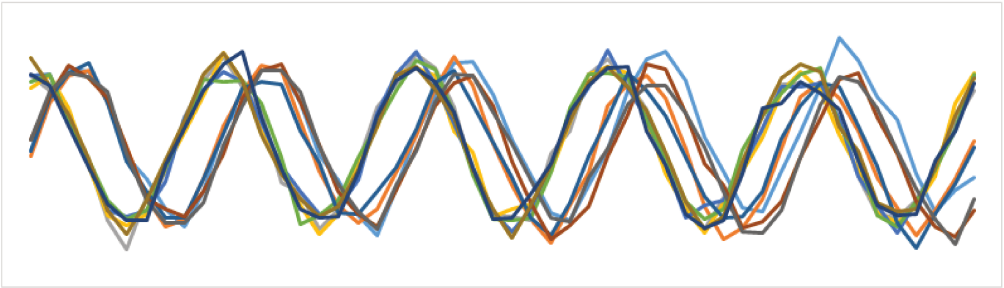
In the next chapter, extensive experiments on real data sets and in controlled situations with synthetic data sets under a wide variety of parameter values are conducted to vindicate the potential of the proposed approaches for making early decisions.



(a) An example of different categories of generated time series. Time series A_1 and A_2 are labeled $+1$, time series B_1 and B_2 are labeled -1 and the time series C is duplicated and arbitrarily labeled -1 or $+1$.



(b) A sample of time series depicted from a group obtained by the ECONOMY-K segmentation technique. Time series in the same group are similar in shape.



(c) A sample of time series depicted from a group obtained by the ECONOMY-γ segmentation technique. Time series are of different shapes but are predicted similarly.

Figure 5.3: Comparison over the synthetic data set and under same conditions between the composition of two groups of time series obtained by two different segmentation techniques from ECONOMY-K and ECONOMY-γ approaches.

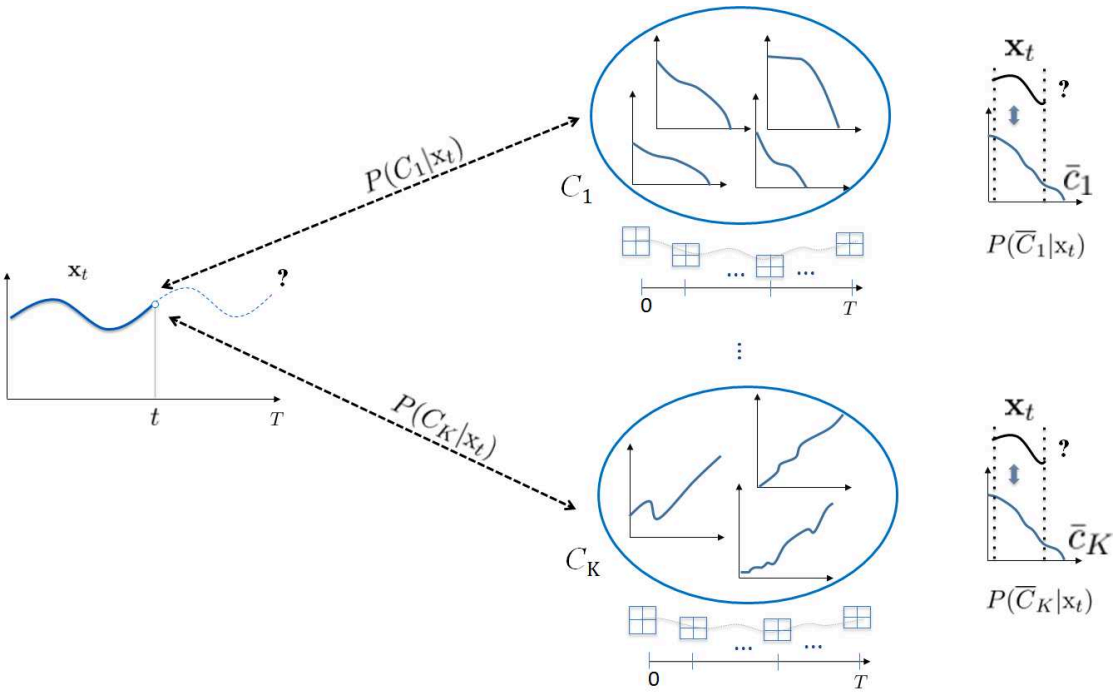


Figure 5.4: An incoming (incomplete) time series is compared to each cluster $\mathfrak{c}_k$ obtained from the training set of complete time series. The confusion matrices for each time step t and each cluster $\mathfrak{c}_k$ are computed as explained in the text.

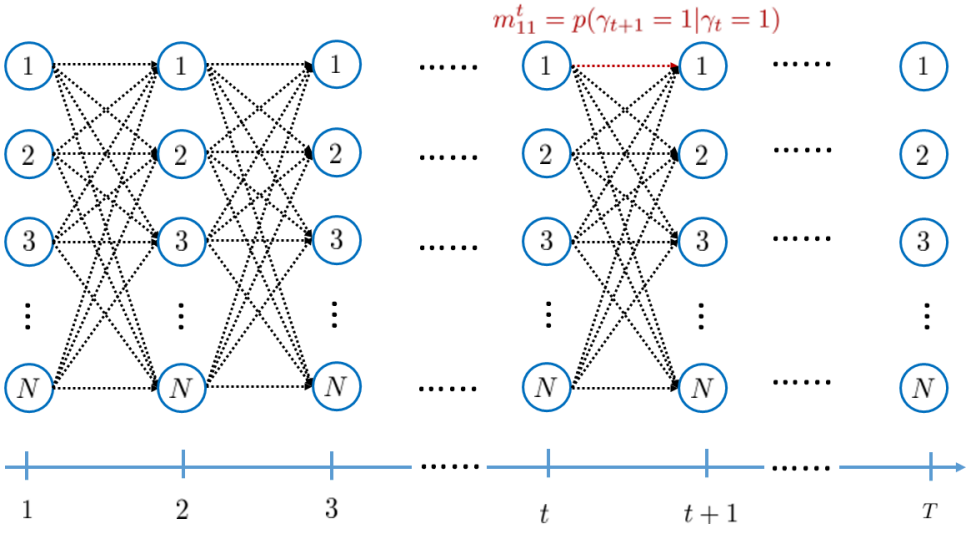


Figure 5.5: A special Markov chain of transition probabilities between states over time.

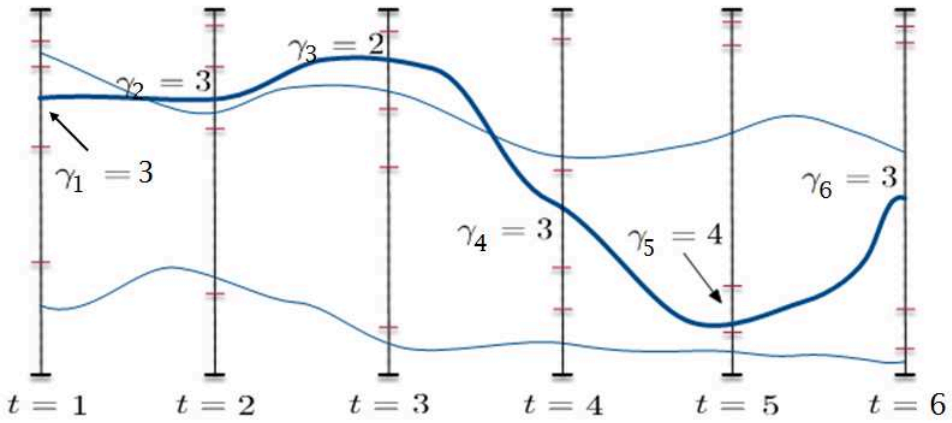


Figure 5.6: How time series are coded using confidence intervals (Markov states). Here, the thick curve is coded as $\langle \gamma_1 = 3, \gamma_2 = 3, \gamma_3 = 2, \gamma_4 = 3, \gamma_5 = 4, \gamma_6 = 3 \rangle$. Actually, the time series here depicted as curves in order to better visualize them are sequence of points $\langle x_1, x_2, x_3, x_4, x_5, x_6 \rangle$ with no curves in between. The confidence intervals vary from one time step to another as explained in the text.

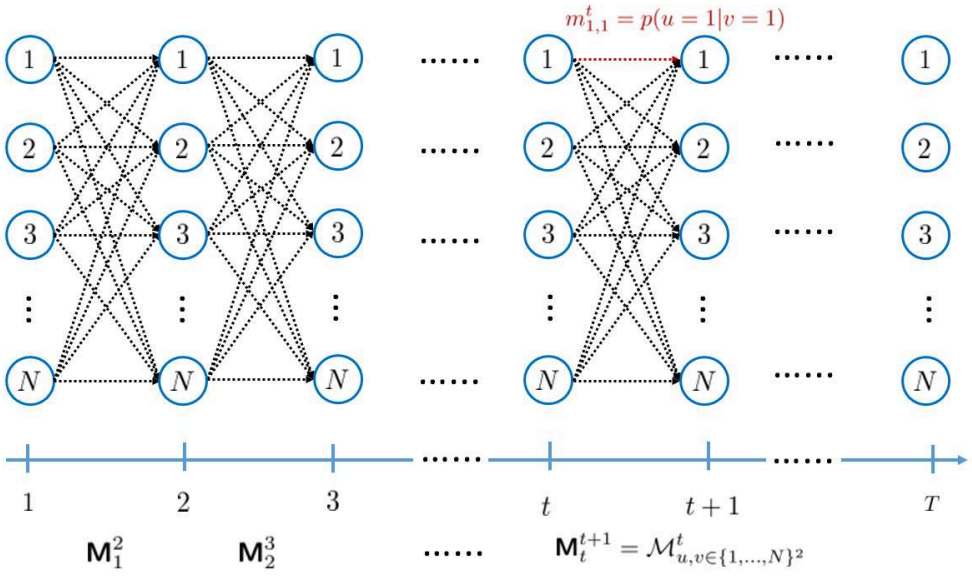


Figure 5.7: Dependency matrices learned at each time step $t \in \{1, \ldots, T-1\}$.

Chapter 6

Experimental study

Introduction

In the proposed ECONOMY-K and ECONOMY-γ approaches (see Section 5), the problem of early classification of time series was cast to a cost-sensitive decision making problem with three properties: (i) both the quality and the earliness of the prediction are taken into account in the total criterion to be optimized, (ii) the criterion is adaptive, in that, the output prediction depends upon the incoming time series $\mathbf{x}_t$, and (iii) the proposed methods lead to non-myopic decision schemes where the expected optimal horizon τ^* is estimated instead of just deciding that now is, or is not, the time to make a prediction.

In this chapter, we conduct extensive experiments to check the validity of the proposed methods and explore their capacities for various conditions. To this end, we devise controlled experiments with synthetic data sets for which we could vary a number of control parameters. Then, we examine the behaviors of both methods on real data sets selected from the UCR Time Series Classification/Clustering Repository [62].

6.1 Experimental evaluation on synthetic data sets

The goal of the experimental evaluation is to measure how the methods behave when (i) the difficulty of the decision task varies, and (ii) faced with varying level of increasing costs when delaying the decision.

6.1.1 The generation of the synthetic data sets

The difficulty of making early predictions can naturally be determined using two types of controlling parameters. One that controls the information that can be gained about the class of the incoming time series with each new data point. And one that controls the noise level of data points. The two parameters are not independent as more noise decreases the information that can be gained, but they still are complementary as the noise level is supposed to be constant over time when the gain of information can vary.

The overall idea is to generate sets of time series according to two class models, one for the $+1$ class and one for the -1 class. In addition, within each class, there are sub-classes, some of them that can share a strong similarity with sub-classes of the other class.

Specifically, the time series have been generated according to the following equation:

$$\mathbf{x}_t = \underbrace{t \times \text{slope} \times \text{class}}_{\text{information gain}} + \underbrace{\mathrm{x}_{max} \sin(\omega_i \times t + \varphi_j)}_{\text{sub shape within class}} + \underbrace{\eta(t)}_{\text{noise factor}} \tag{6.1}$$

The higher the value of the *slope* factor (noted m below), the higher the gain at each time step. At the same time, x_{max} controls the importance of the sub-classes within each class. If $\mathrm{x}_{max} = 0$ there are no sub-classes, and little possible confusion between the classes, except for the noise factor $\eta(t)$. If x_{max} has a large value, the sub-shape tends to dominate the information gain factor, at least for not large enough time step t. Figure 6.1 illustrates what can be obtained for three classes of time series, one with slope $\mathrm{m} = 0.01$ (class $y = +1$), one with $\mathrm{m} = -0.01$ (class $y = -1$) and one with $\mathrm{m} = 0$ (a confusing class). Here $\mathrm{x}_{max} = 5$ and the sub-classes are determined by the period ω_i and phase φ_j. The noise factor $\eta(t)$ is randomly chosen from a Gaussian distribution with a mean μ and a standard deviation σ.

We conducted experiments using two disjoint training sets $\mathcal{S}_1$ and $\mathcal{S}_2$ each containing $2,500$ time series and a testing set $\mathcal{T}$ containing $5,000$ time series. Each set is equally divided between the two classes $y = -1$ and $y = +1$.

In ECONOMY-K approach, the set $\mathcal{S}_1$ is used for learning the set of classifiers $\mathcal{H} = \{h_t\}_{1 \leq t \leq T}$, while the set $\mathcal{S}_2$ is used to form a set of clusters $\mathcal{C} = \{\mathfrak{c}_k\}_{1 \leq k \leq K}$ and compute the confusion matrices $P_{t+\tau}(\hat{y}|y, \mathfrak{c}_k)$, for each cluster $\mathfrak{c}_k$ and for each time step t.

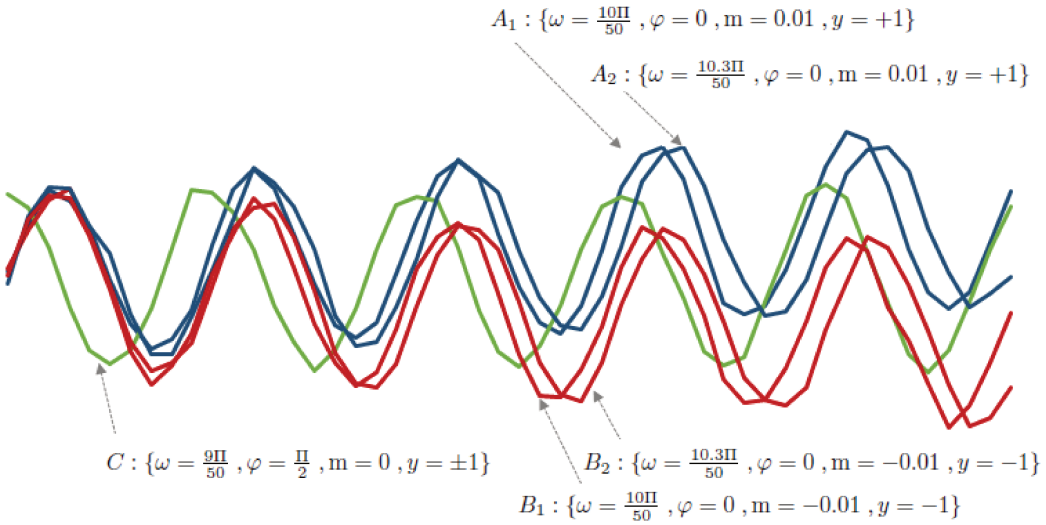


Figure 6.1: An example from the synthetic data set $\mathcal{S}$ where $\eta : (\mu = 0, \sigma = 0.2)$. Time series A_1 and A_2 are labeled $+1$, time series B_1 and B_2 are labeled -1 and the time series C is duplicated and arbitrarily labeled -1 or $+1$.

In ECONOMY-γ approach, the set $\mathcal{S}_1$ is used for learning the set of functions $\mathcal{G} = \{g_t\}_{1 \leq t \leq T}$, while the set $\mathcal{S}_2$ is used to compute Markov states $\{\gamma_s\}_{1 \leq s \leq N}$ for each time step t, the transition matrices $\mathbf{M}_t^{t+1}$ and the confusion matrices $P_{t+\tau}(\hat{y}|y, \gamma_{t+\tau} = \ell)$, $\ell \in \{1, \ldots, N\}$.

Table 6.1 displays the range of parameters used for generating the synthetic data sets we used in the experiments. Note that T, the maximal number of data points in a time series is relatively low here: 50. This is because, on one hand, the study was motivated by electrical time series measured each 30' over one day, and, on the other hand, because, even in this short timespan, the relevant properties of the algorithms can be evaluated. Experimental results with higher values of T (e.g. time series in *Strawberry* real data set are composed of 235 data points) will presented in Section 6.2.

In addition to varying some parameters when generating the data sets, we varied the delaying cost function C(t) that expresses how costly it is to delay making a decision. The cost function is a non decreasing function of time. In our experiments, we used linear cost functions: C(t) = d $\times$ t, with d is a constant parameter. Other forms of the cost function (e.g. polynomial, exponential, etc.) can be used depending on the context.

Notation	Description	Value(s)
T	Number of data points	50
x_{max}	Sine amplitude	5
ω_i, $1 \leq i \leq 3$	Sine period	$\omega_i \in \{\frac{9\Pi}{T}, \frac{10\Pi}{T}, \frac{10.3\Pi}{T}\}$
φ_j, $1 \leq 2$	Sine phase	$\varphi_j \in \{0, \frac{\Pi}{2}\}$
m	Slope	$\mathrm{m} \in \{0.005, 0.01, 0.05, 0.07, 0.1\}$
η	Noise level as Gaussian distribution	$\mu = 0$, $\sigma \in \{1 : 200\}$
$K_{y=+1}$	Number of sub-groups in each class	$K_{y=+1} \in \{3, 5\}$

Table 6.1: The set of parameters used for the generation of the data sets.

6.1.2 Experimental settings

In these controlled experiments, we varied the following parameters:

- The level of distinction between the classes and specifically the rate of information gain controlled by m.
- The number $K_{y=+1}$ of sub-groups in each class and their shape (given by the term $\mathrm{x}_{max}\ \sin(\omega_i \times t + \varphi_j)$).
- The noise level $\eta(t)$.
- The cost of delaying the decision $\mathrm{C}(t)$.

and, we examine the impact of these controlling factors on the estimated optimal decision time and the estimated costs. Specifically, for each method, and for each experimental condition determined by the above mentioned controlling factors, we measure the following quantities, detailed in Table 6.2: $\overline{\tau}^{\star}_{\mathrm{ETM}}$, $\overline{\mathrm{C}}_{\mathrm{RCM}}$, $\overline{\tau}^{\star}_{\mathrm{PETM}}$, $\overline{\mathrm{C}}_{\mathrm{PRCM}}$, $\overline{\tau}^{\star}_{\mathrm{ITM}}$, $\overline{\mathrm{C}}_{\mathrm{ICM}}$.

The real cost $\overline{\mathrm{C}}_{\mathrm{RCM}}$ is the average real cost obtained for each test time series $\mathbf{x}_t$ by computing the predicted class $\hat{y} = h_{t^\star}(\mathbf{x}_{t^\star})$ and comparing it with the real label y at the estimated time $t^\star$ and evaluating: $\mathrm{C}(\hat{y}|y) + \mathrm{C}(t^\star)$.

The real cost $\overline{\mathrm{C}}_{\mathrm{PRCM}}$ is the average real cost computed a posteriori using all measurements in the complete time series. This measures how much the estimated cost is

Quantity	Description
$\overline{\tau}^{\star}_{\text{ETM}}$	mean of the decision time $\pm$ standard deviation computed for ECONOMY-K (resp. ECONOMY-γ) by Equation 5.5 (resp. 5.16)
$\overline{\text{C}}_{\text{RCM}}$	mean real cost using decision time $\overline{\tau}^{\star}_{\text{ETM}}$
$\overline{\tau}^{\star}_{\text{PETM}}$	decision time a posteriori $t^* = \text{ArgMin}_{t\in\{1,\ldots,T\}} f_t(\mathbf{x}_t)$ computed for ECONOMY-K (resp. ECONOMY-γ) by Equation 5.5 (resp. 5.16).
	(using the knowledge of the complete series)
$\overline{\text{C}}_{\text{PRCM}}$	mean real cost using decision time $\overline{\tau}^{\star}_{\text{PETM}}$
$\overline{\tau}^{\star}_{\text{ITM}}$	mean time before $h_t(\mathbf{x}_t) = y$ (perfect algorithm)
$\overline{\text{C}}_{\text{ICM}}$	mean real cost when deciding the first time that $h_t(\mathbf{x}_t) = y$ (perfect algorithm)

Table 6.2: Quantities measured in the experiments.

far from the optimal cost computed by the systems (using Equations 5.5 and 5.16).

The $\overline{\text{C}}_{\text{ICM}}$ value is an optimistic optimal value. It is the cost (or gain if this is a negative value) that the system would endure if it made a decision as soon as the prediction is correct, $h_t(\mathbf{x}_t) = y$, which can happen accidentally even though the decision function h_t is bad. We still report this value since it gives an idea of how far is the method to this (unrealistic) optimal early decision method.

6.1.3 Empirical results

An overview of the results for various combinations of the above discussed parameters are shown in Table 6.3 as obtained on testing set with a fixed trend parameter m = 0.07. More results when varying m are reported in Appendix A.

For the classifiers, we have used Naive Bayes classifiers and Multi-layer Perceptrons with one hidden layer of [t + 2/2] neurons. In this section, we choose to show results obtained only using the Multi-Layer Perceptron since both classifiers give similar results.

C(t)	$\eta(t)$	ECONOMY-K				ECONOMY-γ					
		$\overline{\tau}^{\star}_{\text{ETM}}$	$\overline{\text{C}}_{\text{RCM}}$	$\overline{\tau}^{\star}_{\text{PETM}}$	$\overline{\text{C}}_{\text{PRCM}}$	$\overline{\tau}^{\star}_{\text{ETM}}$	$\overline{\text{C}}_{\text{RCM}}$	$\overline{\tau}^{\star}_{\text{PETM}}$	$\overline{\text{C}}_{\text{PRCM}}$	$\overline{\tau}^{\star}_{\text{ITM}}$	$\overline{\text{C}}_{\text{ICM}}$
	0.1	5.9±0.4	-0.76	7.1±2.8	-0.76	8.4±8.6	-0.76	8.6±4.8	-0.76	4.0±0.2	-0.77
	0.2	5.1±0.4	-0.64	6.8±3.1	-0.65	17.4±14.6	-0.74	10.0±4.5	-0.74	4.2±0.9	-0.8
	0.5	9.0±4.2	-0.45	12.2±4.0	-0.66	16.3±10.5	-0.74	18.8±10.0	-0.74	5.1±2.4	-0.87
	1.0	14.4±2.3	-0.72	12.3±4.4	-0.6	20.8±10.0	-0.74	23.2±8.3	-0.74	5.3±3.0	-0.92
0.001	1.5	16.1±1.5	-0.65	9.6±4.8	-0.27	23.6±13.2	-0.68	30.3±10.4	-0.7	6.5±4.5	-0.96
	5.0	10.9±7.1	-0.1	9.3±5.4	-0.01	32.8±5.2	-0.65	42.3±6.1	-0.67	9.7±7.8	-0.96
	10.0	13.1±11.5	-0.08	11.1±7.5	-0.02	41.4±5.7	-0.52	45.2±4.4	-0.5	11.4±10.0	-0.95
	15.0	10.9±6.7	-0.03	11.4±6.4	-0.02	34.5±16.6	-0.25	39.2±14.9	-0.27	12.2±10.7	-0.95
	20.0	12.4±10.6	-0.01	12.6±9.2	0.0	44.5±3.9	-0.24	46.3±3.2	-0.24	11.3±9.9	-0.96
	0.1	4.0±0.0	-0.71	5.5±1.3	-0.71	5.3±1.4	-0.71	6.4±2.2	-0.71	4.0±0.2	-0.74
	0.2	5.1±0.4	-0.6	5.9±2.1	-0.59	6.7±1.9	-0.67	7.8±2.7	-0.67	4.2±0.9	-0.77
	0.5	5.1±0.6	-0.24	7.0±3.4	-0.27	11.1±4.5	-0.64	12.0±2.8	-0.64	5.1±2.4	-0.83
	1.0	6.7±2.5	-0.26	8.1±3.8	-0.33	15.0±4.0	-0.62	16.3±4.6	-0.59	5.3±3.0	-0.88
0.01	1.5	7.2±3.8	-0.09	8.8±4.3	-0.13	17.0±9.2	-0.51	19.1±5.5	-0.53	6.5±4.5	-0.9
	5.0	6.2±3.7	0.03	8.3±4.2	0.08	26.9±6.6	-0.34	33.3±5.5	-0.33	9.7±7.8	-0.87
	10.0	4.2±1.0	0.04	8.5±4.3	0.09	18.1±8.4	0.01	38.1±10.3	-0.08	11.4±10.0	-0.85
	15.0	4.9±1.8	0.04	6.4±2.4	0.05	5.4±0.5	0.05	8.3±9.7	0.06	12.2±10.7	-0.84
	20.0	4.1±1.1	0.04	6.9±3.9	0.07	9.7±1.6	0.08	10.2±2.0	0.08	11.3±9.9	-0.86
	0.1	4.0±0.0	-0.47	4.5±0.9	-0.44	4.6±0.8	-0.45	4.7±1.1	-0.44	4.0±0.2	-0.5
	0.2	5.0±0.2	-0.3	5.3±0.4	-0.28	4.6±0.8	-0.33	5.2±1.5	-0.35	4.2±0.9	-0.51
	0.5	5.0±0.2	0.06	5.5±1.2	0.12	7.5±1.3	-0.11	8.2±2.1	-0.11	5.0±2.0	-0.52
	1.0	4.0±0.0	0.13	5.4±0.8	0.13	7.1±2.1	0.03	8.3±2.6	0.07	5.2±2.2	-0.56
0.07	1.5	4.1±0.6	0.28	5.7±1.2	0.38	4.0±0.0	0.27	7.7±3.4	0.23	6.2±3.2	-0.52
	5.0	4.0±0.0	0.28	5.4±1.4	0.38	4.0±0.0	0.28	4.0±0.0	0.28	9.2±7.0	-0.29
	10.0	4.0±0.0	0.28	4.8±1.0	0.34	4.0±0.0	0.28	4.0±0.0	0.28	9.8±8.0	-0.18
	15.0	4.0±0.0	0.28	4.3±0.6	0.3	5.1±0.7	0.36	4.9±0.7	0.34	10.1±8.4	-0.14
	20.0	4.0±0.0	0.28	4.2±0.5	0.29	4.0±0.0	0.28	4.0±0.0	0.28	9.7±7.8	-0.2
	0.1	4.0±0.0	-0.35	4.5±0.9	-0.31	4.0±0.0	-0.35	4.1±0.6	-0.34	4.0±0.2	-0.37
	0.2	4.0±0.0	-0.18	4.5±0.9	-0.14	4.6±0.8	-0.2	4.6±1.1	-0.2	4.2±0.9	-0.38
	0.5	5.0±0.2	0.21	5.3±0.5	0.25	4.7±1.0	0.26	5.4±1.7	0.27	5.0±1.7	-0.37
	1.0	4.0±0.0	0.25	5.4±0.5	0.28	5.0±0.0	0.29	6.2±2.0	0.26	5.1±2.0	-0.41
0.1	1.5	4.0±0.1	0.39	5.5±0.8	0.53	4.0±0.0	0.39	4.3±1.1	0.39	6.1±3.0	-0.33
	5.0	4.0±0.0	0.4	5.1±0.8	0.51	4.0±0.0	0.4	4.0±0.0	0.4	8.1±5.7	-0.03
	10.0	4.0±0.0	0.4	4.6±0.8	0.46	4.0±0.0	0.4	4.0±0.0	0.4	7.6±5.6	0.07
	15.0	4.0±0.0	0.4	4.1±0.4	0.41	5.0±0.6	0.5	4.6±0.5	0.46	8.0±6.4	0.14
	20.0	4.0±0.0	0.4	4.1±0.4	0.41	4.0±0.0	0.4	4.0±0.0	0.4	7.6±5.6	0.05

Table 6.3: Comparison of early classification costs and time decision between the ECONOMY-K vs ECONOMY-γ approaches. Experiments are performed over the simulated sine data sets with a fixed rate of information gain m = 0.07.

Impacts of different parameters

Globally, both ECONOMY-K and ECONOMY-γ methods meet behaviors that are expected from early classification systems. From the results reported in Table 6.3, one can see that when the noise level is low and the delaying cost is low too, the systems are able to reach a high level of performance by waiting increasingly as the noise level augments. When the delaying cost is high ($\mathrm{C}(t) = 0.1 \times t$), on the other hand, the systems take a decision earlier at the cost of a somewhat lower prediction performance. Indeed, with rising levels of noise, the systems decide that it is not worth waiting and make a prediction early, often at the earliest possible moment, which was set to 4 in our experiments[1]. Figure 6.2 and Figure 6.3 better show these observations.

More specifically:

- **Impact of the noise level** $\eta(t)$: As expected, up to a certain value, rising levels of noise $\eta(t)$ entail increasing delays before a decision is decided upon by the systems. For example in Figure 6.2, for a fixed delaying cost ($\mathrm{C}(t) = 0.01 \times t$) and a rate of information gain ($\mathrm{m} = 0.07$), ECONOMY-K reaches its high value of the estimated decision time mean $\overline{\tau}^{\star}_{\mathrm{ETM}} = 7.2 \pm 3.8$ when the noise factor $\eta(t) = 1.5$, while for ECONOMY-γ, the estimated decision time $\overline{\tau}^{\star}_{\mathrm{ETM}} = 26.9 \pm 6.6$ when $\eta(t) = 5.0$. Then, for both approaches, a decrease of $\overline{\tau}^{\star}_{\mathrm{ETM}}$ is observed, which corresponds to the fact that there is no gain to be expected by waiting further. In fact, quite often the systems decide to make their prediction at the earliest possible time, which was set to 4 in our experiments. Accordingly, the decision costs, as measured with $\overline{\mathrm{C}}_{\mathrm{RCM}}$ and $\overline{\mathrm{C}}_{\mathrm{PRCM}}$, becomes expensive as well when $\eta(t)$ rises.

- **Impact of the delaying cost** $\mathrm{C}(t)$: The role of the delaying cost $\mathrm{C}(t)$ appears clearly. When $\mathrm{C}(t)$ is very low, the algorithms tend to wait longer before making a decision, often waiting the last possible time. On the other hand, with rising $\mathrm{C}(t)$, the optimal decision time $\overline{\tau}^{\star}_{\mathrm{ETM}}$ decreases sharply, converging to the minimal possible value of 4. This yields increasing decision costs $\overline{\mathrm{C}}_{\mathrm{RCM}}$ as the systems are constrained to make quick decisions that commonly entail more prediction errors (see Figure 6.3).

- **Impact of the rate of information gain controlled by** m: the value of m, which controls the level of distinction of the classes $y = +1$ and $y = -1$, is striking

[1] Below 4 measurements, the classifiers are not effective.

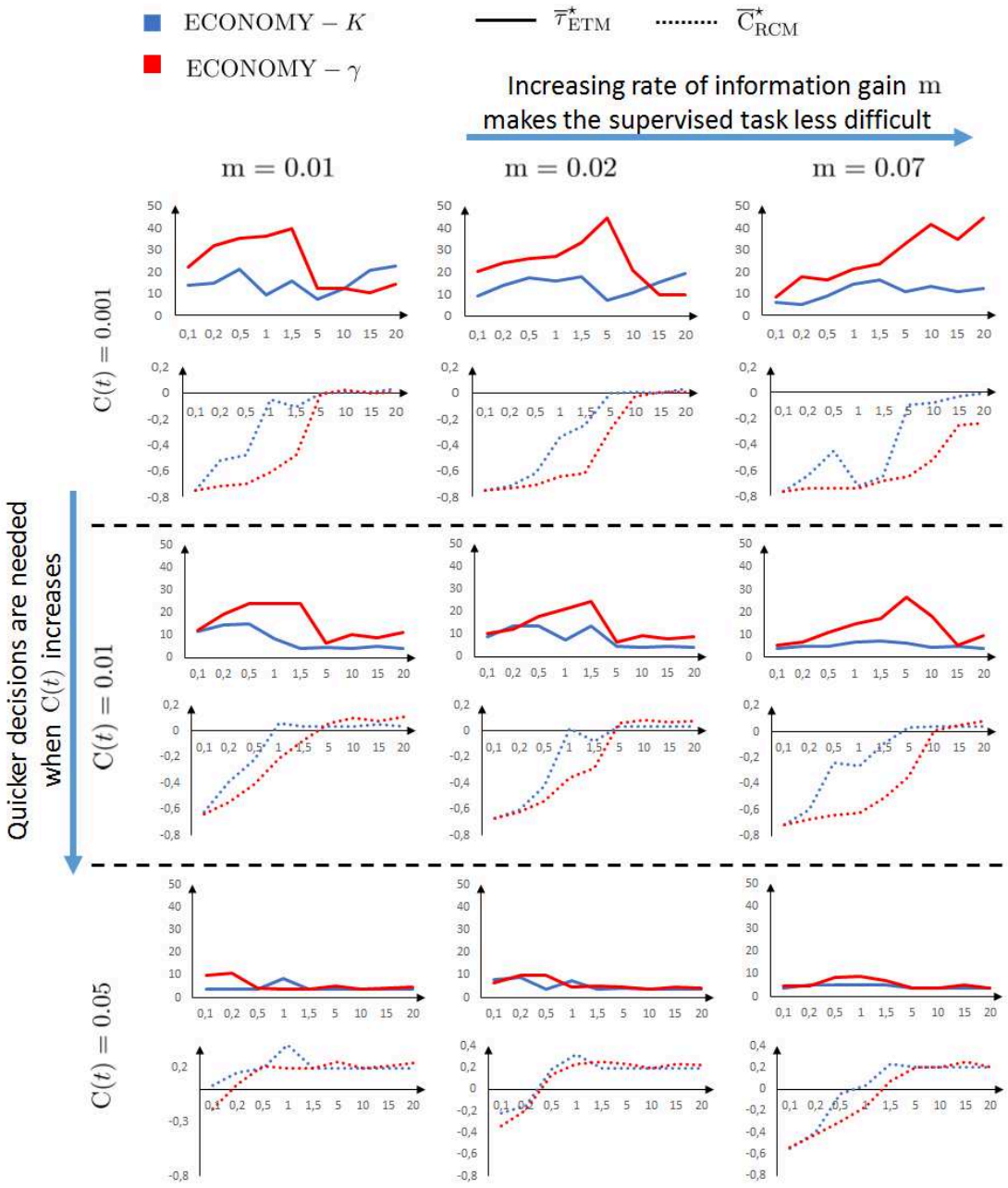


Figure 6.2: **Impact of the noise level** $\eta(t)$. The performances of ECONOMY-K vs ECONOMY-γ over the synthetic data sets. The x-axis represents the noise level $\eta(t)$ and the y-axis represents the estimated time $\overline{\tau}_{\mathrm{ETM}}$ (solid line) and its associated real cost $\overline{\mathrm{C}}_{\mathrm{RCM}}$ (dashed line). Decision time *curves* can only show evolutions of decision times $\overline{\tau}_{\mathrm{ETM}}$ when varying m and C(t), nothing can be said about the best approach. By contrast, cost *curves*, in addition to show evolutions of costs $\overline{\mathrm{C}}_{\mathrm{RCM}}$ when varying m and C(t), they give the winning approach (the one with the lowest costs).

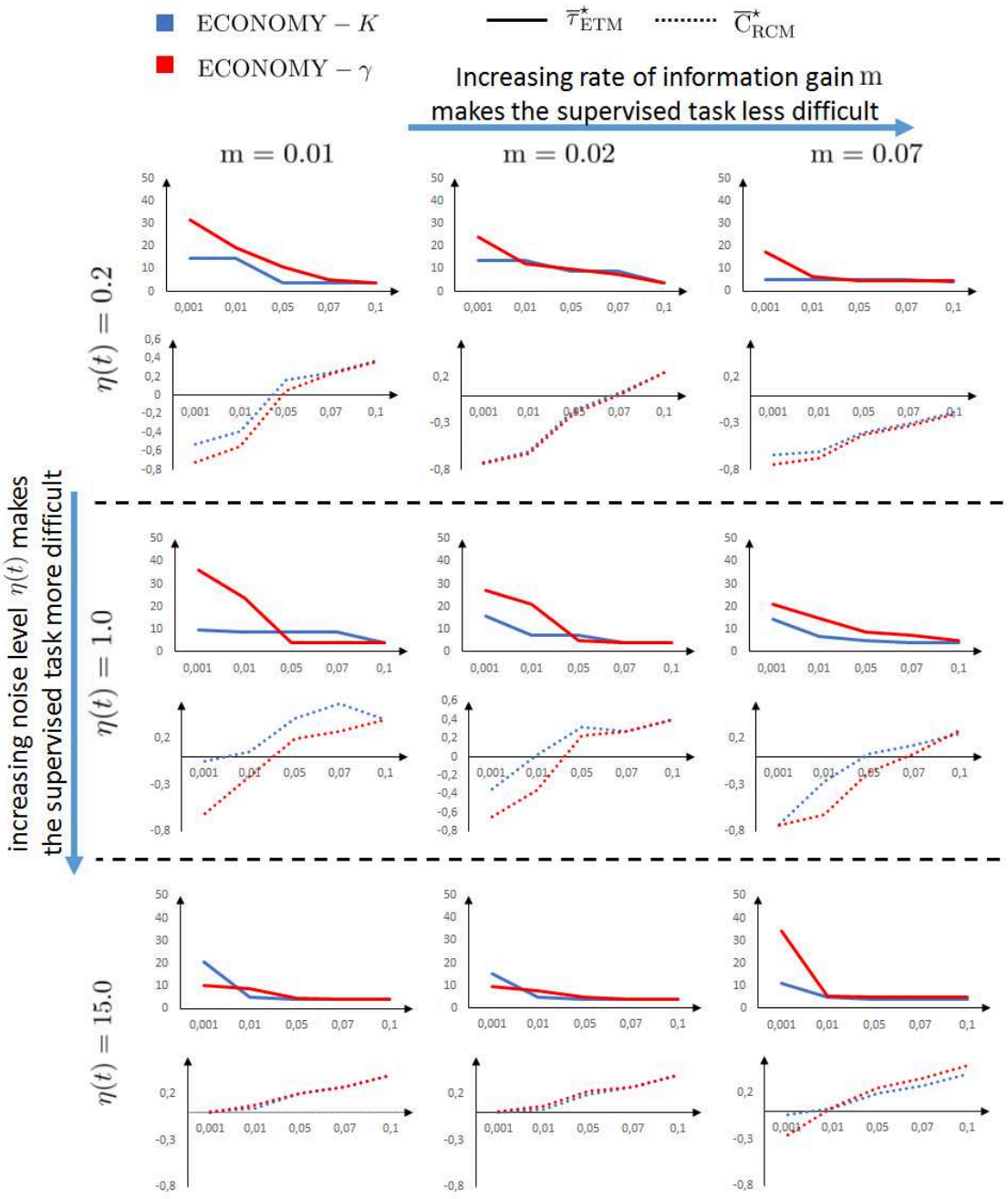


Figure 6.3: **Impact of the delaying cost** $\mathrm{C}(t)$. The performances of ECONOMY-K vs ECONOMY-γ over the synthetic data sets. The x-axis represents the delaying cost $\mathrm{C}(t)$ and the y-axis represents the estimated decision time $\bar{\tau}_{\mathrm{ETM}}$ depicted with solid lines and its associated real cost $\overline{\mathrm{C}}_{\mathrm{RCM}}$ depicted with dashed lines. Results are reported for noise levels $\eta(t) \in \{0.2, 1.0, 15.0\}$, and rates of information gain $\mathrm{m} \in \{0.01, 0.02, 0.07\}$.

on the average time of decision $\overline{\tau}^{\star}_{\text{ETM}}$ particularly for small values of noise level. For example, in Figure 6.4, for both approaches and for fixed values of the noise level ($\eta(t) = 0.2$) and the delaying cost ($\text{C}(t) = 0.01 \times t$), the decision time $\overline{\tau}^{\star}_{\text{ETM}}$ decreases whenever m increases. At the same time, m strongly impacts the decision costs as $\overline{\text{C}}_{\text{RCM}}$ are less expensive when m increases. However, for high values of the noise level and delaying cost (e.g. $\eta(t) = 15.0$ and $\text{C}(t) = 0.01$), the increase of the rate of information gain m has no further impact.

- **Impact of the number of sub-groups in each class**: In order to measure the *effect of the complexity of each class* on the decision problem, we changed the number of shapes in each class as well. This is easily done in our setting by using sets of different values of the parameters in Equation 6.1. For instance, Tables 6.4 and 6.5 respectively report the results obtained from ECONOMY-K and ECONOMY-γ when the number of sub-groups of class $y = -1$ was set to $K_{-1} = 3$ while it was set to $K_{+1} = 3$ then $K_{+1} = 5$ for class $y = +1$.

  Additionally, the Area Under the ROC Curve AUC is also reported in order to evaluate the quality of the prediction at the estimated optimal decision time $\overline{\tau}_{\text{ETM}}$.

  In both approaches, for low values of the rate of information gain ($\text{m} \in \{0.01, 0.02\}$), the number of sub-groups in each class, and hence the complexity of the classes, slightly influences the results. We observe that, although the decision task becomes harder, the decision time slightly decreases yielding thus higher costs. At the same time, the AUC decreases. However, when the rate of information gain increases ($\text{m} = 0.07$), the decision task becomes easier, and globally the results are not impacted by the complexity of classes in both approaches.

- **Impact of varying the number of clusters in ECONOMY-K approach and the number of Markov chain states in ECONOMY-γ approach over synthetic data sets:** Tables 6.6 and 6.7 show the impact of varying the number of clusters K in ECONOMY-K approach and the number of Markov chain states N in ECONOMY-γ approach over the sine synthetic data sets. Results of the optimal time decision and the cost incurred by both methods are given when varying $K \in \{3, 5, 6, 8, 9, 15\}$ and $N \in \{5, 10\}$.

  From Tables 6.6 and 6.7 one can observe that varying the number of clusters impacts the results on time decision and costs for the ECONOMY-K approach. By contrast,

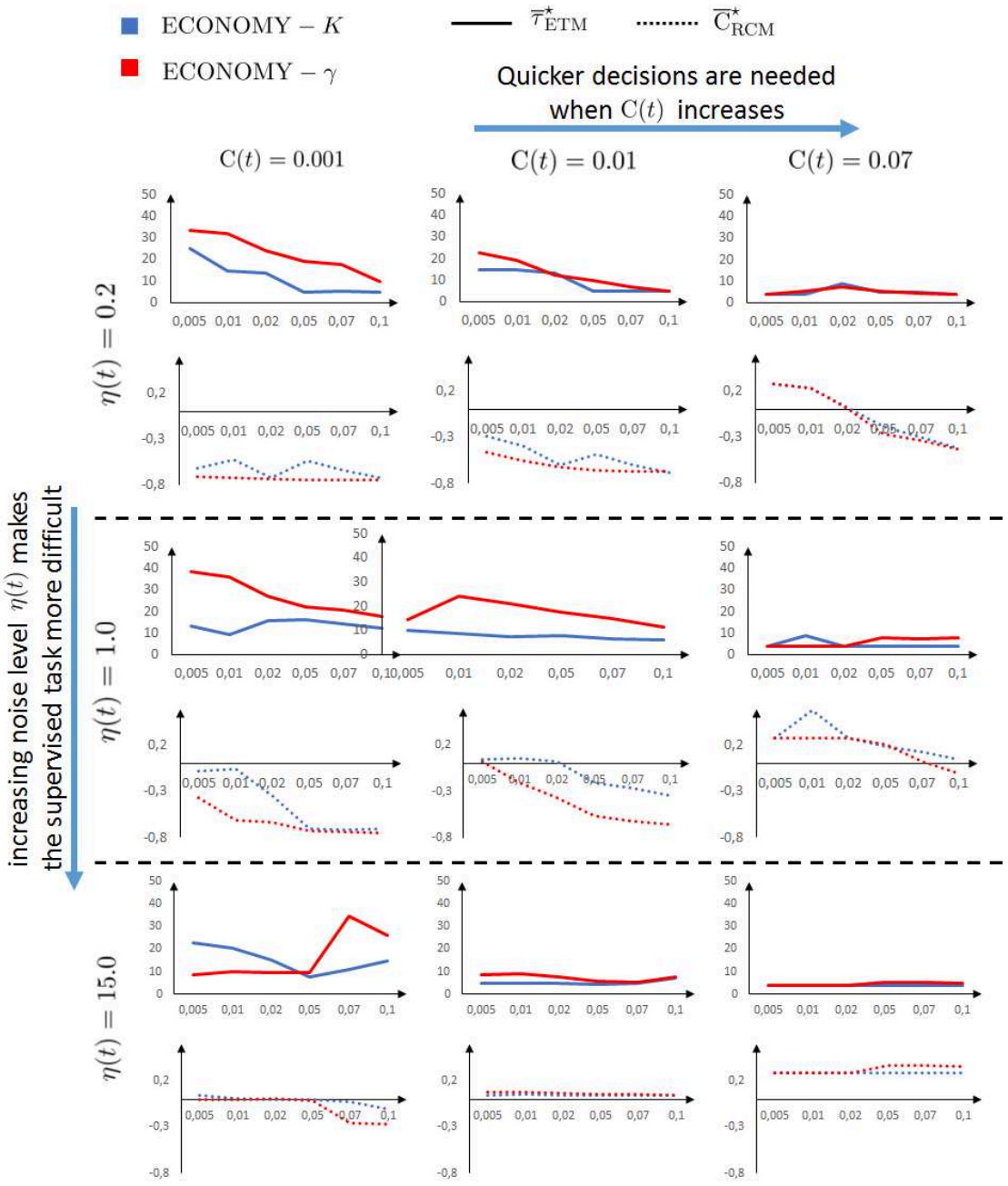


Figure 6.4: **Impact of the rate of information gain** m. The performances of ECONOMY-K vs ECONOMY-γ over the synthetic sine data sets. The x-axis represents the rate of information gain m and the y-axis represents the estimated decision time $\overline{\tau}_{\mathrm{ETM}}$ depicted with solid lines and its associated real cost $\overline{\mathrm{C}}_{\mathrm{RCM}}$ depicted with dashed lines. Results are reported for noise levels $\eta(t) \in \{0.2, 1.0, 15.0\}$, and delaying costs $\mathrm{C}(t) \in \{0.001, 0.01, 0.07\}$.

	±m	0.01			0.02			0.07		
(K_{-1}, K_{+1})	$\eta(t)$	$\overline{\tau}^{\star}_{\text{ETM}}$	$\overline{\text{C}}_{\text{RCM}}$	AUC	$\overline{\tau}^{\star}_{\text{ETM}}$	$\overline{\text{C}}_{\text{RCM}}$	AUC	$\overline{\tau}^{\star}_{\text{ETM}}$	$\overline{\text{C}}_{\text{RCM}}$	AUC
	0.1	11.7±1.6	-0.62	0.93	8.8±1.0	-0.67	0.94	4.0±0.0	-0.71	0.91
	0.2	14.6±2.0	-0.39	0.88	13.5±2.2	-0.6	0.93	5.1±0.4	-0.6	0.86
	0.5	14.8±1.0	-0.22	0.76	13.6±2.2	-0.42	0.86	5.1±0.6	-0.24	0.76
(3,3)	1.0	8.6±1.1	0.06	0.47	7.4±0.9	0.02	0.45	6.7±2.5	-0.26	0.27
	1.5	4.1±0.5	0.04	0.49	13.8±5.2	-0.08	0.33	7.2±3.8	-0.09	0.33
	5.0	4.8±1.1	0.04	0.5	4.6±0.5	0.04	0.5	6.2±3.7	0.03	0.54
	10.0	4.0±0.0	0.04	0.51	4.0±0.0	0.04	0.51	4.2±1.0	0.04	0.51
	15.0	4.9±2.2	0.05	0.5	4.8±1.9	0.04	0.49	4.9±1.8	0.04	0.49
	20.0	4.0±0.0	0.04	0.49	4.0±0.0	0.04	0.49	4.1±1.1	0.04	0.49
	0.1	4.0±0.0	-0.28	0.72	8.7±0.5	-0.63	0.9	5.0±0.0	-0.66	0.88
	0.2	10.7±2.5	-0.39	0.82	6.2±0.5	-0.31	0.75	6.0±0.0	-0.62	0.88
	0.5	4.0±0.0	-0.11	0.63	6.8±1.9	-0.21	0.7	4.0±0.0	-0.4	0.81
(3,5)	1.0	7.2±2.0	-0.02	0.62	6.9±1.4	-0.07	0.65	5.7±1.4	-0.3	0.24
	1.5	4.0±0.1	-0.01	0.55	9.8±2.5	-0.16	0.69	9.5±4.7	-0.32	0.18
	5.0	5.3±0.9	0.02	0.51	4.0±0.0	0.02	0.51	7.6±1.9	0.02	0.58
	10.0	4.1±0.4	0.04	0.49	4.4±0.8	0.05	0.5	5.7±2.6	0.06	0.49
	15.0	4.0±0.0	0.04	0.49	4.0±0.0	0.04	0.49	4.1±0.8	0.04	0.49
	20.0	4.0±0.0	0.04	0.51	4.0±0.0	0.04	0.51	4.0±0.0	0.04	0.48

Table 6.4: **Performances of ECONOMY-K when varying the number of sub-groups in each class.** Results are obtained by varying the noise level $\eta(t)$, the rate of the gain of information m, and the number of sub-groups (K_{-1}, K_{+1}) in each class. The delaying cost $\text{C}(t)$ is fixed to 0.01.

the number N of Markov states (here $N = 5$ and $N = 10$) has no noticeable effect on the results.

6.1.4 Comparison of the methods and interpretation

A first look at the results in Table 6.3 and Figures 6.2, 6.3 and 6.4 shows that:

- For both methods, when the cost of delaying decision increases (from $0.001 \times t$ to $0.1 \times t$), the algorithms decrease the decision time (if one looks for the same noise level $\eta(t)$ in each method in Figure 6.2).
- As the difficulty of the task increases, with mounting noise level (from 0.1 to 20) the algorithms tend to first increase the time of decision, because it is more difficult

	±m	0.01			0.02			0.07		
(K_{-1}, K_{+1})	$\eta(t)$	$\overline{\tau}^{\star}_{\text{ETM}}$	$\overline{\text{C}}_{\text{RCM}}$	AUC	$\overline{\tau}^{\star}_{\text{ETM}}$	$\overline{\text{C}}_{\text{RCM}}$	AUC	$\overline{\tau}^{\star}_{\text{ETM}}$	$\overline{\text{C}}_{\text{RCM}}$	AUC
	0.1	12.1±1.6	-0.64	0.93	10.2±1.2	-0.67	0.94	5.3±1.4	-0.71	0.92
	0.2	19.4±4.1	-0.55	0.92	12.3±2.1	-0.62	0.93	6.7±1.9	-0.67	0.91
	0.5	23.9±9.2	-0.41	0.86	17.7±5.0	-0.54	0.89	11.1±4.5	-0.64	0.93
(3,3)	1.0	24.0±9.9	-0.21	0.24	21.2±10.2	-0.36	0.17	15.0±4.0	-0.62	0.07
	1.5	23.9±10.0	-0.07	0.32	24.4±9.2	-0.28	0.21	17.0±9.2	-0.51	0.13
	5.0	6.5±1.8	0.06	0.5	6.6±2.5	0.06	0.5	26.9±6.6	-0.34	0.81
	10.0	10.4±2.1	0.1	0.51	9.3±2.7	0.09	0.51	18.1±8.4	0.01	0.38
	15.0	8.9±1.8	0.08	0.5	7.7±1.4	0.07	0.5	5.4±0.5	0.05	0.5
	20.0	11.4±4.6	0.11	0.49	8.8±2.0	0.08	0.49	9.7±1.6	0.08	0.48
	0.1	9.6±3.5	-0.61	0.87	9.4±1.8	-0.62	0.9	5.3±1.3	-0.67	0.89
	0.2	12.3±5.3	-0.56	0.85	12.9±2.7	-0.59	0.9	8.1±1.9	-0.63	0.9
	0.5	16.9±9.5	-0.45	0.8	15.3±6.9	-0.54	0.85	10.8±2.2	-0.62	0.91
(3,5)	1.0	19.1±11.3	-0.32	0.76	16.2±8.2	-0.43	0.79	13.9±6.2	-0.7	0.94
	1.5	15.1±9.9	-0.18	0.69	17.4±8.9	-0.35	0.77	16.1±8.2	-0.61	0.92
	5.0	15.5±12.3	0.05	0.58	17.7±9.9	-0.01	0.62	26.5±8.8	-0.47	0.88
	10.0	6.4±0.9	0.06	0.51	5.4±0.9	0.06	0.5	30.9±8.6	-0.18	0.77
	15.0	4.0±0.0	0.04	0.49	4.0±0.0	0.04	0.49	13.9±6.3	0.05	0.57
	20.0	6.4±0.9	0.06	0.51	7.2±1.0	0.07	0.51	5.1±2.1	0.05	0.52

Table 6.5: **Performances of ECONOMY-γ when varying the number of sub-groups in each class.** Results are obtained when varying the noise level $\eta(t)$, the rate of the gain of information m, and the number of sub-groups (K_{-1}, K_{+1}) in each class. The waiting cost $\text{C}(t)$ is fixed to 0.01.

to make a good prediction early on, before deciding that it is not worth waiting, and making a prediction after 4 time steps, which is the minimum amount of time set in our experiments.

- The ECONOMY-γ method tends to delay the "discouragement" phase more than the ECONOMY-K method. This is advantageous for small and medium values of $\eta(t)$ and tends to be slightly disadvantageous when the noise level is high.

Paired statistical t-test for comparison

In order to compare both algorithms, we performed the paired statistical t-test, which is given by:

$$\frac{\overline{d}}{s_d/\sqrt{N}}$$

$(m = 0.01, C(t) = 0.01 \times t)$													
	K=3		K=5		K=6		K=8		K=9		K=15		
$\eta(t)$	$\overline{\tau}^{\star}_{\text{ETM}}$	$\overline{\text{C}}_{\text{ECM}}$	$\overline{\tau}^{\star}_{\text{ETM}}$	$\overline{\text{C}}_{\text{ECM}}$	$\overline{\tau}^{\star}_{\text{ETM}}$	$\overline{\text{C}}_{\text{ECM}}$	$\overline{\tau}^{\star}_{\text{ETM}}$	$\overline{\text{C}}_{\text{ECM}}$	$\overline{\tau}^{\star}_{\text{ETM}}$	$\overline{\text{C}}_{\text{ECM}}$	$\overline{\tau}^{\star}_{\text{ETM}}$	$\overline{\text{C}}_{\text{ECM}}$	$\overline{\text{C}}_{\text{ICM}}$
0.1	5.92	-0.08	11.29	**-0.58**	9.28	**-0.52**	11.68	**-0.62**	11.68	**-0.62**	4.0	-0.12	-0.87
0.2	23.38	**-0.5**	5.68	0.03	5.68	0.03	14.6	**-0.39**	14.59	**-0.39**	14.56	**-0.39**	-0.89
0.5	4.01	0.04	4.03	0.04	4.93	0.05	14.84	**-0.21**	14.76	**-0.22**	16.48	**-0.23**	-0.88
1.0	5.63	**0.03**	5.01	**0.03**	8.4	0.07	8.54	0.06	8.59	0.06	9.54	**0.04**	-0.86
1.5	4.1	0.04	4.0	0.04	4.0	0.04	4.1	0.04	4.09	0.04	4.04	0.04	-0.87
5.0	4.0	0.04	4.03	0.04	4.01	0.04	4.0	0.04	4.76	0.04	4.59	0.04	-0.79
10.0	4.0	0.04	4.0	0.04	4.0	0.04	4.0	0.04	4.0	0.04	4.0	0.04	-0.74
15.0	4.0	0.04	4.0	0.04	4.03	0.04	4.14	0.04	4.87	0.05	4.45	0.04	-0.76
20.0	4.0	0.04	4.0	0.04	4.0	0.04	4.0	0.04	4.0	0.04	4.0	0.04	-0.78
$(m = 0.02, C(t) = 0.01 \times t)$													
	K=3		K=5		K=6		K=8		K=9		K=15		
$\eta(t)$	$\overline{\tau}^{\star}_{\text{ETM}}$	$\overline{\text{C}}_{\text{ECM}}$	$\overline{\tau}^{\star}_{\text{ETM}}$	$\overline{\text{C}}_{\text{ECM}}$	$\overline{\tau}^{\star}_{\text{ETM}}$	$\overline{\text{C}}_{\text{ECM}}$	$\overline{\tau}^{\star}_{\text{ETM}}$	$\overline{\text{C}}_{\text{ECM}}$	$\overline{\tau}^{\star}_{\text{ETM}}$	$\overline{\text{C}}_{\text{ECM}}$	$\overline{\tau}^{\star}_{\text{ETM}}$	$\overline{\text{C}}_{\text{ECM}}$	$\overline{\text{C}}_{\text{ICM}}$
0.1	4.96	**-0.36**	8.8	**-0.67**	4.96	**-0.36**	8.8	**-0.67**	8.8	**-0.67**	4.0	-0.35	-0.8
0.2	9.94	**-0.58**	5.2	-0.11	9.97	**-0.58**	13.91	**-0.6**	13.49	**-0.6**	8.8	-0.5	-0.86
0.5	4.93	0.04	4.92	0.04	4.93	0.04	13.9	**-0.44**	13.64	-0.42	13.66	**-0.43**	-0.87
1.0	5.98	0.02	9.7	**-0.08**	6.77	**0.0**	7.32	0.04	7.44	0.02	7.15	0.03	-0.88
1.5	4.69	0.05	6.17	0.05	6.38	0.05	4.42	0.03	13.85	**-0.08**	14.25	**-0.11**	-0.89
5.0	4.55	0.05	4.33	0.04	4.17	0.04	4.39	0.04	4.58	0.04	4.79	0.04	-0.82
10.0	4.0	0.04	4.0	0.04	4.0	0.04	4.0	0.04	4.0	0.04	4.0	0.04	-0.77
15.0	4.0	0.04	4.01	0.04	4.12	0.04	4.39	0.04	4.84	0.04	4.34	0.04	-0.8
20.0	4.0	0.04	4.0	0.04	4.0	0.04	4.0	0.04	4.0	0.04	4.0	0.04	-0.81

Table 6.6: Impact of varying the number of clusters on results of the ECONOMY-K approach.

where $\overline{d}$ is the difference between the two observations in each pair, s_d is the standard deviation of the differences and $N = 225$, the number of examples. We compared the estimated costs given by both systems when following their decision policies: $\overline{\text{C}}_{\text{RCM}}$. The question was: is one algorithm significantly superior to the other in the experimental setting?

(m = 0.01, C(t) = 0.01 × t)					
	N=5		N=10		
$\eta(t)$	$\overline{\tau}^{\star}_{\text{ETM}}$	$\overline{\text{C}}_{\text{ECM}}$	$\overline{\tau}^{\star}_{\text{ETM}}$	$\overline{\text{C}}_{\text{ECM}}$	$\overline{\text{C}}_{\text{ICM}}$
0.1	12.08	**-0.64**	12.9	**-0.64**	-0.87
0.2	19.37	**-0.55**	17.54	**-0.53**	-0.89
0.5	23.86	**-0.41**	24.98	**-0.42**	-0.88
1.0	23.98	**-0.21**	24.15	**-0.23**	-0.86
1.5	23.91	**-0.07**	27.03	**-0.08**	-0.87
5.0	6.5	**0.06**	5.98	**0.05**	-0.79
10.0	10.42	**0.1**	11.9	**0.11**	-0.74
15.0	8.87	**0.08**	8.12	**0.07**	-0.76
20.0	11.37	**0.11**	6.49	**0.06**	-0.78
(m = 0.02, C(t) = 0.01 × t)					
	N=5		N=10		
$\eta(t)$	$\overline{\tau}^{\star}_{\text{ETM}}$	$\overline{\text{C}}_{\text{ECM}}$	$\overline{\tau}^{\star}_{\text{ETM}}$	$\overline{\text{C}}_{\text{ECM}}$	$\overline{\text{C}}_{\text{ICM}}$
0.1	10.21	**-0.67**	9.92	**-0.67**	-0.8
0.2	12.33	**-0.62**	13.62	**-0.61**	-0.86
0.5	17.7	**-0.54**	18.43	**-0.53**	-0.87
1.0	21.24	**-0.36**	21.4	**-0.36**	-0.88
1.5	24.38	**-0.28**	26.01	**-0.31**	-0.89
5.0	6.64	**0.06**	7.69	**0.07**	-0.82
10.0	9.33	**0.09**	9.27	**0.09**	-0.77
15.0	7.67	**0.07**	8.96	**0.09**	-0.8
20.0	8.81	**0.08**	7.58	**0.07**	-0.81

Table 6.7: Impact of varying the number of Markov states on results of the ECONOMY-γ approach.

Table 6.8 gives the results for the paired t-test when we consider the difference:

$$\overline{\text{C}}_{\text{RCM}}(\text{ECONOMY-}K) - \overline{\text{C}}_{\text{RCM}}(\text{ECONOMY-}\gamma)$$

For small and medium delaying costs, the confidence-based method ECONOMY-γ is significantly better than the clustering-based method ECONOMY-K (largely above the significance level for $\alpha = 0.05\%$). Nothing can be said one way or the other for C(t) = 0.07, while ECONOMY-K is better for C(t) = 0.1, because it does not wait to

Paired t-test	$\mathbf{C}_{\text{delay}}(t)$				
	0.001	0.01	0.05	0.07	0.1
t-statistic	**6.68**	**5.03**	**2.77**	1.31	**-2.42**

Table 6.8: Comparison of ECONOMY-K vs ECONOMY-γ. The paired t-test is computed using the real costs $\overline{\text{C}}_{\text{RCM}}$ incurred by each algorithm on 225 synthetic data sets for 5 different delay costs. Here, $\alpha = 0.05$ and the degree of freedom $df = 1.960$.

make a decision.

These results show that the confidence-based method ECONOMY-γ, even in its baseline implementation, is clearly superior to the clustering-based method ECONOMY-K when the delaying cost is rather low. For higher costs, ECONOMY-K decides earlier.

We also compared the proximity of the real costs $\overline{\text{C}}_{\text{RCM}}$, given by each algorithm, with the optimal costs $\overline{\text{C}}_{\text{ICM}}$ given by the omniscient algorithm. In Table 6.9, we provide the results for the paired t-tests when we respectively consider the differences $\overline{\text{C}}_{\text{RCM}}$(ECONOMY-$K$)-$\overline{\text{C}}_{\text{ICM}}$(perfect algorithm) and $\overline{\text{C}}_{\text{RCM}}$(ECONOMY-$\gamma$)-$\overline{\text{C}}_{\text{ICM}}$(perfect algorithm).

Paired t-test	$\mathbf{C}_{\text{delay}}(t)$				
	0.001	0.01	0.05	0.07	0.1
t-statistic (ECONOMY-K vs critère idéal)	-12.40	-16.79	-16.70	**-14.34**	**-11.64**
t-statistic (ECONOMY-γ vs critère idéal)	**-9.51**	**-11.91**	**-16.53**	-14.86	-12.07

Table 6.9: Comparison of ECONOMY-K vs ECONOMY-γ according to the proximity to the perfect algorithm. Paired t-tests over the real costs $\overline{\text{C}}_{\text{RCM}}$ incurred by ECONOMY-K (resp. ECONOMY-γ) and the optimal costs $\overline{\text{C}}_{\text{ICM}}$ are computed on 225 synthetic data sets for 5 different delay costs. Here $\alpha = 0.05$ and degree of freedom $df = 1.960$.

From Table 6.9, we observe that even if the obtained results from both approaches are not close to ideal ones, again, the confidence-based method ECONOMY-γ significantly better approximates the optimal decision time.

Another mean to compare the methods is to look at the highest level of noise $\eta(t)$ for which a method yields a cost that is better than the cost incurred when deciding at

the first possible moment (4 in these experiments) with a margin of at least 0.1. For instance, (see Table 6.3), for m = 0.07 and C(t) = 0.001, $\overline{\text{C}}_{\text{ECM}} \leq 0.004 - 0.1$ up to noise level $\eta(t) = 5$ (for which $\overline{\text{C}}_{\text{ECM}} = -0.1$ for ECONOMY-K), while this is true up to $\eta(t) = 20$ for ECONOMY-γ. Thus, for m = 0.07 and C(t) = 0.001, the ECONOMY-K is significantly winning, according to our rule, for 5 values of noise levels, while the confidence-based method ECONOMY-γ is winning for all the 7 noise levels reported in the experiments.

This comparison, for all values of m and values of C(t) can be expressed as the histogram of Figure 6.5. It is apparent that the confidence-based method ECONOMY-γ is winning in all the situations in which the clustering-based method ECONOMY-K wins, plus others. It thus brings significant gains in a wider spectrum of situations than the ECONOMY-K method.

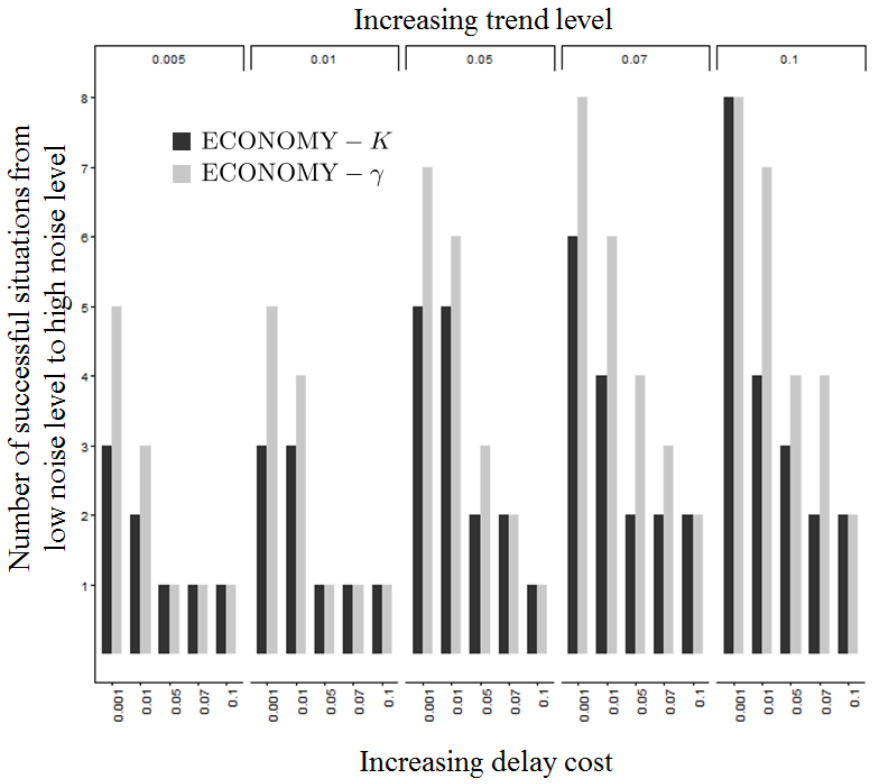


Figure 6.5: Histogram over synthetic data sets showing the number of noise levels for which each method brings a significant gain as compared to the earliest possible decision.

Individual behaviors

The above results aggregate the measures on the whole testing set. It is interesting to look as well at individual behaviors. For instance, Figure 6.6 shows the expected costs $f_\tau(\mathbf{x}_t)$ for the same incoming time series $\mathbf{x}_t$, for each of the potentially remaining time steps $\tau \in \{0, \ldots, T-t\}$. The delaying cost $C(t)$ being fixed to 0.01, we observe, that as at the global level, ECONOMY-γ waits longer before making a decision yielding thus better costs compared with ECONOMY-K approach that decides earlier.

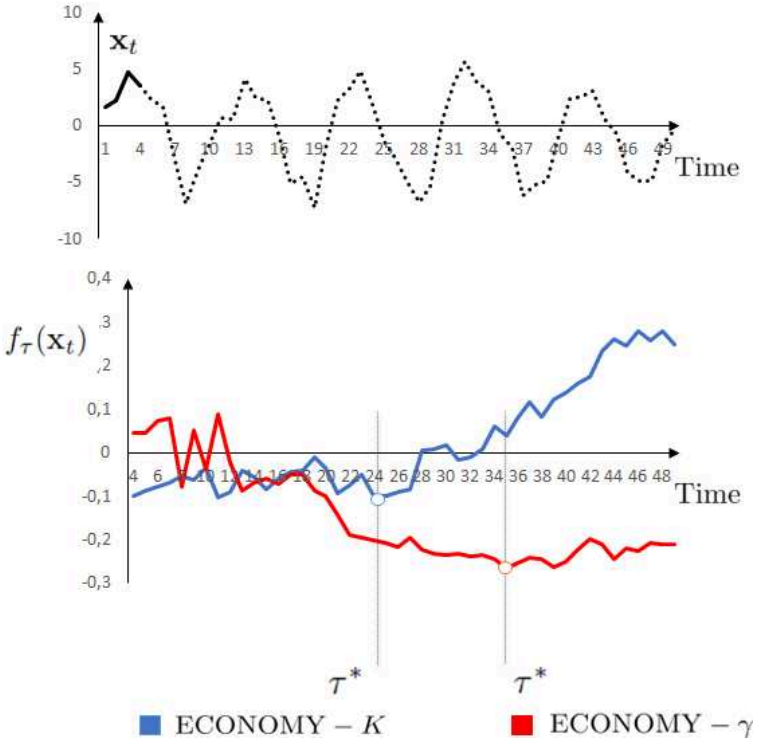


Figure 6.6: For the same incoming time series $\mathbf{x}_t$ (top figure), the expected costs (bottom figure) obtained from ECONOMY-K and ECONOMY-γ approaches are different. Their minima have different values and occur at different instants. Here, the delaying cost $C(t) = 0.01 \times t$.

6.2 Experimental evaluation on real-like data sets

The goal here is to test the methods with real like data sets. For this, we choose data sets from the UCR Time-Series Classification/Clustering Repository [62].

6.2.1 Real data

A variety of real data sets from the UCR repository [62] are provided from a wide range of application domains with different characteristics. The data sets contain different number of time series with different lengths. The classification tasks within these data sets are also different in terms of the number of target classes.

In this thesis, since we treat only binary classification problems, the data sets used in the experiments correspond to real binary classification problems. As an example, the data set *FordA* includes time series collected from an automotive subsystem. Each example in the data set consists of a time series composed of 500 data points recording the engine noise and a label describing the diagnostic result according to a certain symptom. Class $+1$ is associated with the diagnosis that the symptom exists and -1 with the diagnosis that the symptom does not exist. The complete list of data sets on which we experiment the methods are summarized in Table A.8.

Originally, the data sets were provided with two separate train and test sets. As we need three sets in our algorithm ($\mathcal{S}_1$, $\mathcal{S}_2$ and $\mathcal{T}$), with $\mathcal{S}_1$ is the training set, $\mathcal{S}_2$ is the validation set and $\mathcal{T}$ is the test set, we combined the original train and test sets and then randomly divided the total into three sets of the same size. For data sets of small sizes (less than 500 time series), 10-fold cross-validation is used.

Since for all data sets costs are not specified, we set the misclassification costs to $\mathrm{C}(\hat{y}|y) = +1$ if $\hat{y} = y$ and -1 if $\hat{y} \neq y$. And, to make the early-decision tasks, we set cost functions for delaying decisions as an increasing cost function: $\mathrm{C}(t) = \mathrm{d} \times t$, where $\mathrm{d} \in \{0.001, 0.01, 0.05, 0.07, 0.1\}$ ranging the delay cost from low to high values.

6.2.2 Empirical results

We choose to report here an abbreviated table 6.10 that depicts the results of 5 real data sets: DistalPhalanxOC, ECGFiveDays SonyAIRobotSII, Strawberry and TwoLeadECG. The complete results are available in Tables A.6 and A.7.

From Table 6.10, we notice that both approaches show the behavior expected from an early decision system on real data sets as it was vindicated on synthetic data sets, viz: (i) the time of decision decreases when the delaying cost function increases, and (ii) the system decides that it is not worth waiting and it is better to make a decision at the

Data	C(t)	ECONOMY-K				ECONOMY-γ					
		$\overline{\tau}^{\star}_{\text{ETM}}$	$\overline{\text{C}}_{\text{RCM}}$	$\overline{\tau}^{\star}_{\text{PETM}}$	$\overline{\text{C}}_{\text{PRCM}}$	$\overline{\tau}^{\star}_{\text{ETM}}$	$\overline{\text{C}}_{\text{RCM}}$	$\overline{\tau}^{\star}_{\text{PETM}}$	$\overline{\text{C}}_{\text{PRCM}}$	$\overline{\tau}^{\star}_{\text{ITM}}$	$\overline{\text{C}}_{\text{ICM}}$
DistalPhalanxOC	0.001	8.0±0.0	-0.31	9.0±1.4	-0.34	21.7±13.1	-0.42	54.7±14.0	-0.44	14.1±13.9	-0.88
	0.01	8.0±0.0	-0.23	8.6±0.9	-0.25	10.9±2.5	-0.25	13.8±3.5	-0.23	14.1±13.9	-0.76
	0.05	8.0±0.0	0.09	8.2±0.4	0.09	8.9±1.9	0.11	9.2±2.1	0.11	10.3±6.4	-0.22
	0.07	8.0±0.0	0.25	8.2±0.4	0.25	8.0±0.0	0.25	8.2±0.9	0.22	9.8±5.2	-0.02
	0.1	8.0±0.0	0.49	8.1±0.4	0.48	8.0±0.0	0.49	8.1±0.4	0.48	8.9±2.5	0.25
ECGFiveDays	0.001	16.5±4.1	-0.6	16.9±3.1	-0.61	58.7±16.2	-0.92	63.0±15.2	-0.91	16.8±10.7	-0.98
	0.01	13.0±0.0	-0.51	15.8±1.6	-0.47	17.9±9.4	-0.5	23.7±10.1	-0.44	16.8±10.7	-0.83
	0.05	13.0±0.0	0.01	14.9±1.4	0.12	13.2±0.4	0.03	13.9±1.4	0.05	15.1±6.2	-0.18
	0.07	13.0±0.0	0.27	14.1±1.4	0.35	13.0±0.0	0.27	13.2±0.6	0.3	14.7±5.2	0.12
	0.1	13.0±0.0	0.66	13.4±1.0	0.71	13.0±0.0	0.66	13.1±0.4	0.69	14.1±3.9	0.55
SonyAIBORobotS	0.001	19.5±5.1	-0.76	12.0±4.9	-0.72	14.8±11.7	-0.77	25.5±16.1	-0.82	8.0±4.5	-0.99
	0.01	7.0±0.0	-0.71	9.6±1.7	-0.63	10.1±5.5	-0.66	11.9±4.1	-0.62	8.0±4.5	-0.92
	0.05	7.0±0.0	-0.43	8.3±1.3	-0.36	8.3±2.1	-0.36	9.3±2.2	-0.28	7.8±3.8	-0.6
	0.07	7.0±0.0	-0.29	7.8±1.1	-0.22	8.3±2.1	-0.2	8.9±2.2	-0.14	7.7±3.0	-0.45
	0.1	7.0±0.0	-0.08	7.3±0.6	-0.04	8.1±2.1	0.03	8.4±2.1	0.06	7.5±2.1	-0.22
SonyAIBORobotSII	0.001	7.9±3.0	-0.56	10.1±3.8	-0.72	16.7±12.2	-0.86	23.3±16.2	-0.86	7.0±2.5	-0.99
	0.01	6.3±0.7	-0.52	8.0±2.3	-0.6	13.7±6.0	-0.72	13.8±5.3	-0.74	7.0±2.5	-0.92
	0.05	6.0±0.0	-0.28	7.0±1.2	-0.23	6.0±0.0	-0.28	8.1±2.9	-0.31	7.0±2.5	-0.65
	0.07	6.0±0.0	-0.16	6.6±1.0	-0.14	6.0±0.0	-0.16	6.7±1.7	-0.16	7.0±2.5	-0.51
	0.1	6.0±0.0	0.02	6.3±0.7	0.03	6.0±0.0	0.02	6.3±1.0	0.02	6.9±2.1	-0.3
Strawberry	0.001	25.8±0.4	-0.4	25.8±1.3	-0.37	51.8±38.8	-0.85	57.0±17.4	-0.86	26.7±9.0	-0.96
	0.01	24.6±1.4	-0.16	25.7±1.1	-0.14	37.0±11.3	-0.41	38.3±11.2	-0.41	26.7±9.0	-0.72
	0.05	23.0±0.0	0.73	25.0±0.5	0.84	23.9±1.9	0.76	24.8±2.4	0.79	26.1±6.5	0.34
	0.07	23.0±0.0	1.19	24.8±0.7	1.33	23.0±0.0	1.19	23.4±1.2	1.22	25.6±5.3	0.86
	0.1	23.0±0.0	1.88	24.7±0.7	2.05	23.0±0.0	1.88	23.2±0.8	1.9	25.0±4.1	1.62
TwoLeadECG	0.001	8.0±0.0	-0.36	17.7±2.9	-0.83	25.1±9.0	-0.95	26.8±9.3	-0.94	9.0±2.1	-0.99
	0.01	8.0±0.0	-0.29	16.1±3.4	-0.64	15.8±5.6	-0.76	17.1±5.6	-0.75	9.0±2.1	-0.91
	0.05	8.0±0.0	0.03	8.6±2.2	0.06	10.1±2.5	-0.14	10.3±2.6	-0.13	9.0±2.1	-0.55
	0.07	8.0±0.0	0.19	8.1±1.1	0.2	8.7±1.6	0.13	9.5±1.9	0.06	9.0±2.1	-0.37
	0.1	8.0±0.0	0.43	8.0±0.2	0.43	8.0±0.0	0.43	8.9±1.5	0.38	9.0±2.1	-0.1

Table 6.10: Experimental results of ECONOMY-K vs ECONOMY-γ approaches over UCR real data sets.

first possible moment even though the quality of the decision might be quite low. From Figure 6.7, one can observe that over all the real data sets used in this experiment and for low values of delay cost C(t), ECONOMY-γ approach outperforms ECONOMY-K approach. In fact, ECONOMY-γ tends to wait further than ECONOMY-K which yields more benefit.

Wilcoxon Signed-Rank Test for comparison

In order to compare both methods ECONOMY-K and ECONOMY-γ, we used the Wilcoxon Signed-Rank test since we had only 11 data sets over which the performances of the methods were measured.

$\mathbf{C}_{\text{delay}}(t)$	0.001	0.01	0.05	0.07	0.1
z	**63**	**51**	**23**	10	10

Table 6.11: Wilcoxon Signed-Rank Test over the real data sets with $\alpha = 0.05$, $n-1 = 10$ degrees of freedom and a significance level 8.

Table 6.11 provides the results for the Wilcoxon Signed-Rank Test over the real 11 data sets with $\alpha = 0.05$, $n-1 = 10$ degrees of freedom and a significance level equals to 8. The test considers the differences between the real incurred costs $\overline{\text{C}}_{\text{RCM}}$ estimated by both methods. Similarly to the results obtained using the synthetic data sets, the Confidence-based method ECONOMY-γ is remarkably performing better than the Clustering-based method ECONOMY-K (largely above the significance level for $\alpha = 0.05$).

Impact of varying the number of clusters in ECONOMY-K approach and the number of Markov states in ECONOMY-γ approach

	$\overline{\tau}^{\star}_{\text{ETM}}$	$\overline{\text{C}}_{\text{RCM}}$	$\overline{\tau}^{\star}_{\text{PETM}}$	$\overline{\text{C}}_{\text{PRCM}}$
#Quantiles	ECONOMY-γ			
5	16.6±4,3	**-0.907**	17.3±3.48	-0.82
10	17.1±3,8	**-0.901**	18.3±3.9	-0.85
#Clusters	ECONOMY-K			
5	5.7±1.0	**-0.13**	14.5±7.1	-0.64
10	8.32±6.7	**-0.51**	8.60±6.7	-0.516

Table 6.12: Impact of varying the number of clusters (ECONOMY-K) and the number of Markov states (ECONOMY-γ) over the ItalyPowerDemand real data set.

Table 6.12 shows the impact of varying the number of clusters K in ECONOMY-K approach and the number of Markov states N in ECONOMY-γ approach over the ItalyPowerDemand real data set. Results of the optimal time decision and the cost incurred

	ECONOMY-K				ECONOMY-γ				
# groupes	K=5		K=10		K=5		K=10		
Bases	$\overline{\tau}^{\star}_{\text{ETM}}$	$\overline{\text{C}}_{\text{RCM}}$	$\overline{\tau}^{\star}_{\text{ETM}}$	$\overline{\text{C}}_{\text{RCM}}$	$\overline{\tau}^{\star}_{\text{ETM}}$	$\overline{\text{C}}_{\text{RCM}}$	$\overline{\tau}^{\star}_{\text{ETM}}$	$\overline{\text{C}}_{\text{RCM}}$	$\overline{\text{C}}_{\text{ICM}}$
ItalyPowerDemand	5.7	**-0.13**	8.32	**-0.51**	16.6	**-0.907**	17.1	**-0.901**	-0.99
DistalPhalanxOC	19.90	**-0.41**	8.0	**-0.31**	21.73	**-0.424**	19.03	**-0.424**	-0.88
ECGFiveDays	14.8	**-0.61**	20.64	**-0.58**	58.72	**-0.917**	54.43	**-0.92**	-0.98
MoteStrain	63.18	**-0.63**	41.51	**-0.7**	15.72	**-0.69**	15.28	**-0.69**	-0.96

Table 6.13: Impact of varying the number of clusters (ECONOMY-K) and the number of Markov states (ECONOMY-γ) over the ItalyPowerDemand real data set.

by both methods are given when varying $K \in \{\in 5, 8, 12, 15, 16\}$ and $N \in \{5, 10\}$.

One important observation is that, for the ECONOMY-K approach, the number of clusters K heavily influences the results. By contrast, the number N of Markov states (here $N = 5$ and $N = 10$) has no noticeable effect on the results.

Individual behaviors

The above results obtained by applying the proposed approaches over real data sets, aggregate the measures on the whole testing set. It is interesting to look as well at individual behaviors.

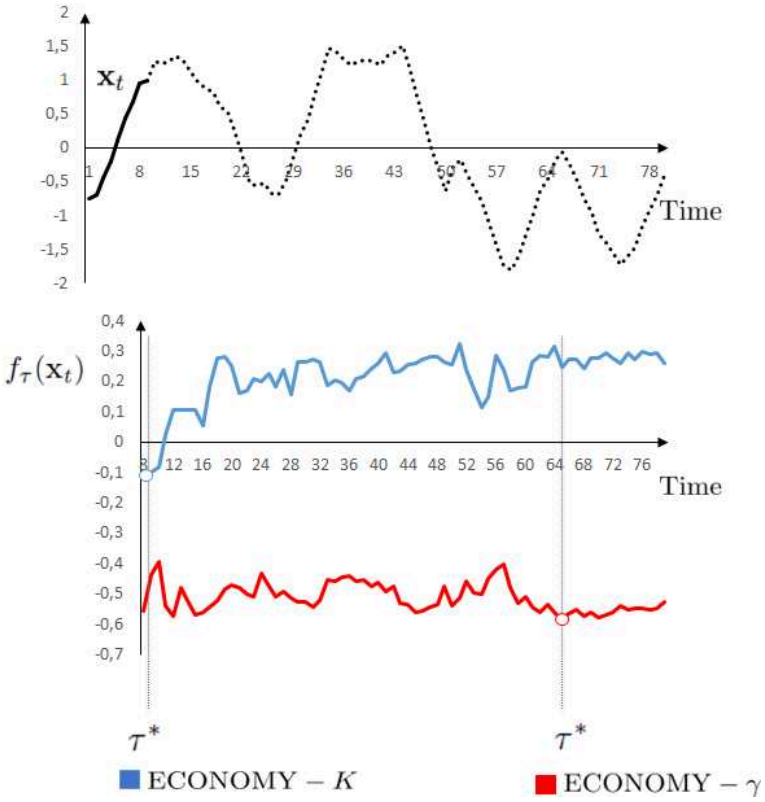


Figure 6.8: For the same incoming time series $\mathbf{x}_t$ (top figure), selected from the real data set *DistalPhalanxOutlineCorrect*, the expected costs (bottom figure) obtained from ECONOMY-K and ECONOMY-γ approaches are different. Their minima have different values and occur at different instants.

For instance, Figure 6.8 shows the expected costs $f_\tau(\mathbf{x}_t)$ for the same incoming time series $\mathbf{x}_t$, selected from the real data set *DistalPhalanxOutlineCorrect*, for each of the potentially remaining time steps $\tau \in \{0, \dots T - t\}$. For a fixed delaying cost ($\mathrm{C}(t) = 0.01$), we observe, that ECONOMY-K decides earlier (even at the first instant), however ECONOMY-γ waits longer before making a prediction yielding thus more benefit compared with ECONOMY-K approach. Here, again, we observe the same behavior as the one obtained on synthetic data sets.

6.3 Towards applying ECONOMY-K and ECONOMY-γ on individual electricity demand

This thesis is carried out under the CIFRE agreement (industrial agreement for training through research) at *Electricité de France* EDF. As part of the smart grid project and energy transition, the first objective of the thesis was to reduce peak energy demand through customers targeting. The idea is to improve peak curtailment by focusing efforts and resources on those customers most likely to participate to the daily peak demand. It is noteworthy that the daily peak demand in France is triggered only few days annually when extreme cold leads to high demand for electric heating. During these periods, customers are encouraged to reduce their consumption by changing electricity prices that increase during periods of high demand and decline during periods of low demand. Trials have shown that customers that are aware of the high price of peak electricity usage tend to consume less during periods of high demand (e.g. running dishwashers at 9 pm instead of 6 pm). This plan can be more effective if customers are targeted as early as possible during the day when peak demand is expected. Communication and metering technologies can inform smart devices in targeted households when energy demand is high and encourage them to reduce their consumption.
Targeting customers for peak curtailment should then be performed, as early as possible during the day, with respect to expected peak demand days. However, bringing in practice this plan using our early classification framework is not trivial and entails confronting two substantial difficulties:

1. The first difficulty is that properties of individual electricity demand are not controlled which makes it difficult to evaluate results when applying ECONOMY-K and ECONOMY-γ.

2. The second difficulty is to set the cost matrix where each cell of the matrix roughly corresponds to a different trade within EDF.

Regarding the first difficulty, we succeeded to resolve it by simulating data that have the advantage of resembling to individual electricity demand and have the same properties of real data, in addition to being controlled. However, estimating the elementary costs, in the cost matrix needs: to involve different trades, to standardize their respective costs and probably to build models to estimate these costs from some available information. For all these reasons, and since it needs huge efforts to set the cost matrix, we did not delve into this since it is not the crux objective of this thesis. Setting the cost matrix in the specific case of electricity demand seems difficult, but this does not exclude that in

other domains this may be easy to obtain.
In the following, we give a brief description of the framework and properties of the simulated data as part in energy demand. Then, we present a possible scenario of two classification tasks on which ECONOMY-K and ECONOMY-γ can be evaluated. In a future work, it would be interesting to estimate the cost matrix and make meaningful predictions using ECONOMY-K and ECONOMY-γ early classification approaches.

6.3.1 Realistic simulation of individual electricity consumption

Synthetic data were often deemed unrealistic mainly because they lack the underlying properties and content of real data. Using realistically simulated data with a known ground truth provides hence a good baseline for evaluating the considered algorithms and makes easy the transition for applying these algorithms on real data.
From this, we suggest that it would be interesting to test our proposed approaches on data generated using the realistic and very fast simulator we contributed in [16]. In fact, we suggested a new approach for realistic and very fast data generation. In a nutshell, the so-called MODL-Markov generative model of time series used in this approach, proposes to efficiently simulate realistic individual electricity consumption data by combining Markov chains and co-clustring models. The main idea is to partition the training time series using the MODL co-clustring approach [18] in order to construct as much as diverse clusters of individual behaviors. Then, Markov chains are build on each cluster in order to learn the dynamic temporal correlations within time series. We refer the interested reader to [16] for more details on this approach.

Using the time series simulation model proposed in [16], we generate data sets based on the real world database provided by the Irish CER[1]. This real data set originates from a smart metering trial in 4600 households across the Ireland. The individual electricity consumption of each household were recorded during 500 days at 30-minute intervals.
The generator being learned over this real data set makes it possible to simulate as much as needed realistic and very large quantities of data in few timings. The simulated data set used in our experiments has the following characteristics:

- 10.000 meters are used to simulate daily electricity consumption.
- 55 models are learned on the training real data.
- 152 clusters are provided using the MODL co-clustering approach.

[1] CER(Commity of Energy Regulation) Smart Metering Trial Data Publication, 2012.

- Each cluster includes the electricity consumption provided with the meter id and the date including the day in the year and the timestamp.
- The generated time series are not labeled.

6.3.2 Two possible supervised tasks

The idea behind using such a data set is (i) to evaluate the proposed approaches on data that are very similar to real electricity individual consumption data, and (ii) to examine the impact of varying the difficulty of the task on the decision time $\overline{\tau}^{\star}_{\text{ETM}}$ and the real cost $\overline{\text{C}}_{\text{RCM}}$. Varying the difficulty of the task corresponds to the parameter $\eta(t)$ varied in the synthetic data .
It is expected that the decision time increases as the task is more difficult and decreases if the task is less difficult. To confirm these properties which are desired from early classification algorithms and in order to check the validity of the conclusions obtained from synthetic data sets, two binary classification problems could be considered:

1. **Classification problem 1:** In the first problem, as the total data is optimally partitioned into 152 clusters, we propose to compose the training set of time series using the two most distant clusters. Actually, a similarity measure or a distance can be used to determine these clusters. We build a supervised classification task by labeling time series in a one cluster with $+1$ class and time series in the other cluster are thus labeled with -1 class. This setting supposes that the classes are well discriminated as time series from both clusters are well separated. This potentially makes the classification task less difficult to solve.
2. **Classification problem 2:** In the second problem, time series in the two closest clusters are used to train the classifier. The time series are then labeled according to their respective cluster as described earlier. Here, since time series from the two clusters are close, their classes may be confused. This potentially makes the classification task difficult when discriminating between the classes.

To measure the the similarity between a pair of clusters, the Kullbak-Leibler Divergence (KLD) can be used. It is defined as:

$$D_{KL}(P,Q) = \sum_{i} P(i) \log \left(\frac{P(i)}{Q(i)}\right) \tag{6.2}$$

Where P and Q are two discrete probability distributions.

Summary

In this chapter, we have applied our contributions (see Chapter 5), two different cost-sensitive online decision algorithms (named ECONOMY-K and ECONOMY-γ), to different classification tasks.

We have examined their performances in different environments by using synthetic and real data sets, compared their behaviors when varying some parameters that impact the decision making process and concluded that the obtained results meet the specific properties and behaviors expected from early decision systems.

Specifically, both proposed ECONOMY-K and ECONOMY-γ approaches, which try to solve the same generic optimization criterion ECONOMY (see Section 5.2), using two different segmentation techniques, yield experimental results that remarkably agree with what one would expect from an early decision system, viz:

- The time of decision rises when the classification task is increasingly harder (for instance, if the data is increasingly noisy), up until a point when, given the difficulty of extracting information from the signal, the systems know from their experience that classification gains in the future cannot overcome the delaying cost, thus deciding that it is not worth waiting and it is better to make decision at the first possible moment even though the quality of this decision might be quite low.
- The time of decision decreases when the delay cost function increases more rapidly with time.

Furthermore, ECONOMY-γ approach shows significant better results than ECONOMY-K on the synthetic and real data sets. We can interpret this by the fact that the information about the class labels of time series are taken into account when segmenting leads to groups of time series that significantly differs one from the other. Consequently, this ensures confusion matrices that are significantly different one from the other. This is less clear regarding the confusion matrices obtained by a clustering technique in the ECONOMY-K approach.
Finally, we have described a use case for applying ECONOMY-K and ECONOMY-γ on individual electricity demand and shown difficulties of such an application regarding the setting of the cost matrix and the control of data properties. For this latter, we have proposed a simulation model that has the advantage of generating data having the same properties of individual electricity demand and suggested two supervised problems on which ECONOMY-K and ECONOMY-γ could be applied and evaluated.

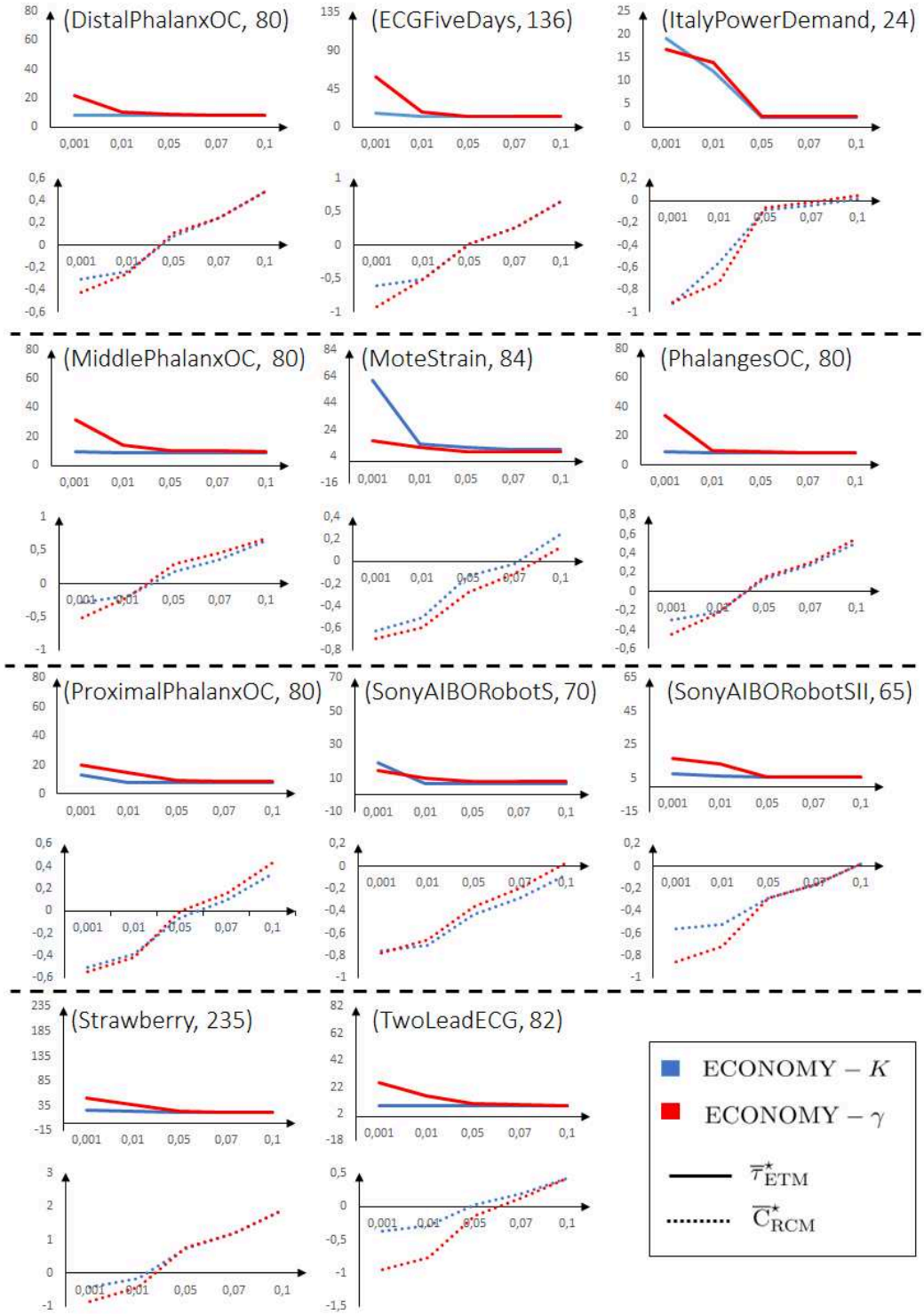


Figure 6.7: The performance of ECONOMY-K vs ECONOMY-γ over 11 real data sets from UCR archive (lengths of time series in each data set are also specified). The x-axis represents the delaying cost $\mathrm{C}(t)$ and the y-axis represents the estimated decision time $\overline{\tau}_{\mathrm{ETM}}$ depicted with solid lines and its associated real cost $\overline{\mathrm{C}}_{\mathrm{RCM}}$ depicted with dashed lines.

Chapter 7

Conclusion and Perspectives

This thesis investigates the early classification of time series problem. Our main contribution is to provide solutions to a clearly identified problem: the optimization of the trade-off between the quality and the earliness of predictions when delaying the decision is costly.

In the next sections, we summarize our major contributions and emphasize the main properties of the adaptive and non-myopic cost-sensitive online decision making framework (ECONOMY). We then point out some future works including some areas of our work extensions and possible perspectives.

7.1 Contributions

We have made a number of novel contributions to time series early classification problem:

- We have explicitly identified two sub-problems for making early predictions: (i) classifying incomplete time series, and (ii) estimating online the optimal time for making a prediction. These two sub-problems are independent. While the first concerns the implementation of early classifiers and examines their ability to label incomplete time series, the second deals with strategies for online decision making.

- We have proposed an early classification framework that explicitly endows early classifiers with a decision function (named $\mathcal{T}rigger$) that decides the time for making a prediction. The way $\mathcal{T}rigger$ is implemented allows one to make a fair comparison with the state-of-the-art early classification methods.

- We have cast the early classification problem to a cost-sensitive online decision making problem and proposed a generic optimization criterion that explicitly takes

into account the cost of delaying the decision, along with misclassification costs. This new formalization allows one to optimize the gain of information that, that is expected to incur lower misclassification costs when delaying the decision, against the cost of such a delay.

- We have proposed two algorithms, ECONOMY-K and ECONOMY-γ that use different segmentation techniques and implement the generic optimization criterion, along with estimating, online, the optimal future time for triggering the predictions and offer adaptive and non-myopic decisions.
- We have tested the proposed algorithms on synthetic and real world data sets and shown that they meet behaviors expected from early classification systems.

7.2 Research methodology

In the following, we describe the basic ideas behind our research methodology and try to point out the choices that we were led to make and their potential impacts.

7.2.1 The question of the thesis

In this thesis, we described the problem of making early classifications of time series in terms of the following question: how to decide, online, that now is the optimal time to make a prediction given an incoming yet incomplete time series $\mathbf{x}_t$? We answered this question by proposing a new formalization that explicitly takes into account the cost of delaying the decision and optimizes the time against the gain of information trade-off.

7.2.2 Ideal early decision rule

We started by assuming that in an ideal framework, the optimal time τ_{ideal} for classifying a time series $\mathbf{x}_t$ would be decided by an ideal (unrealistic) early decision function, defined as:

$$f_t^{\text{ideal}}(\mathbf{x}_t) = \text{C}(h_t(\mathbf{x}_t) = y|y) + \text{C}(t) \tag{7.1}$$

where $h_t(\mathbf{x}_t)$ is the prediction output for the class label of $\mathbf{x}_t$ using a classifier h_t (learnt over training time series trimmed to their t first data points).

The ideal time to make a prediction is thus:

$$\tau_{\text{ideal}} = \underset{t\in\{1,\dots,T\}}{\text{ArgMin}}\ f_t^{\text{ideal}}(\mathbf{x}_t) \tag{7.2}$$

Ideally, $\mathrm{C}_{\text{ideal}} = f^{\text{ideal}}_{\tau_{\text{ideal}}}(\mathbf{x}_{\tau_{\text{ideal}}})$ is the illusory cost that the system would endure if it made a decision as soon as the prediction is correct $h_t(\mathbf{x}_t) = y$.

This being not reachable in practice, we proposed a possible estimate of the ideal early decision function that we called optimal early decision (**ECONOMY**: **E**arly **C**lassification for **O**ptimized adaptive and **NO**n-**MY**opic online decision making).

7.2.3 ECONOMY: Optimal early decision

To obtain an optimal estimate of the ideal time decision τ_{ideal} that, in addition, takes into account the peculiarities of the incoming time series $\mathbf{x}_t$, we proposed:

$$f(\mathbf{x}_t) = \sum_{y \in \mathcal{Y}} P(y|\mathbf{x}_t) \sum_{\hat{y} \in \mathcal{Y}} P_t(\hat{y}|y, \mathbf{x}_t) \times \mathrm{C}(\hat{y}|y) + \mathrm{C}(t) \tag{7.3}$$

If the cost is computed for all time steps $t \in \{1 \dots, T\}$, the optimal time $t^\star$ for the decision problem is defined as:

$$t^* = \underset{t \in \{1,\dots,T\}}{\mathrm{ArgMin}}\, f(\mathbf{x}_t)$$

This formulation is optimal a posteriori when all T data points of the input time series are available and costs for all time steps are computed. However, it does not readily yield a method for finding, online, the optimal decision time.

In order to make a decision online, we proposed an online optimal early decision described in the following.

7.2.4 Online optimal early decision

To overcome these problems, i.e. estimate online the optimal time while taking into account the incoming time series, we proposed two meta-algorithms that are based on the same idea: given an incoming time series $\mathbf{x}_t$, and although $T - t$ data points are still missing, we computed the expected costs $f_\tau(\mathbf{x}_t)$ of classifying $\mathbf{x}_t$ for each future time step $\tau \in \{0, \dots, T - t\}$. This yields the expected best future time for making a decision as:

$$t^* = t + \underset{\tau \in \{0,\dots,T-t\}}{\mathrm{ArgMin}}\, f_\tau(\mathbf{x}_t)$$

where $f_\tau(\mathbf{x}_t)$ is a function assumed to be able to estimate the cost associated to the incoming time series $\mathbf{x}_t$, which $t < T$, for each future time step $\tau \in \{0, \dots, T - t\}$. In

the following, we present how we formulated this function.

7.2.5 The need of segmentation

To implement the above mentioned idea, i.e. estimate the costs for each future time step $\tau \in \{0, \ldots, T-t\}$, an alluring solution is to leverage the complete information contained in the training set to make adaptive estimate of the future costs given $\mathbf{x}_t$. To achieve this, we suggested to capture these complete information by using a segmentation technique that is able to capture typical evolutions of the training time series.

We proposed two meta-algorithms ECONOMY-K and ECONOMY-γ that extend the generic optimization criterion ECONOMY and use two different segmentations:

1. the segmentation in ECONOMY-K is performed on time series in their time-domain form. It is concerned only with values of time series, and uses a clustering technique to segment the complete training time series once and for all.

2. the segmentation in ECONOMY-γ is realized over the set of complete training series, like in the clustering approach. But, by contrast with the latter, the segmentation here should be more informed because it also uses the class labels of the time series thanks to the confidence levels computed by the used probabilistic classifier (see explanations in Section 7.2.7).

7.2.6 ECONOMY-K: Clustering-based approach

The segmentation of the complete training time series is performed in this approach by using a clustering technique. A number of K clusters is decided and a distance function is used to form these clusters. The optimal early decision equation thus becomes:

$$f_\tau(\mathbf{x}_t) = \sum_{\mathfrak{c}_k \in \mathcal{C}} P(\mathfrak{c}_k|\mathbf{x}_t) \sum_{y \in \mathcal{Y}} P(y|\mathfrak{c}_k) \sum_{\hat{y} \in \mathcal{Y}} P_{t+\tau}(\hat{y}|y, \mathfrak{c}_k)\, \mathrm{C}(\hat{y}|y) + \mathrm{C}(t+\tau) \tag{7.4}$$

where $P(\mathfrak{c}_k|\mathbf{x}_t)$ is a specific similarity measure computed between an incoming time series $\mathbf{x}_t$ and each cluster $\mathfrak{c}_k$. The conditional probabilities $P_{t+\tau}(\hat{y}|y, \mathfrak{c}_k)$ computed at each time step and for each cluster $\mathfrak{c}_k$ are the terms of the confusion matrices estimated at each time step and associated with the classifier $h_t(\mathbf{x}_t) = \hat{y}$ that predicts the class label of $\mathbf{x}_t$.

7.2.7 ECONOMY-γ: Confidence-based approach

The segmentation in this approach is performed by computing the quantiles of the outputs of a probabilistic classifier g_t over the training set, at each time step $t \in \{1, \ldots, T\}$. Each quantile, that we named confidence interval, is considered as a state in a special Markov chain in which the states at time t are not connected between each other but fully connected with the states in the next time step. This makes it possible to estimate the transition matrices over time and code an incoming time series by a sequence of states, thanks to its probabilistic predictions $g_t(\mathbf{x}_t)$, obtained at each time step, for the available data points. In this second approach, the optimal early decision equation becomes:

$$f_\tau(\mathbf{x}_t) \;=\; \sum_{y,\hat{y}\in\mathcal{Y}} \sum_{\ell=1}^{N} P(\gamma_{t+\tau} = \ell | \langle \gamma_t \rangle)\; P_{t+\tau}(\hat{y}|y, \gamma_{t+\tau} = \ell) \;\times \mathrm{C}(\hat{y}|y) + \mathrm{C}(t+\tau)$$

Note that this is a very simplified version of the decision rule we proposed in Section 5.5.

In order to assess the robustness of the proposed approaches and their ability to make optimal early predictions, we conducted a number of empirical studies in which we: (i) assessed the performances of the proposed methods regarding their abilities to make early predictions as it is expected from early classification systems, and (ii) compared the proposed methods the one against the other.

7.2.8 Empirical assessments of approaches with respect to the expected behaviors of early classification systems

To achieve this task, we conducted an empirical study in a controlled context and defined the behaviors we want to examine in our approaches. The question we asked was the following: how does the estimated decision time varies with:

- The rate of information gain (i.e. the hardness of the supervised task)

$\longrightarrow$ The higher the rate, the earlier the decision (except if very low rate)

- The similarity between time series in different classes

$\longrightarrow$ The higher the similarity, the later the decision

- The cost of delaying decision

$\longrightarrow$ The higher the cost, the earlier the decision

Based on experimental results we obtained using synthetic and real data sets, we showed that both approaches not only successfully meet what was expected from early classification systems, but also provide an interesting property: in some situations, when estimating that additional data points will not improve the prediction, the proposed methods decide that it is not worth waiting and make a prediction early on, often at the earliest possible moment.

7.2.9 Empirical assessments: comparing one approach against the other

From the empirical results (see Chapter 6), we observed that the Markov chain-based approach ECONOMY-γ outperforms, in majority of cases, the clustering-based approach ECONOMY-K over synthetic and real data sets. We interpreted this finding by the following: in ECONOMY-γ, the fact that the information about the class labels of time series are taken into account when segmenting leads to groups of time series that significantly differs one from the other. Consequently, this ensures confusion matrices that are significantly different one from the other. This is less clear regarding the confusion matrices obtained by a clustering technique in the ECONOMY-K approach.

To recap, according to the empirical assessments, it is clear that ECONOMY-γ outperforms ECONOMY-K, but at the same time, both approaches succeeded to meet early classification requirements.

Furthermore, what was not questioned in our work is, whether our estimates are optimal, how can we theoretically assess this, can we do better, i.e. can there be any improvements? In this thesis, we had not formally answered these questions, but, in the following paragraphs, the intention is to give some directions for future works.

7.3 Future works

Early classification of time series is emerging as an active area of research in many real application areas. Many challenges still lie ahead. Regarding our work, we have contributed to explicitly formalize the problem and propose solutions to solve it. In the following, we propose to highlight and identify some interesting leads.

7.3.1 Some leads for possible improvements

We are aware that in this thesis, no impact assessment has been carried out to take stock of how different choices (detailed below) could have been taken in our decision policies. That is why we give here some leads for improvements that it would be very interesting

to explore in future works.

- **Choice of the time series representation.** A first improvement can come from a good choice of data representation. Many recent researches (e.g. [44], [60], [67], and many others) point out that there is a growing consensus that a good choice of data representation, along with an appropriate distance function or similarity measure can considerably enhance the predictive performance of classifiers (or predictive models in general). Here, it would be useful to assess the contribution of using a representation in the enhancement of final results in our framework. In our work, while the proposed ECONOMY-K is performed on time series in their original form, ECONOMY-γ is carried out over sequences that code time series using a specific transformation technique based on Markov chains. From the experimental results, we observed that ECONOMY-γ outperforms ECONOMY-K. Of course, here, we are not sure about the contribution of changing the representation of time series in the success of ECONOMY-γ over ECONOMY-K since both approaches use different segmentation techniques. However, it would be interesting to compare, under the same conditions, ECONOMY-γ against ECONOMY-K and then make meaningful conclusions about the role that would play time series representations.

- **Choice of the early classification strategy.** The second improvement can be obtained by a good choice of the early classification strategy. In Chapter 3, we presented a non-exhaustive list of possible strategies that can be used to make an early classifier able to label incomplete time series. We distinguished, (i) strategies that implicitly or explicitly use the complete information contained in the training time series in order to impute the missing values, (ii) strategies based on changing the representation of time series to another time-invariant representation where it is possible to obtain complete data vector from incomplete time series, and (iii) strategies that directly deal with missing values, for example, by training a set of classifiers, each at each time step. In our experimental settings, we used the latter strategy where a series of classifiers $\mathcal{H} = \{h_t\}_{1 \leq t \leq T}$ are independently learnt over training time series trimmed to their t first components. In future work, under the same conditions, one can compare the different strategies, for instantiating early classifiers, regarding their contribution in improving the early predictions?

- **Choice of the type of classifier.** The third improvement can be achieved by making a good choice of the classifier type. On the one hand, when a large amount

of data is available, the choice of a classifier over another, generally has little impact on final results, making it difficult to make a clear-cut choice. In this case, a best choice may be based on the scalability of training and the complexity of execution. On the other hand, for small sizes of the training data sets, the choice of a classifier may depend on its performance in specific classes of problems.

- **Choice of the segmentation.** The fourth improvement can be achieved by the choice of a good segmentation method. The objective of the segmentation is to build meaningful groups of training time series in order to yield confusion matrices that are significantly different for a better estimation of the optimal decision time. In Chapter 5, we proposed two different segmentation techniques. The first uses a clustering algorithm and the second is based on special Markov chains constructed over the training set of complete time series. In the empirical assessment, we showed that the use of different segmentations impacts the early prediction results (see Figure 5.3). The question that can arise here, is: besides the ones already explored, are there other, better ways to segment training data, with the aim to make better estimates of the optimal decision time?

7.3.2 Is there an equivalent to empirical risk and real risk?

As in statistical learning theory, one wants to have a bound on the expected error of a considered algorithm. In our context, it would be also interesting to compare the performance of our early decision rule to the best possible one. However, one should first define what the best early decision rule is. In Section 6.1, we assumed that an optimal decision rule can be defined by $\mathrm{C}(h_t(\mathbf{x}_t) = y|y)+\mathrm{C}(t)$. However, this rule is unrealistically optimal since it makes a decision as soon as the prediction is correct, $h_t(\mathbf{x}_t) = y$, which can happen accidentally even though the prediction function h_t is bad. Otherwise, how one can define the optimal decision rule? This is an interesting but non-trivial problem that needs to be particularly addressed.

7.3.3 Why not learning the optimal time?

Another question that can arise when making online decisions is why not directly estimate the optimal decision time given an incoming time series? A possible straightforward approach would be to learn a regression model on the training set $\mathcal{S}$, where the target variable is no longer the class label describing a time series but its corresponding optimal time.

Here, again, before solving the supervised learning task, one should first define what the optimal time is. It turns out that this question is strongly related to the hard question of how to define an optimal early decision rule to determine optimal costs? Therefore, the problem is twofold and it will be particularly interesting to investigate it. In addition, the way this optimal time is defined here, one can fear that results exhibit a high variance level, which would certainly lower the quality of the regression.

7.3.4 Possible extensions of the proposed approaches

There are a number of extensions that could be made to the ECONOMY framework. Particularly, extending it in order to deal with multi-class and multi-label classification problems.

The cost-sensitive early classification approaches proposed in this work successfully addressed binary and *single*-label classification problems. However, extensions to multi-class and multi-label classification problems are not straightforward for both approaches. While approach ECONOMY-K can be naturally extended to the multi-class case, the ECONOMY-γ needs some special reformulations to be extended.

The extension to the multi-label problems is difficult for both approaches and is generally difficult for the multi-objective optimization problem we considered in this work.

7.3.4.1 Extending ECONOMY-K to multi-class problems

To be extended to the multi-class problem, ECONOMY-K has only to use a classifier that supports multi-class problems. In our experiments, we used a Multilayer Perceptron (MLP) that is able as well to provide natural extension to the multi-class problem.

Thus, extending ECONOMY-K is straightforward provided the used classifier should be able to give predictions in multi-class case.

7.3.4.2 Extending ECONOMY-γ to multi-class problems

The extension of ECONOMY-γ to multi-class problems is not a trivial task. Indeed, the segmentation technique used in this approach is mainly based on discretizing a confidence interval obtained, at each time step, from the outputs of a probabilistic classifier g_t (over a training set), that somewhat reflects the probability that the class $+1$ has been predicted for a time series $\mathbf{x}_t$. In the binary case, it is possible to have good estimates of such score or probability and it is easy to consider a class against the other. However, it becomes generally more difficult to obtain efficient estimates of these probabilities in the

multi-class case. In addition, it would be also difficult to apply one-vs-one or one-vs-all techniques mainly when trying to combine results.

7.3.4.3 Extension to multi-label problems

In multi-label classification, each time series is associated to a set of labels. For example, in medical diagnosis, a patient may be suffering, at the same time, of two distinct diseases such as diabetes and asthma. In the literature (see [97]), the problem is handled using either (i) transformation methods that transform the problem into one or more single-label problems, or (i) adaptation methods that arrange to deal directly with multi-label problems. These extensions are increasingly studied, yielding so far many solutions and learning algorithms. Now, in our case, how to address the extension of our framework to multi-label problems while a trade-off between two contradictory objectives: the earliness and the quality of predictions should be optimized online? Intuitively, using transformation methods will give bad results, since, the average over all the optimal results obtained from single-label problems may be less optimal. We think that this needs to solve a nice problem of multi-label multi-objective classification problem.

7.4 Conclusions

In this thesis, we focused on the early classification of time series task that is useful in many real application domains. From a thorough study of the state of the art, we have addressed two basic challenges for making early predictions: (i) the capacity of making a prediction on the class label of incomplete time series, and (ii) the estimation of the optimal time for making a prediction.

In Chapter 2, we explicitly defined the early classification problem and proposed a new framework that endows early classification systems with a decision function, which we named $\mathcal{T}rigger$ function, that decides when to make predictions.

In Chapter 3, we examined the problem of classifying incomplete time series. We proposed a categorization of the different state-of-the-art methods for implementing early classifiers.

In Chapter 4, we focused on the problem of online decision making and critically examined the state-of-the-art methods according to the $\mathcal{T}rigger$ decision function.

In Chapter 5, we formalized the early classification problem as a cost-sensitive online decision making problem and proposed a generic optimization criterion that we called **ECONOMY**, **E**arly **C**lassification for **O**ptimized and **NO**n-**MY**opic online decision making, that involves two types of costs: misclassification costs and delaying of decision

costs. Through ECONOMY, we proposed two efficient algorithms that have many interesting properties including: the estimation of the future optimal time for making a prediction, adaptive and non-myopic decisions.

In Chapter 6, extensive experiments on synthetic and real data sets vindicated the robustness of the proposed algorithms.

The methods presented here should provide a useful tool for many early classification tasks. Besides this, the proposed methods can contribute to other tasks such as anytime classification, etc.

Appendix A

Appendix: Exhaustive results

C(t)	$\eta(t)$	ECONOMY-K				ECONOMY-γ					
		$\overline{\tau}^{\star}_{\text{ETM}}$	$\overline{\text{C}}_{\text{RCM}}$	$\overline{\tau}^{\star}_{\text{PETM}}$	$\overline{\text{C}}_{\text{PRCM}}$	$\overline{\tau}^{\star}_{\text{ETM}}$	$\overline{\text{C}}_{\text{RCM}}$	$\overline{\tau}^{\star}_{\text{PETM}}$	$\overline{\text{C}}_{\text{PRCM}}$	$\overline{\tau}^{\star}_{\text{ITM}}$	$\overline{\text{C}}_{\text{ICM}}$
	0.1	16.8±1.2	-0.67	14.8±1.6	-0.59	26.1±12.7	-0.73	22.2±6.9	-0.66	6.5±3.4	-0.96
	0.2	24.9±1.5	-0.61	24.5±2.7	-0.61	33.3±12.5	-0.7	30.3±7.5	-0.64	7.3±5.2	-0.99
	0.5	17.5±3.2	-0.2	17.2±2.6	-0.21	39.9±7.3	-0.59	41.6±7.8	-0.53	11.1±9.5	-0.95
	1.0	13.5±5.4	-0.08	21.0±3.9	-0.13	38.5±8.8	-0.37	42.8±7.0	-0.36	11.5±11.5	-0.92
0.001	1.5	18.4±5.5	-0.09	12.4±4.3	-0.04	35.7±9.3	-0.24	43.1±6.5	-0.25	12.4±10.6	-0.96
	5.0	10.5±6.2	0.0	10.2±5.0	0.01	6.2±2.6	0.0	9.3±4.3	0.01	14.5±13.8	-0.9
	10.0	17.1±9.9	0.01	14.0±10.0	0.01	13.5±1.9	0.01	13.4±2.0	0.0	15.6±15.1	-0.86
	15.0	23.0±10.6	0.04	15.4±10.8	0.03	8.4±1.2	0.0	9.8±3.2	0.0	13.5±12.6	-0.88
	20.0	22.1±11.3	0.04	16.7±11.6	0.03	14.3±4.9	0.02	13.9±4.4	0.01	13.8±12.9	-0.9
	0.1	12.9±3.4	-0.4	14.0±0.7	-0.44	17.2±5.3	-0.57	16.7±3.4	-0.47	6.5±3.4	-0.9
	0.2	14.8±1.6	-0.28	15.8±2.5	-0.28	22.8±6.4	-0.46	23.6±5.5	-0.36	7.3±5.2	-0.92
	0.5	10.2±3.0	0.01	16.9±2.5	-0.06	22.7±11.3	-0.18	32.9±10.4	-0.13	11.1±9.5	-0.85
	1.0	10.2±1.4	0.05	14.3±1.9	0.05	14.6±6.4	0.02	33.0±12.6	0.02	11.5±11.5	-0.82
0.01	1.5	4.1±0.6	0.04	8.9±3.3	0.07	15.4±12.1	0.06	27.7±11.7	0.06	12.4±10.6	-0.85
	5.0	4.7±1.0	0.04	7.1±2.0	0.07	5.0±0.0	0.05	8.0±3.7	0.08	14.5±13.8	-0.77
	10.0	4.0±0.0	0.04	4.3±0.6	0.04	10.7±3.9	0.1	11.0±3.5	0.11	15.6±15.1	-0.72
	15.0	4.7±1.4	0.04	5.1±1.7	0.05	8.4±1.2	0.08	9.1±2.6	0.09	13.5±12.6	-0.76
	20.0	4.0±0.0	0.04	4.8±1.4	0.05	6.3±1.5	0.06	8.3±4.5	0.08	13.8±12.9	-0.78
	0.1	4.0±0.0	0.13	4.3±1.4	0.14	9.9±3.4	0.03	10.9±3.4	0.08	6.5±3.4	-0.64
	0.2	4.0±0.0	0.2	5.6±3.9	0.26	5.3±2.6	0.2	10.4±5.1	0.23	7.3±5.2	-0.63
	0.5	4.0±0.0	0.2	4.4±1.7	0.22	4.0±0.0	0.2	6.9±4.0	0.32	10.7±8.8	-0.41
	1.0	4.0±0.0	0.2	4.1±0.4	0.2	4.0±0.0	0.2	4.0±0.0	0.2	10.8±10.6	-0.36
0.05	1.5	4.0±0.0	0.2	4.8±1.3	0.24	4.0±0.0	0.2	4.0±0.0	0.2	11.7±9.7	-0.35
	5.0	4.0±0.0	0.2	4.3±0.5	0.21	5.0±0.0	0.25	5.0±0.0	0.25	13.0±12.4	-0.2
	10.0	4.0±0.0	0.2	4.0±0.0	0.2	4.0±0.0	0.2	5.2±2.4	0.26	12.5±12.5	-0.11
	15.0	4.0±0.0	0.2	4.1±0.3	0.2	4.3±0.7	0.21	4.4±0.8	0.22	12.0±11.0	-0.22
	20.0	4.0±0.0	0.2	4.0±0.2	0.2	4.6±1.2	0.23	4.7±1.3	0.24	12.2±11.2	-0.23
	0.1	4.0±0.0	0.21	4.3±1.4	0.23	5.1±2.4	0.22	8.2±3.4	0.21	6.5±3.4	-0.51
	0.2	4.0±0.0	0.28	4.3±1.4	0.3	4.0±0.0	0.28	6.9±4.5	0.34	7.3±5.1	-0.48
	0.5	4.0±0.0	0.28	4.1±0.8	0.29	4.0±0.0	0.28	4.0±0.0	0.28	9.8±7.8	-0.21
	1.0	4.0±0.0	0.28	4.0±0.1	0.28	4.0±0.0	0.28	4.0±0.0	0.28	8.3±7.2	-0.17
0.07	1.5	4.0±0.0	0.28	4.0±0.4	0.28	4.0±0.0	0.28	4.0±0.0	0.28	10.1±7.9	-0.14
	5.0	4.0±0.0	0.28	4.1±0.3	0.29	5.0±0.0	0.35	5.0±0.0	0.35	9.4±9.1	0.02
	10.0	4.0±0.0	0.28	4.0±0.0	0.28	4.0±0.0	0.28	4.0±0.0	0.28	8.6±8.6	0.1
	15.0	4.0±0.0	0.28	4.0±0.2	0.28	4.0±0.0	0.28	4.4±0.8	0.3	10.0±8.8	-0.01
	20.0	4.0±0.0	0.28	4.0±0.1	0.28	4.6±1.1	0.32	4.6±1.2	0.32	9.8±8.7	-0.02
	0.1	4.0±0.0	0.33	4.0±0.0	0.33	4.0±0.0	0.33	5.7±2.9	0.35	6.5±3.4	-0.32
	0.2	4.0±0.0	0.4	4.0±0.0	0.4	4.0±0.0	0.4	4.0±0.0	0.4	6.8±4.3	-0.27
	0.5	4.0±0.0	0.4	4.0±0.0	0.4	4.0±0.0	0.4	4.0±0.0	0.4	8.0±5.8	0.06
	1.0	4.0±0.0	0.4	4.0±0.0	0.4	4.0±0.0	0.4	4.0±0.0	0.4	6.6±4.7	0.05
0.1	1.5	4.0±0.0	0.4	4.0±0.1	0.4	4.0±0.0	0.4	4.0±0.0	0.4	8.7±6.5	0.14
	5.0	4.0±0.0	0.4	4.0±0.2	0.4	5.0±0.0	0.5	5.0±0.1	0.49	6.1±5.2	0.25
	10.0	4.0±0.0	0.4	4.0±0.0	0.4	4.0±0.0	0.4	4.0±0.0	0.4	5.9±4.8	0.31
	15.0	4.0±0.0	0.4	4.0±0.1	0.4	4.0±0.0	0.4	4.4±0.8	0.44	7.3±6.4	0.26
	20.0	4.0±0.0	0.4	4.0±0.0	0.4	4.0±0.0	0.4	4.6±1.2	0.46	7.0±5.7	0.23

Table A.1: Comparison table: results of clustering approach vs. confidence approach 2 (with trend = 0.005, over simulated sine data)

A. APPENDIX: EXHAUSTIVE RESULTS

C(t)	$\eta(t)$	ECONOMY-K				ECONOMY-γ					
		$\overline{\tau}^{\star}_{\text{ETM}}$	$\overline{\text{C}}_{\text{RCM}}$	$\overline{\tau}^{\star}_{\text{PETM}}$	$\overline{\text{C}}_{\text{PRCM}}$	$\overline{\tau}^{\star}_{\text{ETM}}$	$\overline{\text{C}}_{\text{RCM}}$	$\overline{\tau}^{\star}_{\text{PETM}}$	$\overline{\text{C}}_{\text{PRCM}}$	$\overline{\tau}^{\star}_{\text{ITM}}$	$\overline{\text{C}}_{\text{ICM}}$
	0.1	13.5±2.0	-0.75	15.5±1.0	-0.76	22.1±12.5	-0.75	18.7±7.4	-0.74	5.5±2.2	-0.92
	0.2	14.6±2.0	-0.52	15.5±0.9	-0.57	31.8±14.8	-0.72	22.9±6.5	-0.68	6.5±3.5	-0.95
	0.5	20.9±2.2	-0.48	20.7±1.9	-0.47	35.3±10.5	-0.7	34.6±7.8	-0.64	8.9±6.3	-0.96
	1.0	9.5±1.2	-0.05	18.7±4.7	-0.21	36.0±9.8	-0.61	42.0±7.8	-0.56	10.4±9.3	-0.95
0.001	1.5	15.7±7.6	-0.11	14.2±5.6	-0.07	39.6±6.9	-0.48	44.5±5.2	-0.48	11.6±9.0	-0.98
	5.0	7.2±2.5	0.0	9.7±3.9	0.01	12.3±13.5	-0.01	18.5±16.9	-0.03	13.9±12.8	-0.92
	10.0	12.1±6.9	0.01	10.0±6.4	0.02	12.1±4.2	0.02	12.6±4.4	0.02	15.2±14.4	-0.88
	15.0	20.6±8.7	0.01	15.8±9.5	0.01	10.2±3.4	0.0	11.7±3.6	0.01	13.6±12.6	-0.89
	20.0	22.3±9.2	0.03	15.0±9.9	0.0	14.0±6.6	0.01	12.6±5.0	0.01	13.7±12.7	-0.91
	0.1	11.7±1.6	-0.62	15.4±1.1	-0.62	12.1±1.6	-0.64	12.1±2.8	-0.62	5.5±2.2	-0.87
	0.2	14.6±2.0	-0.39	15.5±0.9	-0.43	19.4±4.1	-0.55	17.5±4.0	-0.49	6.5±3.5	-0.89
	0.5	14.8±1.0	-0.22	15.2±0.8	-0.2	23.9±9.2	-0.41	26.1±6.6	-0.36	8.9±6.3	-0.88
	1.0	8.6±1.1	0.06	16.0±2.0	-0.02	24.0±9.9	-0.21	33.7±8.3	-0.17	10.4±9.3	-0.86
0.01	1.5	4.1±0.5	0.04	9.6±4.3	0.08	23.9±10.0	-0.07	38.1±9.2	-0.09	11.6±9.0	-0.87
	5.0	4.8±1.1	0.04	6.9±2.1	0.07	6.5±1.8	0.06	6.7±2.4	0.06	13.9±12.8	-0.79
	10.0	4.0±0.0	0.04	4.7±1.3	0.05	10.4±2.1	0.1	9.5±3.0	0.09	15.2±14.4	-0.74
	15.0	4.9±2.2	0.05	5.2±2.4	0.04	8.9±1.8	0.08	10.1±3.8	0.1	13.6±12.6	-0.76
	20.0	4.0±0.0	0.04	4.8±1.4	0.05	11.4±4.6	0.11	10.7±4.8	0.1	13.7±12.7	-0.78
	0.1	4.0±0.0	0.04	5.1±2.5	0.06	9.8±1.1	-0.18	9.6±2.0	-0.15	5.5±2.2	-0.65
	0.2	4.0±0.2	0.16	15.0±2.4	0.2	10.9±1.6	0.05	10.7±3.5	0.04	6.5±3.5	-0.63
	0.5	4.0±0.0	0.2	13.0±4.5	0.4	4.2±0.4	0.21	9.4±5.6	0.25	8.9±6.2	-0.52
	1.0	8.6±1.1	0.41	8.1±4.7	0.3	4.0±0.0	0.2	4.0±0.0	0.2	10.2±8.9	-0.44
0.05	1.5	4.0±0.0	0.2	4.6±1.3	0.23	4.0±0.0	0.2	4.0±0.0	0.2	11.4±8.7	-0.41
	5.0	4.0±0.0	0.2	4.2±0.4	0.21	5.0±0.0	0.25	4.9±0.7	0.24	12.8±11.8	-0.24
	10.0	4.0±0.0	0.2	4.0±0.1	0.2	4.0±0.0	0.2	7.2±2.7	0.36	13.3±12.9	-0.14
	15.0	4.0±0.0	0.2	4.1±0.3	0.21	4.3±0.7	0.21	4.4±0.8	0.22	12.5±11.4	-0.22
	20.0	4.0±0.0	0.2	4.0±0.2	0.2	4.8±1.0	0.24	4.8±1.1	0.24	12.2±11.1	-0.24
	0.1	4.0±0.0	0.12	4.3±1.5	0.14	8.4±2.5	0.02	9.1±2.1	0.03	5.5±2.2	-0.53
	0.2	4.0±0.2	0.24	4.3±1.6	0.27	5.4±2.9	0.23	7.8±3.6	0.23	6.5±3.5	-0.5
	0.5	4.0±0.0	0.28	4.3±1.5	0.3	4.2±0.4	0.29	5.9±3.5	0.34	8.8±6.0	-0.35
	1.0	8.6±1.1	0.58	4.5±1.3	0.31	4.0±0.0	0.28	4.0±0.0	0.28	9.0±7.2	-0.25
0.07	1.5	4.0±0.0	0.28	4.1±0.5	0.29	4.0±0.0	0.28	4.0±0.0	0.28	10.6±7.9	-0.19
	5.0	4.0±0.0	0.28	4.1±0.3	0.29	5.0±0.0	0.35	4.8±0.4	0.33	10.0±9.2	-0.01
	10.0	4.0±0.0	0.28	4.0±0.0	0.28	4.0±0.0	0.28	4.0±0.0	0.28	9.0±8.8	0.08
	15.0	4.0±0.0	0.28	4.0±0.2	0.28	4.0±0.0	0.28	4.4±0.8	0.3	10.0±8.9	0.0
	20.0	4.0±0.0	0.28	4.0±0.1	0.28	4.8±1.0	0.33	4.5±1.0	0.32	9.9±8.8	-0.02
	0.1	4.0±0.0	0.24	4.2±1.3	0.26	6.3±2.9	0.24	7.4±2.8	0.25	5.5±2.2	-0.37
	0.2	4.0±0.2	0.37	4.0±0.2	0.37	4.0±0.0	0.36	4.8±2.0	0.4	6.5±3.5	-0.3
	0.5	4.0±0.0	0.4	4.0±0.0	0.4	4.2±0.4	0.42	4.2±0.4	0.42	8.0±5.0	-0.09
	1.0	4.0±0.0	0.4	4.1±0.4	0.41	4.0±0.0	0.4	4.0±0.0	0.4	7.2±4.9	-0.01
0.1	1.5	4.0±0.0	0.4	4.0±0.2	0.4	4.0±0.0	0.4	4.0±0.0	0.4	9.1±6.6	0.11
	5.0	4.0±0.0	0.4	4.0±0.1	0.4	5.0±0.0	0.5	4.8±0.4	0.47	7.1±6.3	0.24
	10.0	4.0±0.0	0.4	4.0±0.0	0.4	4.0±0.0	0.4	4.0±0.0	0.4	6.0±4.9	0.29
	15.0	4.0±0.0	0.4	4.0±0.1	0.4	4.0±0.0	0.4	4.4±0.8	0.44	7.4±6.5	0.26
	20.0	4.0±0.0	0.4	4.0±0.1	0.4	4.4±0.8	0.44	4.5±1.0	0.45	7.0±5.7	0.22

Table A.2: Comparison table: results of clustering approach vs. confidence approach 2 (with trend = 0.01, over simulated sine data)

A. APPENDIX: EXHAUSTIVE RESULTS

C(t)	$\eta(t)$	ECONOMY-K				ECONOMY-γ					
		$\overline{\tau}^{\star}_{\text{ETM}}$	$\overline{\text{C}}_{\text{RCM}}$	$\overline{\tau}^{\star}_{\text{PETM}}$	$\overline{\text{C}}_{\text{PRCM}}$	$\overline{\tau}^{\star}_{\text{ETM}}$	$\overline{\text{C}}_{\text{RCM}}$	$\overline{\tau}^{\star}_{\text{PETM}}$	$\overline{\text{C}}_{\text{PRCM}}$	$\overline{\tau}^{\star}_{\text{ITM}}$	$\overline{\text{C}}_{\text{ICM}}$
0.001	0.1	6.9±0.6	-0.73	13.6±0.9	-0.76	16.0±16.0	-0.75	10.3±5.7	-0.75	4.1±0.4	-0.78
	0.2	5.0±0.0	-0.53	13.6±1.0	-0.74	19.2±14.0	-0.74	11.0±4.8	-0.74	4.4±1.1	-0.83
	0.5	11.8±3.6	-0.63	13.9±2.4	-0.71	19.6±10.7	-0.74	18.3±7.5	-0.73	5.5±2.5	-0.9
	1.0	16.1±0.4	-0.7	15.0±2.8	-0.64	22.1±13.2	-0.73	23.7±8.2	-0.71	5.5±3.1	-0.94
	1.5	20.3±4.3	-0.62	16.0±7.2	-0.44	26.6±13.4	-0.68	31.8±9.1	-0.7	7.1±4.6	-0.97
	5.0	8.3±4.3	-0.02	8.2±4.4	0.0	36.9±6.3	-0.6	43.9±4.4	-0.61	10.8±9.0	-0.96
	10.0	7.8±6.7	-0.02	9.4±5.4	0.01	41.3±5.9	-0.36	45.6±4.3	-0.37	12.3±11.1	-0.94
	15.0	7.8±4.9	0.0	9.6±4.7	0.0	9.7±12.4	-0.01	15.0±17.2	-0.04	13.0±11.8	-0.93
	20.0	14.4±12.0	-0.01	12.9±9.6	0.0	45.7±3.9	-0.16	45.0±8.6	-0.16	11.9±10.6	-0.96
0.01	0.1	4.0±0.0	-0.66	13.5±1.0	-0.64	6.7±2.0	-0.67	6.8±2.4	-0.69	4.1±0.4	-0.75
	0.2	5.0±0.0	-0.48	13.5±1.0	-0.62	9.9±0.9	-0.66	8.7±3.0	-0.66	4.4±1.1	-0.79
	0.5	8.4±4.1	-0.33	12.8±3.0	-0.55	13.3±5.1	-0.62	13.1±3.0	-0.61	5.5±2.5	-0.85
	1.0	7.7±3.5	-0.21	11.8±4.0	-0.4	17.4±8.9	-0.56	16.8±5.2	-0.54	5.5±3.1	-0.89
	1.5	10.2±4.2	-0.19	11.4±4.0	-0.2	20.4±10.3	-0.47	22.3±6.5	-0.48	7.1±4.6	-0.9
	5.0	5.7±1.4	0.05	6.9±3.4	0.07	30.1±7.2	-0.24	37.5±5.8	-0.24	10.8±9.0	-0.86
	10.0	4.0±0.0	0.04	7.4±3.9	0.08	11.8±2.6	0.09	31.9±14.9	0.04	12.3±11.1	-0.83
	15.0	4.5±1.1	0.04	5.7±1.7	0.05	5.6±1.1	0.05	5.6±1.0	0.05	13.0±11.8	-0.81
	20.0	4.0±0.0	0.04	6.2±3.3	0.06	10.1±2.4	0.09	9.2±2.3	0.08	11.9±10.6	-0.85
0.05	0.1	4.0±0.0	-0.5	4.1±0.4	-0.5	4.3±0.7	-0.51	5.8±1.7	-0.47	4.1±0.4	-0.58
	0.2	5.0±0.0	-0.28	5.0±0.3	-0.28	5.8±1.7	-0.37	6.9±2.2	-0.37	4.4±1.1	-0.61
	0.5	5.0±0.4	0.1	5.4±1.4	0.12	9.9±0.9	-0.21	10.1±2.1	-0.2	5.5±2.2	-0.63
	1.0	4.0±0.0	0.11	5.5±1.3	0.1	10.6±2.5	0.02	11.6±2.9	-0.02	5.5±3.0	-0.67
	1.5	4.0±0.1	0.2	6.1±2.1	0.3	6.1±2.5	0.22	9.7±4.0	0.13	7.0±4.1	-0.62
	5.0	4.0±0.0	0.2	5.2±0.9	0.26	6.3±1.2	0.32	5.8±1.5	0.29	10.6±8.7	-0.43
	10.0	4.0±0.0	0.2	4.8±1.0	0.24	4.0±0.0	0.2	4.0±0.0	0.2	11.7±10.3	-0.34
	15.0	4.0±0.0	0.2	4.4±0.7	0.22	5.3±1.0	0.27	5.4±1.0	0.27	12.2±10.9	-0.29
	20.0	4.0±0.0	0.2	4.2±0.5	0.21	4.0±0.0	0.2	4.4±0.9	0.22	11.2±9.6	-0.38
0.07	0.1	4.0±0.0	-0.42	4.1±0.4	-0.42	4.0±0.0	-0.42	4.8±1.3	-0.41	4.1±0.4	-0.5
	0.2	5.0±0.0	-0.18	5.0±0.2	-0.18	5.4±1.6	-0.26	6.0±2.0	-0.26	4.4±1.1	-0.52
	0.5	5.0±0.3	0.2	5.4±1.2	0.23	6.2±2.8	0.18	8.7±2.7	0.02	5.5±2.2	-0.52
	1.0	4.0±0.0	0.19	5.2±0.8	0.19	7.8±1.5	0.21	9.4±3.1	0.22	5.4±2.6	-0.56
	1.5	4.0±0.1	0.28	5.6±1.2	0.39	5.7±1.6	0.33	6.2±2.7	0.3	6.9±3.9	-0.48
	5.0	4.0±0.0	0.28	5.0±0.8	0.35	4.0±0.0	0.28	4.0±0.0	0.28	9.8±7.6	-0.23
	10.0	4.0±0.0	0.28	4.6±0.8	0.32	4.0±0.0	0.28	4.0±0.0	0.28	9.8±8.3	-0.13
	15.0	4.0±0.0	0.28	4.2±0.5	0.29	5.3±1.0	0.37	5.2±1.0	0.36	10.2±9.0	-0.07
	20.0	4.0±0.0	0.28	4.1±0.4	0.29	4.0±0.0	0.28	4.2±0.5	0.29	9.9±8.1	-0.17
0.1	0.1	4.0±0.0	-0.3	4.1±0.4	-0.3	4.0±0.0	-0.3	4.3±0.7	-0.29	4.1±0.4	-0.38
	0.2	5.0±0.0	-0.03	5.0±0.2	-0.03	4.9±1.2	-0.08	5.2±1.6	-0.1	4.4±1.1	-0.39
	0.5	4.0±0.0	0.33	5.1±0.4	0.36	4.0±0.0	0.33	4.6±1.8	0.37	5.4±2.1	-0.36
	1.0	4.0±0.0	0.31	4.9±0.7	0.33	4.0±0.0	0.31	6.8±2.8	0.45	5.4±2.5	-0.39
	1.5	4.0±0.0	0.4	5.2±0.7	0.52	4.8±0.4	0.48	4.9±0.9	0.46	6.8±3.7	-0.27
	5.0	4.0±0.0	0.4	4.9±0.7	0.49	4.0±0.0	0.4	4.0±0.0	0.4	8.2±6.1	0.04
	10.0	4.0±0.0	0.4	4.3±0.6	0.43	4.0±0.0	0.4	4.0±0.0	0.4	7.3±5.5	0.12
	15.0	4.0±0.0	0.4	4.1±0.3	0.41	5.1±1.1	0.51	4.7±0.8	0.47	7.1±5.8	0.18
	20.0	4.0±0.0	0.4	4.1±0.3	0.41	4.0±0.0	0.4	4.0±0.0	0.4	7.4±5.5	0.09

Table A.3: Comparison table: results of clustering approach vs. confidence approach 2 (with trend = 0.05, over simulated sine data)

A. APPENDIX: EXHAUSTIVE RESULTS

C(t)	$\eta(t)$	ECONOMY-K				ECONOMY-γ					
		$\overline{\tau}^{\star}_{\text{ETM}}$	$\overline{\text{C}}_{\text{RCM}}$	$\overline{\tau}^{\star}_{\text{PETM}}$	$\overline{\text{C}}_{\text{PRCM}}$	$\overline{\tau}^{\star}_{\text{ETM}}$	$\overline{\text{C}}_{\text{RCM}}$	$\overline{\tau}^{\star}_{\text{PETM}}$	$\overline{\text{C}}_{\text{PRCM}}$	$\overline{\tau}^{\star}_{\text{ITM}}$	$\overline{\text{C}}_{\text{ICM}}$
	0.1	5.9±0.4	-0.76	7.1±2.8	-0.76	8.4±8.6	-0.76	8.6±4.8	-0.76	4.0±0.2	-0.77
	0.2	5.1±0.4	-0.64	6.8±3.1	-0.65	17.4±14.6	-0.74	10.0±4.5	-0.74	4.2±0.9	-0.8
	0.5	9.0±4.2	-0.45	12.2±4.0	-0.66	16.3±10.5	-0.74	18.8±10.0	-0.74	5.1±2.4	-0.87
	1.0	14.4±2.3	-0.72	12.3±4.4	-0.6	20.8±10.0	-0.74	23.2±8.3	-0.74	5.3±3.0	-0.92
0.001	1.5	16.1±1.5	-0.65	9.6±4.8	-0.27	23.6±13.2	-0.68	30.3±10.4	-0.7	6.5±4.5	-0.96
	5.0	10.9±7.1	-0.1	9.3±5.4	-0.01	32.8±5.2	-0.65	42.3±6.1	-0.67	9.7±7.8	-0.96
	10.0	13.1±11.5	-0.08	11.1±7.5	-0.02	41.4±5.7	-0.52	45.2±4.4	-0.5	11.4±10.0	-0.95
	15.0	10.9±6.7	-0.03	11.4±6.4	-0.02	34.5±16.6	-0.25	39.2±14.9	-0.27	12.2±10.7	-0.95
	20.0	12.4±10.6	-0.01	12.6±9.2	0.0	44.5±3.9	-0.24	46.3±3.2	-0.24	11.3±9.9	-0.96
	0.1	4.0±0.0	-0.71	5.5±1.3	-0.71	5.3±1.4	-0.71	6.4±2.2	-0.71	4.0±0.2	-0.74
	0.2	5.1±0.4	-0.6	5.9±2.1	-0.59	6.7±1.9	-0.67	7.8±2.7	-0.67	4.2±0.9	-0.77
	0.5	5.1±0.6	-0.24	7.0±3.4	-0.27	11.1±4.5	-0.64	12.0±2.8	-0.64	5.1±2.4	-0.83
	1.0	6.7±2.5	-0.26	8.1±3.8	-0.33	15.0±4.0	-0.62	16.3±4.6	-0.59	5.3±3.0	-0.88
0.01	1.5	7.2±3.8	-0.09	8.8±4.3	-0.13	17.0±9.2	-0.51	19.1±5.5	-0.53	6.5±4.5	-0.9
	5.0	6.2±3.7	0.03	8.3±4.2	0.08	26.9±6.6	-0.34	33.3±5.5	-0.33	9.7±7.8	-0.87
	10.0	4.2±1.0	0.04	8.5±4.3	0.09	18.1±8.4	0.01	38.1±10.3	-0.08	11.4±10.0	-0.85
	15.0	4.9±1.8	0.04	6.4±2.4	0.05	5.4±0.5	0.05	8.3±9.7	0.06	12.2±10.7	-0.84
	20.0	4.1±1.1	0.04	6.9±3.9	0.07	9.7±1.6	0.08	10.2±2.0	0.08	11.3±9.9	-0.86
	0.1	4.0±0.0	-0.55	4.5±0.9	-0.53	4.6±0.8	-0.54	5.4±1.4	-0.5	4.0±0.2	-0.58
	0.2	5.0±0.2	-0.4	5.3±0.4	-0.38	4.8±0.7	-0.42	5.8±1.8	-0.44	4.2±0.9	-0.6
	0.5	5.0±0.2	-0.04	5.5±1.2	0.01	8.4±1.5	-0.3	9.3±1.8	-0.27	5.0±2.0	-0.62
	1.0	5.0±0.2	0.04	5.6±1.2	0.02	8.8±2.8	-0.16	10.2±3.3	-0.13	5.2±2.7	-0.66
0.05	1.5	5.3±1.0	0.23	5.7±1.3	0.26	7.2±4.1	0.08	11.0±3.3	0.0	6.3±3.6	-0.64
	5.0	4.0±0.0	0.2	5.8±2.1	0.29	4.0±0.0	0.2	4.0±0.0	0.2	9.6±7.5	-0.48
	10.0	4.0±0.0	0.2	5.2±1.6	0.26	4.0±0.0	0.2	4.0±0.0	0.2	11.0±9.4	-0.39
	15.0	4.0±0.0	0.2	4.5±0.8	0.22	5.1±0.7	0.25	5.2±0.6	0.26	11.6±9.9	-0.36
	20.0	4.0±0.0	0.2	4.3±0.7	0.22	4.0±0.0	0.2	4.0±0.0	0.2	10.8±9.2	-0.41
	0.1	4.0±0.0	-0.47	4.5±0.9	-0.44	4.6±0.8	-0.45	4.7±1.1	-0.44	4.0±0.2	-0.5
	0.2	5.0±0.2	-0.3	5.3±0.4	-0.28	4.6±0.8	-0.33	5.2±1.5	-0.35	4.2±0.9	-0.51
	0.5	5.0±0.2	0.06	5.5±1.2	0.12	7.5±1.3	-0.11	8.2±2.1	-0.11	5.0±2.0	-0.52
	1.0	4.0±0.0	0.13	5.4±0.8	0.13	7.1±2.1	0.03	8.3±2.6	0.07	5.2±2.2	-0.56
0.07	1.5	4.1±0.6	0.28	5.7±1.2	0.38	4.0±0.0	0.27	7.7±3.4	0.23	6.2±3.2	-0.52
	5.0	4.0±0.0	0.28	5.4±1.4	0.38	4.0±0.0	0.28	4.0±0.0	0.28	9.2±7.0	-0.29
	10.0	4.0±0.0	0.28	4.8±1.0	0.34	4.0±0.0	0.28	4.0±0.0	0.28	9.8±8.0	-0.18
	15.0	4.0±0.0	0.28	4.3±0.6	0.3	5.1±0.7	0.36	4.9±0.7	0.34	10.1±8.4	-0.14
	20.0	4.0±0.0	0.28	4.2±0.5	0.29	4.0±0.0	0.28	4.0±0.0	0.28	9.7±7.8	-0.2
	0.1	4.0±0.0	-0.35	4.5±0.9	-0.31	4.0±0.0	-0.35	4.1±0.6	-0.34	4.0±0.2	-0.37
	0.2	4.0±0.0	-0.18	4.5±0.9	-0.14	4.6±0.8	-0.2	4.6±1.1	-0.2	4.2±0.9	-0.38
	0.5	5.0±0.2	0.21	5.3±0.5	0.25	4.7±1.0	0.26	5.4±1.7	0.27	5.0±1.7	-0.37
	1.0	4.0±0.0	0.25	5.4±0.5	0.28	5.0±0.0	0.29	6.2±2.0	0.26	5.1±2.0	-0.41
0.1	1.5	4.0±0.1	0.39	5.5±0.8	0.53	4.0±0.0	0.39	4.3±1.1	0.39	6.1±3.0	-0.33
	5.0	4.0±0.0	0.4	5.1±0.8	0.51	4.0±0.0	0.4	4.0±0.0	0.4	8.1±5.7	-0.03
	10.0	4.0±0.0	0.4	4.6±0.8	0.46	4.0±0.0	0.4	4.0±0.0	0.4	7.6±5.6	0.07
	15.0	4.0±0.0	0.4	4.1±0.4	0.41	5.0±0.6	0.5	4.6±0.5	0.46	8.0±6.4	0.14
	20.0	4.0±0.0	0.4	4.1±0.4	0.41	4.0±0.0	0.4	4.0±0.0	0.4	7.6±5.6	0.05

Table A.4: Comparison table: results of clustering approach vs. confidence approach 2 (with trend = 0.07, over simulated sine data)

A. APPENDIX: EXHAUSTIVE RESULTS

C(t)	$\eta(t)$	ECONOMY-K				ECONOMY-γ					
		$\overline{\tau}^{\star}_{\text{ETM}}$	$\overline{\text{C}}_{\text{RCM}}$	$\overline{\tau}^{\star}_{\text{PETM}}$	$\overline{\text{C}}_{\text{PRCM}}$	$\overline{\tau}^{\star}_{\text{ETM}}$	$\overline{\text{C}}_{\text{RCM}}$	$\overline{\tau}^{\star}_{\text{PETM}}$	$\overline{\text{C}}_{\text{PRCM}}$	$\overline{\tau}^{\star}_{\text{ITM}}$	$\overline{\text{C}}_{\text{ICM}}$
0.001	0.1	4.0±0.0	-0.77	5.4±1.9	-0.77	5.6±5.1	-0.77	8.7±5.0	-0.76	4.0±0.0	-0.77
	0.2	5.0±0.0	-0.72	5.4±1.8	-0.72	9.6±10.9	-0.74	9.6±4.5	-0.74	4.1±0.6	-0.79
	0.5	6.7±1.9	-0.48	6.0±2.0	-0.4	16.7±12.1	-0.74	16.2±6.6	-0.74	4.6±1.6	-0.85
	1.0	12.3±2.9	-0.7	6.5±2.8	-0.38	17.6±11.7	-0.75	19.8±8.2	-0.76	5.1±3.0	-0.91
	1.5	9.1±4.6	-0.34	6.4±2.5	-0.12	22.5±13.1	-0.7	26.6±11.6	-0.72	6.0±4.6	-0.95
	5.0	18.9±12.1	-0.4	12.4±10.3	-0.17	28.2±7.6	-0.67	40.0±6.8	-0.69	8.6±6.5	-0.95
	10.0	16.2±11.9	-0.17	12.6±9.3	-0.08	40.4±6.0	-0.65	44.5±3.6	-0.64	10.0±8.4	-0.95
	15.0	14.8±7.9	-0.1	14.3±7.8	-0.08	26.3±17.9	-0.27	40.6±11.7	-0.41	11.1±9.5	-0.96
	20.0	15.8±11.1	-0.05	14.7±10.4	-0.03	43.3±5.0	-0.35	46.0±3.7	-0.37	10.5±9.0	-0.97
0.01	0.1	4.0±0.0	-0.73	5.2±1.2	-0.72	4.9±0.9	-0.72	6.0±2.1	-0.71	4.0±0.0	-0.73
	0.2	5.0±0.0	-0.68	5.4±1.8	-0.68	5.1±1.4	-0.67	6.8±2.6	-0.68	4.1±0.6	-0.75
	0.5	5.0±0.2	-0.39	5.4±1.8	-0.38	9.0±1.0	-0.66	11.3±2.9	-0.65	4.6±1.6	-0.81
	1.0	5.9±0.4	-0.34	5.8±1.2	-0.3	11.2±3.6	-0.65	13.7±5.0	-0.64	5.1±3.0	-0.87
	1.5	5.7±2.2	-0.06	5.5±2.0	-0.05	14.9±7.4	-0.58	16.4±4.9	-0.57	6.0±4.6	-0.89
	5.0	10.4±6.6	-0.09	8.9±5.0	0.01	22.1±6.8	-0.41	27.3±5.3	-0.42	8.6±6.5	-0.88
	10.0	9.6±7.7	0.02	9.2±5.0	0.08	28.4±9.4	-0.24	35.1±8.4	-0.24	10.0±8.4	-0.86
	15.0	7.1±4.4	0.04	8.8±4.2	0.06	7.8±8.1	0.04	16.2±16.8	0.02	11.1±9.5	-0.87
	20.0	6.1±4.1	0.06	7.8±4.3	0.08	10.0±0.0	0.07	35.0±15.5	0.01	10.5±9.0	-0.88
0.05	0.1	4.0±0.0	-0.57	5.0±0.2	-0.52	4.5±0.8	-0.54	5.2±1.3	-0.51	4.0±0.0	-0.57
	0.2	4.0±0.0	-0.5	5.0±0.2	-0.48	4.2±0.4	-0.51	5.5±1.4	-0.47	4.1±0.6	-0.59
	0.5	5.0±0.2	-0.19	5.2±1.2	-0.17	7.6±0.8	-0.36	8.8±1.4	-0.31	4.6±1.6	-0.62
	1.0	4.0±0.0	-0.03	5.3±1.2	-0.04	8.8±1.7	-0.25	9.4±2.1	-0.26	5.0±2.6	-0.66
	1.5	5.0±0.4	0.15	5.3±1.2	0.16	7.7±2.7	-0.03	9.8±2.8	-0.1	5.8±3.5	-0.66
	5.0	4.1±0.7	0.2	6.6±2.8	0.32	4.0±0.0	0.2	8.8±4.1	0.29	8.6±6.2	-0.53
	10.0	4.0±0.0	0.2	6.1±2.5	0.31	4.0±0.0	0.2	4.0±0.0	0.2	9.8±8.0	-0.46
	15.0	4.2±0.7	0.21	5.2±1.5	0.25	5.0±0.0	0.25	5.0±0.5	0.25	10.8±9.0	-0.42
	20.0	4.0±0.0	0.2	4.6±1.0	0.23	4.0±0.0	0.2	4.0±0.0	0.2	10.2±8.5	-0.46
0.07	0.1	4.0±0.0	-0.49	5.0±0.2	-0.42	4.2±0.4	-0.48	4.8±1.0	-0.44	4.0±0.0	-0.49
	0.2	4.0±0.0	-0.42	5.0±0.2	-0.38	4.2±0.4	-0.43	4.9±1.1	-0.39	4.1±0.6	-0.51
	0.5	5.0±0.2	-0.09	5.2±1.2	-0.07	6.8±1.7	-0.24	7.6±1.9	-0.17	4.6±1.5	-0.53
	1.0	4.0±0.0	0.05	5.2±0.7	0.06	7.9±1.8	-0.1	8.2±2.3	-0.09	4.9±1.9	-0.57
	1.5	4.1±0.3	0.23	5.3±1.2	0.27	6.8±2.6	0.15	8.3±2.6	0.13	5.6±2.6	-0.54
	5.0	4.0±0.1	0.28	6.0±2.2	0.42	4.0±0.0	0.28	4.0±0.0	0.28	8.3±5.8	-0.36
	10.0	4.0±0.0	0.28	5.4±1.7	0.38	4.0±0.0	0.28	4.0±0.0	0.28	9.3±7.3	-0.27
	15.0	4.0±0.1	0.28	4.7±1.2	0.33	5.0±0.0	0.35	4.9±0.3	0.34	9.8±7.8	-0.22
	20.0	4.0±0.0	0.28	4.4±0.8	0.31	4.0±0.0	0.28	4.0±0.0	0.28	9.4±7.4	-0.26
0.1	0.1	4.0±0.0	-0.37	5.0±0.2	-0.27	4.2±0.4	-0.35	4.2±0.5	-0.35	4.0±0.0	-0.37
	0.2	4.0±0.0	-0.3	5.0±0.2	-0.23	4.2±0.4	-0.3	4.3±0.7	-0.3	4.1±0.6	-0.38
	0.5	4.0±0.0	0.04	5.0±0.2	0.05	5.1±1.5	-0.02	5.2±1.7	-0.02	4.6±1.4	-0.39
	1.0	4.0±0.0	0.17	5.1±0.3	0.2	5.4±1.9	0.12	6.4±2.2	0.16	4.9±1.6	-0.42
	1.5	4.0±0.2	0.34	5.1±0.7	0.41	5.0±0.0	0.4	6.3±1.7	0.38	5.5±2.4	-0.38
	5.0	4.0±0.0	0.4	5.5±1.4	0.55	4.0±0.0	0.4	4.0±0.0	0.4	7.8±5.1	-0.12
	10.0	4.0±0.0	0.4	5.0±1.1	0.5	4.0±0.0	0.4	4.0±0.0	0.4	7.7±5.4	-0.01
	15.0	4.0±0.0	0.4	4.3±0.7	0.43	5.0±0.0	0.5	4.8±0.4	0.48	8.3±6.4	0.06
	20.0	4.0±0.0	0.4	4.2±0.5	0.42	4.0±0.0	0.4	4.0±0.0	0.4	7.6±5.4	-0.01

Table A.5: Comparison table: results of clustering approach vs. confidence approach 2 (with trend = 0.1, over sine data)

A. APPENDIX: EXHAUSTIVE RESULTS

C(t)	$\eta(t)$	ECONOMY-K				ECONOMY-γ					
		$\overline{\tau}^{\star}_{\text{ETM}}$	$\overline{\text{C}}_{\text{RCM}}$	$\overline{\tau}^{\star}_{\text{PETM}}$	$\overline{\text{C}}_{\text{PRCM}}$	$\overline{\tau}^{\star}_{\text{ETM}}$	$\overline{\text{C}}_{\text{RCM}}$	$\overline{\tau}^{\star}_{\text{PETM}}$	$\overline{\text{C}}_{\text{PRCM}}$	$\overline{\tau}^{\star}_{\text{ITM}}$	$\overline{\text{C}}_{\text{ICM}}$
DistalPhalanxOC	0.001	8.0±0.0	-0.31	9.0±1.4	-0.34	21.7±13.1	-0.42	54.7±14.0	-0.44	14.1±13.9	-0.88
	0.01	8.0±0.0	-0.23	8.6±0.9	-0.25	10.9±2.5	-0.25	13.8±3.5	-0.23	14.1±13.9	-0.76
	0.05	8.0±0.0	0.09	8.2±0.4	0.09	8.9±1.9	0.11	9.2±2.1	0.11	10.3±6.4	-0.22
	0.07	8.0±0.0	0.25	8.2±0.4	0.25	8.0±0.0	0.25	8.2±0.9	0.22	9.8±5.2	-0.02
	0.1	8.0±0.0	0.49	8.1±0.4	0.48	8.0±0.0	0.49	8.1±0.4	0.48	8.9±2.5	0.25
ECGFiveDays	0.001	16.5±4.1	-0.6	16.9±3.1	-0.61	58.7±16.2	-0.92	63.0±15.2	-0.91	16.8±10.7	-0.98
	0.01	13.0±0.0	-0.51	15.8±1.6	-0.47	17.9±9.4	-0.5	23.7±10.1	-0.44	16.8±10.7	-0.83
	0.05	13.0±0.0	0.01	14.9±1.4	0.12	13.2±0.4	0.03	13.9±1.4	0.05	15.1±6.2	-0.18
	0.07	13.0±0.0	0.27	14.1±1.4	0.35	13.0±0.0	0.27	13.2±0.6	0.3	14.7±5.2	0.12
	0.1	13.0±0.0	0.66	13.4±1.0	0.71	13.0±0.0	0.66	13.1±0.4	0.69	14.1±3.9	0.55
ItalyPowerDemand	0.001	19.1±2.0	-0.92	10.1±7.8	-0.48	16.7±4.3	-0.91	17.4±3.5	-0.82	4.4±3.5	-1.0
	0.01	12.1±7.8	-0.55	6.4±6.1	-0.31	13.9±5.2	-0.73	14.1±3.5	-0.65	4.4±3.5	-0.96
	0.05	2.1±0.3	-0.08	2.4±0.8	-0.07	2.4±0.5	-0.06	5.2±3.5	-0.14	4.4±3.5	-0.78
	0.07	2.0±0.2	-0.04	2.2±0.7	-0.03	2.4±0.5	-0.01	2.7±1.6	-0.02	4.4±3.5	-0.7
	0.1	2.0±0.0	0.02	2.1±0.3	0.02	2.3±0.4	0.04	2.2±0.4	0.04	4.4±3.5	-0.56
MiddlePhalanxOC	0.001	9.8±1.0	-0.28	31.9±21.0	-0.34	31.9±14.9	-0.51	53.9±15.7	-0.54	14.9±13.8	-0.89
	0.01	9.0±0.0	-0.17	12.8±3.6	-0.19	14.0±5.3	-0.2	17.0±4.4	-0.18	14.9±13.8	-0.76
	0.05	9.0±0.0	0.19	8.8±0.5	0.18	10.7±0.6	0.29	10.6±0.7	0.27	12.0±9.0	-0.19
	0.07	9.0±0.0	0.37	8.7±0.5	0.35	10.2±0.8	0.47	9.9±1.3	0.43	10.6±6.4	0.03
	0.1	9.0±0.0	0.64	8.6±0.5	0.59	9.3±1.2	0.67	9.3±1.2	0.66	9.4±3.6	0.32
MoteStrain	0.001	60.9±14.9	-0.63	40.8±29.9	-0.65	15.7±13.6	-0.69	37.3±19.8	-0.69	10.4±8.9	-0.96
	0.01	13.7±2.6	-0.51	13.5±2.3	-0.53	10.9±6.4	-0.59	12.1±5.3	-0.54	10.4±8.9	-0.87
	0.05	11.0±1.3	-0.13	12.1±1.6	-0.09	8.0±0.0	-0.27	8.4±0.9	-0.28	9.6±5.9	-0.46
	0.07	9.4±1.3	-0.02	11.4±1.7	0.11	8.0±0.0	-0.11	8.2±0.6	-0.13	8.8±3.2	-0.28
	0.1	9.2±1.3	0.24	11.0±1.5	0.42	8.0±0.0	0.13	8.1±0.4	0.12	8.8±3.0	-0.02
PhalangesOC	0.001	9.1±1.4	-0.29	17.1±8.9	-0.32	34.5±26.1	-0.45	46.7±21.1	-0.45	16.0±16.4	-0.9
	0.01	8.7±0.6	-0.22	10.5±2.9	-0.22	9.7±2.0	-0.23	13.7±7.3	-0.23	16.0±16.4	-0.76
	0.05	8.4±0.5	0.13	8.6±0.5	0.13	9.1±1.3	0.15	9.8±1.9	0.18	11.3±8.7	-0.16
	0.07	8.2±0.4	0.28	8.5±0.5	0.3	8.5±1.0	0.3	9.2±1.5	0.34	9.6±5.1	0.04
	0.1	8.0±0.0	0.51	8.4±0.5	0.54	8.4±0.8	0.55	8.5±0.9	0.56	8.6±2.0	0.31

Table A.6: Results (1) - Performance of ECONOMY-K vs ECONOMY-γ over real data sets.

C(t)	$\eta(t)$	ECONOMY-K				ECONOMY-γ					
		$\overline{\tau}^{\star}_{\text{ETM}}$	$\overline{\text{C}}_{\text{RCM}}$	$\overline{\tau}^{\star}_{\text{PETM}}$	$\overline{\text{C}}_{\text{PRCM}}$	$\overline{\tau}^{\star}_{\text{ETM}}$	$\overline{\text{C}}_{\text{RCM}}$	$\overline{\tau}^{\star}_{\text{PETM}}$	$\overline{\text{C}}_{\text{PRCM}}$	$\overline{\tau}^{\star}_{\text{ITM}}$	$\overline{\text{C}}_{\text{ICM}}$
ProximalPhalanxOC	0.001	12.9±4.4	-0.5	14.9±3.9	-0.49	19.9±15.5	-0.54	30.8±21.9	-0.57	11.4±10.4	-0.9
	0.01	8.0±0.0	-0.38	13.4±4.4	-0.37	14.3±8.9	-0.42	16.5±9.9	-0.39	11.4±10.4	-0.8
	0.05	8.0±0.0	-0.06	8.1±0.3	-0.06	9.2±1.6	0.0	10.3±2.1	-0.02	10.2±6.8	-0.36
	0.07	8.0±0.0	0.1	8.1±0.2	0.1	8.9±1.7	0.16	10.1±2.1	0.17	9.3±4.2	-0.17
	0.1	8.0±0.0	0.34	8.0±0.2	0.34	8.9±1.7	0.43	9.6±1.8	0.46	8.9±2.8	0.1
SonyAIBORobotS	0.001	19.5±5.1	-0.76	12.0±4.9	-0.72	14.8±11.7	-0.77	25.5±16.1	-0.82	8.0±4.5	-0.99
	0.01	7.0±0.0	-0.71	9.6±1.7	-0.63	10.1±5.5	-0.66	11.9±4.1	-0.62	8.0±4.5	-0.92
	0.05	7.0±0.0	-0.43	8.3±1.3	-0.36	8.3±2.1	-0.36	9.3±2.2	-0.28	7.8±3.8	-0.6
	0.07	7.0±0.0	-0.29	7.8±1.1	-0.22	8.3±2.1	-0.2	8.9±2.2	-0.14	7.7±3.0	-0.45
	0.1	7.0±0.0	-0.08	7.3±0.6	-0.04	8.1±2.1	0.03	8.4±2.1	0.06	7.5±2.1	-0.22
SonyAIBORobotSII	0.001	7.9±3.0	-0.56	10.1±3.8	-0.72	16.7±12.2	-0.86	23.3±16.2	-0.86	7.0±2.5	-0.99
	0.01	6.3±0.7	-0.52	8.0±2.3	-0.6	13.7±6.0	-0.72	13.8±5.3	-0.74	7.0±2.5	-0.92
	0.05	6.0±0.0	-0.28	7.0±1.2	-0.23	6.0±0.0	-0.28	8.1±2.9	-0.31	7.0±2.5	-0.65
	0.07	6.0±0.0	-0.16	6.6±1.0	-0.14	6.0±0.0	-0.16	6.7±1.7	-0.16	7.0±2.5	-0.51
	0.1	6.0±0.0	0.02	6.3±0.7	0.03	6.0±0.0	0.02	6.3±1.0	0.02	6.9±2.1	-0.3
Strawberry	0.001	25.8±0.4	-0.4	25.8±1.3	-0.37	51.8±38.8	-0.85	57.0±17.4	-0.86	26.7±9.0	-0.96
	0.01	24.6±1.4	-0.16	25.7±1.1	-0.14	37.0±11.3	-0.41	38.3±11.2	-0.41	26.7±9.0	-0.72
	0.05	23.0±0.0	0.73	25.0±0.5	0.84	23.9±1.9	0.76	24.8±2.4	0.79	26.1±6.5	0.34
	0.07	23.0±0.0	1.19	24.8±0.7	1.33	23.0±0.0	1.19	23.4±1.2	1.22	25.6±5.3	0.86
	0.1	23.0±0.0	1.88	24.7±0.7	2.05	23.0±0.0	1.88	23.2±0.8	1.9	25.0±4.1	1.62
TwoLeadECG	0.001	8.0±0.0	-0.36	17.7±2.9	-0.83	25.1±9.0	-0.95	26.8±9.3	-0.94	9.0±2.1	-0.99
	0.01	8.0±0.0	-0.29	16.1±3.4	-0.64	15.8±5.6	-0.76	17.1±5.6	-0.75	9.0±2.1	-0.91
	0.05	8.0±0.0	0.03	8.6±2.2	0.06	10.1±2.5	-0.14	10.3±2.6	-0.13	9.0±2.1	-0.55
	0.07	8.0±0.0	0.19	8.1±1.1	0.2	8.7±1.6	0.13	9.5±1.9	0.06	9.0±2.1	-0.37
	0.1	8.0±0.0	0.43	8.0±0.2	0.43	8.0±0.0	0.43	8.9±1.5	0.38	9.0±2.1	-0.1

Table A.7: Results (2) - Performance of ECONOMY-K vs ECONOMY-γ over real data sets.

Nom	Taille	Longueur des séries
ECGFiveDays	884	136
ItalyPowerDemand	1096	24
MoteStrain	1272	84
SonyAIBORobot SurfaceII	980	65
SonyAIBORobot Surface	621	70
TwoLeadECG	1162	82
PhalangesOutlinesCorrect	2658	80
ProximalPhalanxOutlineCorrect	891	80
DistalPhalanxOutlineCorrect	876	80
MiddlePhalanxOutlineCorrect	891	80
Strawberry	983	235

Table A.8: Basic characteristics of the real data sets used in the experiments.

Résumé en Français

Dans de nombreux domaines dans lesquels les mesures ou les données sont disponibles séquentiellement, il est important de savoir décider le plus tôt possible, même si c'est à partir d'informations encore incomplètes. C'est le cas par exemple en milieu hospitalier où l'apprentissage de règles de décision peut se faire à partir de cas complètement documentés, mais où, devant un nouveau patient, il peut être crucial de prendre une décision très rapidement. Dans ce type de contextes, un compromis doit être optimisé entre la possibilité d'arriver à une meilleure décision en attendant des mesures supplémentaires, et le coût croissant associé à chaque nouvelle mesure.

Ce problème est une tâche d'optimisation classique avec un compromis entre le gain d'informa- tion qui peut être attendu si la décision est retardée et la hausse du coût associé à un tel retard. Ce compromis a été connu depuis des décennies et a des racines historiques dans plusieurs domaines tels que la prise de décision séquentielle, les décisions séquentielles optimales, l'apprentissage sous contraintes, etc., mais de nombreuses nouvelles applications dans la médecine, la gestion du réseau électrique, le transport automatique, etc., ont donné un nouvel élan aux travaux dans ce domaine de recherche.

Des travaux récents, dans ce domaine, s'intéressent au problème de la classification précoce de séries temporelles pour assister à la prise de décision. Leur objectif commun consiste à optimiser en ligne le compromis entre la qualité et la précocité de prédictions afin de déterminer le meilleur instant auquel une prédiction peut être émise. La figure A.1 décrit le cadre général pour décider à quel instant prédire la classe d'une série temporelle en entrée.

Différentes approches ont été proposées pour résoudre ce problème. Toutefois, la

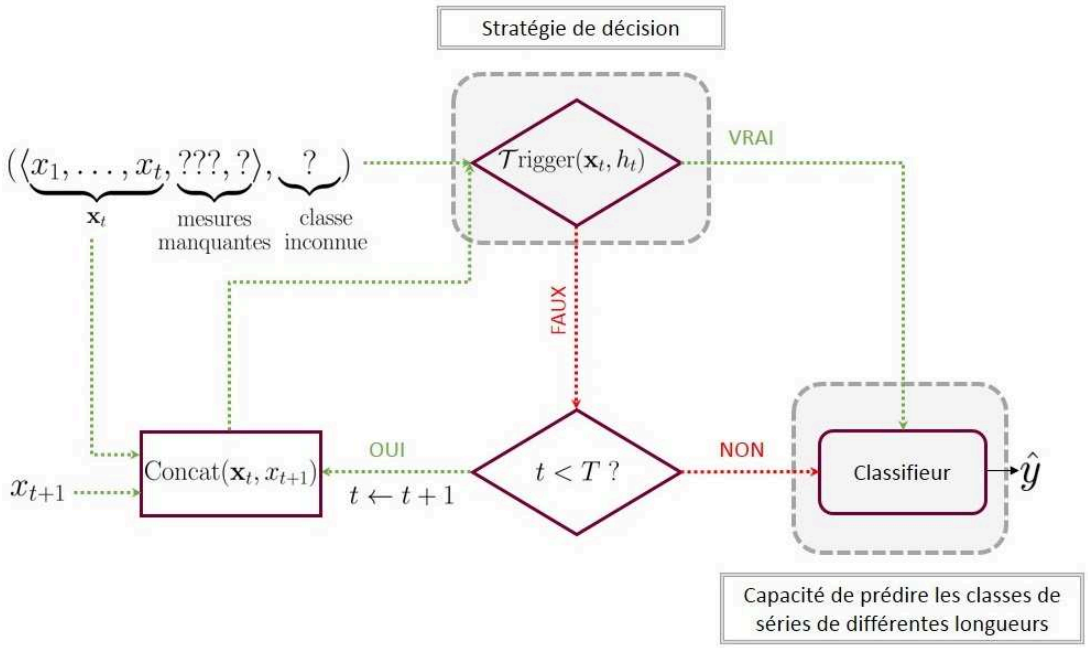


Figure A.1: Cadre général de classification précoce de séries temporelles

majorité se révèle concernée par le problème de la classification de séries temporelles incomplètes, et même si la précocité de la décision est mentionnée comme une motivation dans ces travaux, les procédures de décision elles-même ne la prennent pas en compte explicitement. Elles évaluent plutôt la confiance ou la fiabilité de la prédiction afin de décider s'il est opportun de faire une prédiction immédiate, ou s'il semble préférable d'attendre une donnée supplémentaire. En outre, les procédures sont myopes car elles se limitent à l'instant courant pour décider si une prédiction doit être faite.

Dans ce contexte, nous soutenons que dès que la précocité est impliquée dans un processus de prise de décision en ligne, le coût d'attente doit être explicitement pris en compte dans le critère d'optimisation. À notre connaissance, il n'existe pas de méthodes de classification précoce qui optimisent explicitement le compromis entre le gain d'information qui devrait conduire à moins d'erreurs de prédiction et donc des coûts de prédictions plus faibles et le coût croissant associé au report de cette décision.

Dans cette thèse, nous adressons, donc, le problème de la classification précoce comme un problème de décision en ligne qui implique deux types de coût : (i) le coût associé aux

erreurs de prédiction et (ii) le coût associé au report de la décision[1]. Nous présentons **ECONOMY** (**E**arly **C**lassification for **O**ptimized and **NO**n-**MY**opic online decision making), une approche conçue pour la prédiction précoce à partir de séries temporelles incomplètes et pour la prise de décision en ligne. Notre approche générique est ensuite résolue par deux mécanismes qui capturent l'évolution typique de séries temporelles d'apprentissage pour parvenir à estimer, dans le futur, l'instant optimal pour prendre une décision.

ECONOMY : une nouvelle formalisation

L'objectif est de concevoir une procédure de décision qui permet de déterminer l'instant optimal $t^\star$ auquel une nouvelle série temporelle $\mathbf{x}_t^\star = \langle x_1, x_2, \ldots, x_t^\star \rangle$ peut être classée de façon optimale. Pour ce faire, nous associons un coût à la qualité de prédiction et un coût à l'instant pendant lequel la prédiction est finalement effectuée :

- nous supposons qu'une fonction de coût est associée aux prédictions erronées $\mathrm{C}_t(\hat{y}|y)$: $\mathcal{Y} \times \mathcal{Y} \longrightarrow \mathbb{R}$. Cette fonction fournit le coût estimé à l'instant $t^\star$ pour avoir prédit $\hat{y}$ lorsque la vraie classe est y.

- à chaque instant t est associée une fonction de coût temporel qui signifie qu'il est toujours plus coûteux d'attendre de faire une prédiction[2].

  La fonction du coût pour le problème de décision est alors définie comme:

$$f(\mathbf{x}_t) \;=\; \sum_{y \in \mathcal{Y}} P(y|\mathbf{x}_t) \sum_{\hat{y} \in \mathcal{Y}} P(\hat{y}|y, \mathbf{x}_t)\, \mathrm{C}_t(\hat{y}|y) \;+\; \mathrm{C}(t) \tag{A.1}$$

  Cette équation correspond à l'estimation des coûts de classification après t pas de temps, et à laquelle est ajouté le coût d'avoir retardé la décision jusqu'à l'instant t.

  L'instant optimal t^* est alors défini comme :

$$t^* \;=\; \underset{t \in \{1,\ldots,T\}}{\mathrm{ArgMin}}\; f(\mathbf{x}_t) \tag{A.2}$$

[1] Ces deux coûts doivent être exprimés dans la même unité (e.g. unité monétaire).

[2] À noter que cette fonction, contrairement à la plupart des méthodes de l'état-de-l'art peut être autre qu'une fonction linaire et doit être fixée selon les particularités du domaine d'application. Par exemple, si la tâche est de prédire de décider si une centrale thermique doit démarrer ou non, le coût d'attente augmente fortement en s'approchant des derniers instants, ce qui donne une forme exponentielle à la fonction de coût.

Forces, limites et nouveautés

L'originalité de ECONOMY est triple. Tout d'abord, le problème de la classification précoce de séries temporelles est formalisé comme un problème de prise de décision en ligne impliquant les deux coûts associés (i) à la qualité de prédiction et (ii) au report de la prise de décision. Deuxièmement, la méthode est adaptative, en ce que les propriétés de la série temporelle en entrée sont prises en compte pour décider quel sera le meilleur instant pour faire une prédiction. Troisièmement, contrairement aux techniques de prise de décision habituelles, l'algorithme présenté est générique et offre plusieurs possibilités pour le résoudre.

Par ailleurs, de son caractère générique, cette formulation du problème de décision ne donne pas facilement une méthode pour trouver, en ligne, le temps de décision optimal t^*. Elle exige que l'on soit en mesure de calculer les probabilités conditionnelles $P(y|\mathbf{x}_t)$ et $P(\hat{y}|y, \mathbf{x}_t)$, qui sont difficiles à estimer. La première est inconnue, sinon il n'y aurait pas de problème d'apprentissage en premier lieu. La seconde est associée à un classifieur donné, et est également difficile à estimer (ces termes sont difficiles à estimer sur une seule série temporelle. Ceci nécessite qu'une certaine généralisation sur un éventuel espace de séries temporelles soit effectuée.)

Une façon naïve pour résoudre ce problème et estimer facilement ces probabilités conditionnelles serait de calculer l'espérance des coûts pour n'importe quelle série temporelle. De ce fait, la fonction de décision est maintenant notée $f(t)$:

$$f(t) \;=\; \sum_{y \in \mathcal{Y}} P(y) \sum_{\hat{y} \in \mathcal{Y}} P(\hat{y}|y)\, \mathrm{C}_t(\hat{y}|y) \;+\; \mathrm{C}(t) \tag{A.3}$$

À partir de l'ensemble d'apprentissage $\mathcal{S}$, il est en effet facile de calculer les probabilités $P(y)$ et $P(\hat{y}|y)$ qui n'est autre que la matrice de confusion associée au classifieur considéré. On obtient alors le temps optimal pour la prédiction:

$$t^* \;=\; \underset{t \in \{1,\ldots,T\}}{\mathrm{ArgMin}}\, f(t) \tag{A.4}$$

En simplifiant le problème, cela peut être calculé avant même l'arrivée de toute nouvelle série temporelle, et en fait, l'instant optimal t^* est indépendant de la série en entrée.

Cependant, ce que nous cherchons est de prendre en compte les caractéristiques de la série en entrée et décider de l'instant optimal pour faire une prédiction selon l'évolution

de la série elle-même. Il s'avère donc absurde d'estimer le même instant de décision pour des séries temporelles qui sont différentes.

Proposition

Pour surmonter ces difficultés pour estimer les termes $P(y|\mathbf{x}_t)$ et $P(\hat{y}|y, \mathbf{x}_t)$, l'idée consiste à capturer les caractéristiques de la série temporelle en entrée en utilisant une technique de segmentation qui permet de construire un ensemble cohérent de groupes. Sur la base de ces groupes, l'estimation de ces termes devient possible permettant ainsi d'estimer la fonction de coûts tout en prenant en compte les caractéristiques de la série temporelle d'entrée, $\mathbf{x}_t$.

Dans l'objectif d'estimer les probabilités conditionnelles $P(y|\mathbf{x}_t)$ et $P(\hat{y}|y, \mathbf{x}_t)$ dans l'équation A.1, nous proposons de : (i) segmenter l'ensemble d'apprentissage, contenant des séries complètes, en un ensemble de groupes cohérents noté $\{G_k\}_{1 \leq k \leq K}$. En se basant sur ces groupes, l'idée est de substituer le terme $P(\hat{y}|y, \mathbf{x}_t)$ par le terme désormais estimable $P(\hat{y}|y, G_k)$. Ensuite, (ii) définir une fonction de coût capable de fournir l'instant optimal dans le futur pour une nouvelle série $\mathbf{x}_t$.

Segmentation

L'idée de segmenter les séries temporelles *complètes* consiste à tirer parti de l'information complète et de différents comportements identifiés dans l'ensemble de données d'apprentissage pour faire une estimation adaptative de coûts futurs, compte tenu d'une série temporelle entrante $\mathbf{x}_t$.

Plus précisément, les séries temporelles d'apprentissage, qui sont complètes, doivent être segmentées en groupes cohérents $\{G_k\}_{1 \leq k \leq K}$ à l'aide d'une méthode de segmentation spécifique. Ces groupes seront utilisés plus tard pour calculer les termes $P(\hat{y}|y, G_k)$ qui ne sont d'autres que les termes contenus dans chaque case d'une matrice de confusion calculée pour chaque groupe. Cependant, la construction de ces groupes devrait respecter deux contraintes:

1. Les différents groupes doivent correspondre à différentes matrices de confusion.
2. Les groupes devraient contenir des séries temporelles similaires et être différents des autres groupes.

Dans la contrainte (1), le terme $P(\hat{y}|y, G_k)$ est calculé à partir de séries temporelles appartenant au même groupe G_k à l'aide d'un classificateur déjà appris. Les termes

$P(\hat{y}|y, G_k)$, $1 \leq k \leq K$ calculés sur chaque groupe G_k devraient être différents autant que possible afin de discriminer les coûts entre les groupes. Pour ce faire, l'objectif est de former des groupes différents autant que possible. Brièvement, la segmentation devrait aider à accomplir une tâche supervisée.
Dans la contrainte (2), des séries temporelles similaires, regroupées à l'aide d'une fonction ou mesure de similarité spécifique, doivent appartenir au même groupe et doivent être différentes des séries temporelles d'autres groupes. De cette façon, une série temporelle entrante sera généralement attribuée de manière marquée à l'un de ces groupes. En tant que tel, la segmentation devrait aider à rendre la fonction de coûts adaptée aux séries temporelles entrantes.

Estimation des coûts futurs

La deuxième idée que nous proposons pour surmonter la difficulté de calculer $f(\mathbf{x}_t)$ pour tout $t \in \{1, \ldots, T\}$ est de calculer à l'avance, au temps t, les coûts de décision prévus pour tous les instants futurs. Ceci est possible grâce à l'estimation des termes $P_t(\hat{y}|y, G_k)$ pour chaque groupe G_k, $1 \leq k \leq K$ et à chaque instant t, puis l'estimation de l'appartenance de la série temporelle entrante $\mathbf{x}_t$ à chaque groupe G_k. Cette appartenance permet d'identifier des informations pertinentes pour les instants futurs conditionnellement à la série $\mathbf{x}_t$.

Plus précisément, étant donné qu'à l'instant t, $T - t$ points de mesures sont encore manquants dans la série temporelle entrante $\mathbf{x}_t$, il est possible de calculer le coût de décision prévu pour la classification de $\mathbf{x}_t$ à chaque instant futur $\tau \in \{0, \ldots, T - t\}$.
Maintenant, supposons qu'il existe une fonction f_τ qui estime le coût prévu pour les instants futurs τ en utilisant les informations complètes obtenues en fonction de groupes ainsi formés. Cela permet de prévoir l'horizon optimal $t^\star$ pour classifier la série temporelle $\mathbf{x}_t$:

$$t^* = t + \underset{\tau \in \{0,\ldots,T-t\}}{\text{ArgMin}} \; f_\tau(\mathbf{x}_t) \tag{A.5}$$

Bien sûr, ces coûts futurs, donnés par $f_\tau(\mathbf{x}_t)$, où $\tau \in \{0, \ldots, T - t\}$ et l'horizon optimal estimé $t^* = t + \tau^\star$, où $\tau^\star = \text{ArgMin}_{\tau \in \{0,\ldots,T-t\}} f_\tau(\mathbf{x}_t)$, peuvent être réévalués quand un nouveau point de mesure est ajouté à la série temporelle entrante. La figure 5.1 illustre cette idée.

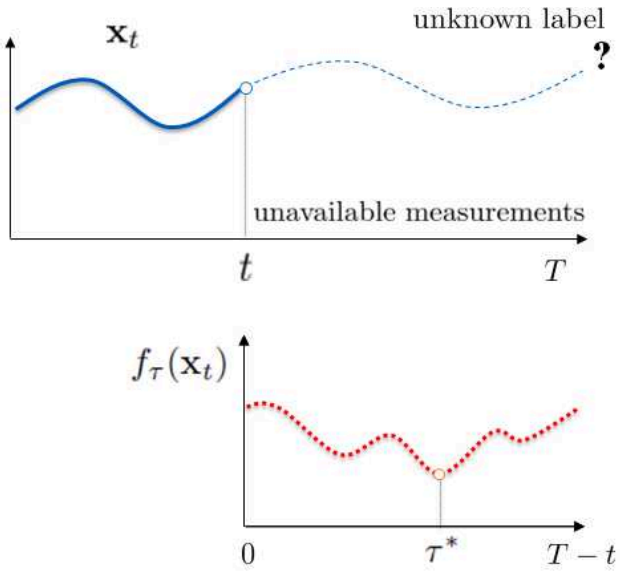


Figure A.2: La première courbe représente une série temporelle entrante $\mathbf{x}_t$. La deuxième courbe représente le coût de décision prévu $f_\tau(\mathbf{x}_t)$ étant donnée $\mathbf{x}_t$, $\forall \tau \in \{0, \dots, T-t\}$. Cela montre l'équilibre entre le gain dans la précision attendue de la prédiction et le coût de l'attente avant de décider. Le minimum de ce compromis devrait se produire à l'instant $t + \tau^\star$. Les nouveaux points de mesure peuvent modifier la courbe du coût de décision prévu et la valeur estimée de l'instant optimal de décision $\tau^\star$.

Politique de décision

Notre politique de décision est définie comme suit. À tout instant t, si l'horizon optimal $\tau^\star = 0$ et pour $\tau > 0$, $f_\tau(\mathbf{x}_t) > f_0(\mathbf{x}_t)$, alors le processus de décision séquentielle s'arrête et une prédiction est faite sur la classe de la série temporelle entrante $\mathbf{x}_t$ utilisant le classificateur h_t : $\hat{y} = h_t(\mathbf{x}_t)$ (ici, un ensemble de classifieurs $\mathcal{H} = \{h_t\}_{1 \leq t \leq T}$ est utilisé où chaque classifieur h_t est appris pendant la phase d'apprentissage sur des séries temporelles tronquées à leurs t premiers éléments. D'autres types de classifieurs conçus pour prédire la classe de séries temporelles de différentes longueurs, peuvent être utilisés). D'autres politiques de décision peuvent être utilisées comme une version moins forte de la règle que nous proposons pour décider de l'instant optimal.

Dans l'objectif d'estimer les probabilités conditionnelles $P(y|\mathbf{x}_t)$ et $P(\hat{y}|y, \mathbf{x}_t)$ dans l'équation A.1 en se basant sur la méthodologie présentée plus haut, nous proposons deux approches différentes pour résoudre et étendre ECONOMY.

1. La première approche, baptisée ECONOMY-K, est intuitive et simple à mettre en

oeuvre. Basée sur une approche de clustering, ECONOMY-K définit une nouvelle fonction de coût f_τ permettant d'estimer l'instant optimal pour classifier une série en entrée $\mathbf{x}_t$ grâce aux différents clusters formés.

2. La deuxième approche, nommée ECONOMY-γ, définit aussi une nouvelle fonction de coût f_τ mais, contrairement à ECONOMY-K, elle utilise une méthode de segmentation supervisée.

ECONOMY-K : Approche basée sur une segmentation non supervisée

ECONOMY-K est une approche basée sur une segmentation non supervisée de l'ensemble d'apprentissage utilisant une méthode de clustering. Cette approche permet d'identifier, à partir de l'ensemble d'apprentissage, un ensemble $\mathcal{C} = \{\mathfrak{c}_k\}_{1 \leq k \leq K}$ composé de K sous-ensembles de séries temporelles. De cette manière, la série incomplète en entrée $\mathbf{x}_t$ peut être associée aux différents sous-ensembles grâce à la probabilité conditionnelle $P(\mathfrak{c}_k|\mathbf{x}_t)$, où $1 \leq k \leq K$. Cela permet de comparer la nouvelle série $\mathbf{x}_t$ aux différentes continuations possibles, présentes dans les séries complètes, qui peuvent la compléter. L'approche ECONOMY-K procède en deux étapes :

Phase d'apprentissage : Soit $\mathcal{H} = \{h_t\}_{1 \leq t \leq T}$, un ensemble de classifieurs où chaque classifieur h_t est appris sur un ensemble d'apprentissage $\mathcal{S}_t$ composé de séries temporelles de longueur t. Chaque classifieur est appliqué par la suite sur chaque sous-ensemble $\mathfrak{c}_k$ pour estimer les matrices de confusion associées $P_t(\hat{y}|y, \mathfrak{c}_k)$.

Phase de test : À l'arrivée d'une nouvelle série temporelle incomplète $\mathbf{x}_t, t < T$, la probabilité conditionnelle $P(\mathfrak{c}_k|\mathbf{x}_t)$ est calculée par rapport à chaque sous-ensemble $\mathfrak{c}_k$. Ensuite, étant donné que $T - t$ points de mesure sont encore non observés, il est possible d'estimer l'espérance de coût pour prédire la classe de $\mathbf{x}_t$ à chaque instant futur $\tau \in \{0, \ldots, T - t\}$:

$$f_\tau(\mathbf{x}_t) = \sum_{\mathfrak{c}_k \in \mathcal{C}} P(\mathfrak{c}_k|\mathbf{x}_t) \sum_{y \in \mathcal{Y}} P(y|\mathfrak{c}_k) \sum_{\hat{y} \in \mathcal{Y}} P_{t+\tau}(\hat{y}|y, \mathfrak{c}_k)\, \mathrm{C}(\hat{y}|y) + \mathrm{C}(t + \tau) \tag{A.6}$$

L'équation A.6 fournit deux propriétés remarquables. Premièrement, les probabilités conditionnelles qui permettent de prendre en compte la série incomplète en entrée, $P(y|\mathfrak{c}_k)$, $P(\mathfrak{c}_k|\mathbf{x}_t)$ et $P_t(\hat{y}|y, \mathfrak{c}_k)$ sont désormais facilement estimables à partir de l'ensemble

d'apprentissage. Deuxièmement, le coût dépend de la série en entrée $\mathbf{x}_t$ par le biais de la probabilité d'appartenance $P(\mathfrak{c}_k|\mathbf{x}_t)$.

L'instant optimal de décision est défini alors par :

$$\begin{aligned} t^* &= t + \underset{\tau \in \{0,\dots,T-t\}}{\text{ArgMin}} f_\tau(\mathbf{x}_t) \\ &= t + \tau^\star \end{aligned} \tag{A.7}$$

Avec l'arrivée de nouveaux points de mesure, cet horizon τ^* peut être mis à jour si les nouveaux points apportent une nouvelle information utile pour l'amélioration de la qualité de prédiction. Pour tout instant t, si $\tau^* = 0$, alors le processus de décision séquentielle s'arrête et l'étiquette de $\mathbf{x}_t$ est prédite par le classifieur h_t : $\hat{y} = h_t(\mathbf{x}_t)$.

Forces, limites et perspectives

Contrairement aux approches existantes qui essaient de faire de prédictions précoces mais qui ne tiennent pas en compte, d'une manière explicite, le coût d'attente de décision, ECONOMY-K propose de résoudre le critère d'optimisation adaptatif ECONOMY qui exprime le problème de décision par l'espérance des coûts d'erreurs de prédiction ajoutée au coût d'attente. ECONOMY-K fournit une fonction de décision qui décide de l'instant optimal pour faire une prédiction tout en prenant en compte les caractéristiques de la série incomplète en entrée $\mathbf{x}_t$ via une technique de segmentation et une manière d'appartenir aux différents sous-ensembles obtenus.

En plus de son comportement adaptatif, ECONOMY-K est une procédure de décision non myope car elle estime à chaque instant, l'horizon futur auquel une décision optimale peut être prise.

Cependant, lors de l'étape de la segmentation, ECONOMY-K utilise une technique de segmentation qui implique le réglage de certains paramètres: (i) le choix de la méthode de segmentation, (ii) le choix de la mesure de similarité pour construire les sous-ensembles de séries temporelles et pour mesurer la distance entre une série temporelle incomplète et un sous-ensemble composé de séries complètes et (iii) le nombre de sous-ensembles K à fixer. Selon les résultats empiriques obtenus, le réglage de ces paramètres influe significativement sur le bon fonctionnement de l'approche.

Par ailleurs, en utilisant une technique de segmentation non supervisée, nous n'avons pas de garanties que tous les choix et les paramètres ainsi réglés ont du sens par rapport à la classe. Ici, le problème supervisé est aidé par une tâche non supervisée. Dans la suite, nous n'étudions pas l'influence de ce choix (segmentation supervisée/non super-

visée) sur la résolution du problème global, mais nous proposons une deuxième approche ECONOMY-γ qui utilise une technique de segmentation supervisée pour résoudre le problème global aussi supervisé.

ECONOMY-γ

Dans la deuxième approche ECONOMY-γ, nous adoptons le même cadre conceptuel proposé dans l'approche ECONOMY-K, i.e. calculer les coûts pour les instants futurs en se basant sur une segmentation de l'ensemble d'apprentissage. Notre objectif est d'exploiter, en plus des séries complètes, leur classe associée afin de faire une segmentation plus informative que le clustering.

Cette nouvelle approche propose donc une segmentation intelligente et naturelle de l'ensemble d'apprentissage par l'utilisation d'une chaîne de Markov qui permet de saisir à la fois l'évolution typique de séries temporelles d'apprentissage qui sont complètes et les suites possibles de toute série temporelle incomplète. Pour capturer ces informations pertinentes, les états de la chaîne de Markov devraient être fixés de manière significative afin de mieux saisir les transitions d'un état à un autre.

ECONOMY-γ se déroule en trois étapes principales :

1. Spécification de la chaîne de Markov et détermination de ses états afin de capturer les évolutions typiques des séries temporelles d'apprentissage et fournir des suites possibles de la série temporelle incomplète entrante $\mathbf{x}_t$,

2. La segmentation de séries temporelles d'apprentissage est effectuée sur la base de comportements significatifs détectés par la chaîne de Markov,

3. Estimation des termes de la fonction de coûts afin de déterminer l'instant optimal $t^\star$ pour prendre une décision.

La fonction de coûts est alors définie par :

$$f_\tau(\mathbf{x}_t) \;=\; \sum_{y,\hat{y}\in\mathcal{Y}} \sum_{\ell=1}^{N} P(\gamma_{t+\tau} = \ell | \langle \gamma_t \rangle)\; P_{t+\tau}(\hat{y}|y, \gamma_{t+\tau} = \ell) \;\times \mathrm{C}(\hat{y}|y) + \mathrm{C}(t+\tau) \quad \text{(A.8)}$$

L'équation (A.8) représente une version très simplifiée de la règle de décision que nous proposons (voir Section **??** pour plus de détails).

Forces, limites et perspectives

La méthode ECONOMY-γ présente plusieurs avantages:

1. Outre le choix de la classe de fonctions de décision h qui doivent être réalisées quelle que soit l'approche, il existe deux paramètres à définir. Le premier est N, le nombre d'intervalles de confiance ou les états de chaîne de Markov que l'on est prêt à considérer à chaque pas temporel. Des valeurs plus élevées de N peuvent sembler préférables parce qu'elles produiraient une plus grande précision. Le deuxième paramètre est l'ordre de la dépendance pris en compte, de même que les modèles de chaînes de Markov qui peuvent dépendre du passé à divers degrés.

2. Les matrices de confusion qui apparaissent dans l'équation **??** ont tendance à différer, ce qui conduit à de meilleures estimations des coûts de décision futurs.

3. Les probabilités conditionnelles $P_{t+\tau}(\hat{y}|y, \gamma_{t+\tau} = \ell)$ ont également tendance à différer pour différentes valeurs de l'intervalle de confiance ℓ , Ce qui favorise de meilleures prévisions.

En outre, on s'attend à ce que l'utilisation du même algorithme avec des dépendances temporelles plus élevées prises en compte améliorerait encore les performances. Ces modèles plus riches devraient en effet pouvoir extraire les informations utiles dans l'ensemble d'apprentissage et les nouvelles séries temporelles entrantes, et se rapprocher du temps de décision optimal et du coût optimal. Cependant, seuls des ensembles d'apprentissage très importants peuvent permettre à un algorithme d'apprentissage d'atteindre ce type de performance, en permettant l'apprentissage du grand nombre de dépendances conditionnelles impliquées dans ces modèles d'ordre supérieur.

A.1 Conclusion

Dans cette, nous avons revisité le problème de la classification précoce de séries temporelles lorsque retarder la prise de décision est coûteux. Nous avons posé le problème comme un problème de prise de décision en ligne sensible aux coûts de classification et d'attente et proposé un critère d'optimisation qui équilibre le gain attendu dans le coût de la classification à l'avenir avec le coût du retrait de la décision.
Dans ce cadre conceptuel, nous avons proposé deux algorithmes qui diffèrent selon la manière dont ils considèrent l'information contenue dans les séries temporelles d'apprentissage et dans les séries temporelles entrantes. Les deux approches sont adaptées aux particularités des séries chronologiques entrantes, car la fonction coût est réestimée avec chaque

information ajoutée et fournit des décisions non myopes. La première approche appelée ECONOMY- K est intuitivement attrayante, car elle fournit une solution simple au problème qui capture les évolutions typiques de la série temporelle à l'aide d'une technique de clustering. La deuxième méthode s'appelle ECONOMY - *gamma*. Il est plus direct et informé car il fournit un schéma naturel utilisant le concept de chaîne de Markov pour capturer les modèles généralisés dans les séries chronologiques d'entraînement. Il prend en compte les comportements typiques des séries chronologiques d'entraînement avec une précision qui dépend de N, du nombre d'intervalles de confiance considérés et de l'ordre de dépendance pris en compte. L'avantage d'ECONOMY - *gamma* contre ECONOMY-K, au-delà de l'implication de moins de paramètres d'utilisateur et de complexité de calcul concurrentielle, est l'utilisation d'une méthode de segmentation qui prend également en compte les informations sur les étiquettes de classe des séries chronologiques d'entraînement. Il réussit donc à regrouper des séries chronologiques décrites par la même classe en dépit de leur dissemblance. Cela rend la méthode de segmentation dans ECONOMY - *gamma* plus informée que le cluster utilisé dans l'approche ECONOMY-K qui segment les données de formation uniquement en fonction de leur forme.

References

[1] Aggarwal, C.C.: Data Streams: Models and Algorithms (Advances in Database Systems). Springer-Verlag New York, Inc., Secaucus, NJ, USA (2006) 17

[2] Aggarwal, C.C.: Data Classification: Algorithms and Applications. Chapman & Hall/CRC, 1st edn. (2014) 12

[3] Agrawal, R., Faloutsos, C., Swami, A.: Efficient Similarity Search In Sequence Databases. In: Proceedings of the 4th International Conference on Foundations of Data Organization and Algorithms (FODO). pp. 69–84 (1993) 13, 15

[4] Agrawal, R., Psaila, G., Wimmers, E.L., Zaït, M.: Querying Shapes of Histories. In: Proceedings of the 21th International Conference on Very Large Data Bases (VLDB '95). pp. 502–514 (1995) 13

[5] Allison, P.D.: Missing data, vol. 136. Sage publications (2001) 38

[6] Anderson, H., Parrish, N., Gupta, M.: Early time-series classification with reliability guarantee. Sandria Report (2012) 39, 59

[7] Antonucci, A., Scanagatta, M., Maua, D.D., de Campos, C.P.: Early classification of time series by hidden markov models with set-valued parameters. International workshop held at NIPS (2015) 63

[8] Aßfalg, J., Kriegel, H.P., Kröger, P., Kunath, P., Pryakhin, A., Renz, M.: Similarity Search on Time Series Based on Threshold Queries, pp. 276–294. Springer Berlin Heidelberg, Berlin, Heidelberg (2006) 15

[9] Babcock, B., Babu, S., Datar, M., Motwani, R., Widom, J.: Models and issues in data stream systems. In: Proceedings of the 21st ACM SIGMOD-SIGACT-SIGART symposium on Principles of Database Systems, PODS 02. pp. 1–16 (2002) 17

[10] Bellman, R., Kalaba, R.: On adaptive control processes. IRE Transactions on Automatic Control 4, 1–9 (1959) 15

[11] Bellman, R.E., Dreyfus, S.E.: Applied Dynamic Programming. Princetown University Press (1962) 7

[12] Berger, J.O.: Statistical decision theory and Bayesian analysis. Springer Science & Business Media (1985) 69

[13] Bernaille, L., Teixeira, R., Akodkenou, I., Soule, A., Salamatian, K.: Traffic classification on the fly. SIGCOMM Comput. Commun. Rev. 36(2), 23–26 (Apr 2006) 54, 64

[14] Berndt, D., Clifford, J.: Using dynamic time warping to find patterns in time series. In AAAI-94 workshop on knowledge discovery in databases 2, 359–370 (1994) 15, 42

[15] Block, H.: The perceptron: a model for brain functioning. Reviews of Modern Physics 34, 123–135 (1962) 12

[16] Bondu, A., Dachraoui, A.: Realistic and very fast simulation of individual electricity consumptions. In: International Joint Conference on Neural Networks (IJCNN). pp. 1–8 (2015) 127

[17] Boreczky, J.S., Rowe, L.A.: Comparison of Video Shot Boundary Detection Techniques. In: Storage and Retrieval for Still Image and Video Databases IV. pp. 170–179 (1996) 15

[18] Boullé, M.: MODL: A Bayes optimal discretization method for continuous attributes. Machine Learning 65(1), 131–165 (Oct 2006) 127

[19] Box, G.E.P., Jenkins, G.: Time Series Analysis, Forecasting and Control. Holden-Day, Incorporated (1990) 38

[20] Bregón, A., Simón, M.A., Rodríguez, J., Alonso, C., Pulido, B., Moro, I.: Early fault classification in dynamic systems using case-based reasoning. In: Proceedings of the 11th Spanish Association Conference on Current Topics in Artificial Intelligence. pp. 211–220. CAEPIA'05, Springer-Verlag (2006) 42, 53, 64

[21] Briggs, A., Clark, T., Wolstenholme, J., Clarke, P.: Missing.... presumed at random: cost-analysis of incomplete data. Health Economics 12(5), 377–392 (2003) 38

[22] Cha Zhang, Z.Z.: A Survey of Recent Advances in Face Detection. Tech. rep., Microsoft Research Technical Report (June 2010) 69

[23] Chan, K., Fu, A.: Efficient Time Series Matching by Wavelets. Proceedings of the 15th IEEE International Conference on Data Engineering pp. 126–133 (1999) 13, 44

[24] Chen, L., Ng, R.: On the marriage of lp-norms and edit distance. In: Proceedings of the Thirtieth international conference on Very large data bases (VLDB'04). vol. 30, pp. 792–803 (2004) 15

[25] Chen, L., ´Ozsu, M.T.: Robust and fast similarity search for moving object trajectories. In: In SIGMOD. pp. 491–502 (2005) 15

[26] Chen, M., Xu, Z.E., Weinberger, K.Q., Chapelle, O., Kedem, D.: Classifier cascade for minimizing feature evaluation cost. In: Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics, pp. 218–226. MIT Press (2012) 69

[27] Chen, Y., Nascimento, M.A., Chin, B., Anthony, O., Tung, K.H.: Spade: On shape-based pattern detection in streaming time series. In: IEEE 29th International Conference on Data Engineering (ICDE). pp. 786–795 (2007) 15

[28] Chiou, J.M.: Dynamical functional prediction and classification, with application to traffic flow prediction. Ann. Appl. Stat. 6(4), 1588–1614 (12 2012) 39

[29] Chiou, J.M., Li, P.L.: Functional clustering and identifying substructures of longitudinal data. Journal of the Royal Statistical Society Series B 69(4), 679–699 (2007) 39

[30] Chow, C.: On optimum recognition error and reject tradeoff. IEEE Transactions on Information Theory 16(1), 41–46 (Jan 1970) 61

[31] Cornuéjols, A.: On-line learning: where are we so far? In: Ubiquitous knowledge discovery, pp. 129–147. Springer (2010) 27

[32] Dachraoui, A., Bondu, A., Cornuéjols, A.: Early classification of individual electricity consumptions. In Internation Workshop (RealStream) held at ECML-PKDD pp. 18–21 (2013) 36

[33] Dachraoui, A., Bondu, A., Cornuéjols, A.: Early classification of time series as a non myopic sequential decision making problem. In: Machine Learning and Knowledge Discovery in Databases, pp. 433–447. Springer (2015) 3, 79

[34] Dainotti, A., Pescapé, A., Sansone, C.: Traffic Monitoring and Analysis: Third International Workshop, TMA 2011, Vienna, Austria, April 27, 2011. Proceedings, chap. Early Classification of Network Traffic through Multi-classification, pp. 122–135. Springer Berlin Heidelberg (2011) 63

[35] Dean, T., Boddy, M.S.: An analysis of time-dependent planning. In: Shrobe, H.E., Mitchell, T.M., Smith, R.G. (eds.) AAAI. pp. 49–54. AAAI Press / The MIT Press (1988) 27

[36] DeGroot, M.H.: Optimal statistical decisions, vol. 82. John Wiley & Sons (2005) 69

[37] Ding, H., Trajcevski, G., Scheuermann, P., Wang, X., Keogh, E.: Querying and mining of time series data: Experimental comparison of representations and distance measures. Proc. VLDB Endow. 1(2), 1542–1552 (Aug 2008) 15, 17

[38] Esling, P., Agon, C.: Time-series data mining. ACM Comput. Surv. 45(1), 12:1–12:34 (Dec 2012) 13

[39] Faloutsos, C., Ranganathan, M., Manolopoulos, Y.: Fast Subsequence Matching in Time-series Databases. In: In Proceedings of the ACM SIGMOD Conference. pp. 419–429 (1994) 17

[40] Freund, Y., Schapire, R.: A decision-theoretic generalization of on-line learning and an application to boosting. J. Comput. Syst. Sci. 55(1), 119–139 (Aug 1997) 52

[41] Fuad, M.M.M., Marteau, P.F.: The extended edit distance metric. In: International Workshop on Content-Based Multimedia Indexing (CBMI). pp. 242–248 (2008) 15

[42] Gaber, M.M., Zaslavsky, A., Krishnaswamy, S.: Mining data streams: A review. SIGMOD Rec. 34(2), 18–26 (Jun 2005) 17

[43] Garrett, D., Peterson, D.A., Anderson, C.W., Thaut, M.H.: Comparison of linear, nonlinear, and feature selection methods for eeg signal classification. IEEE Transactions on Neural Systems and Rehabilitation Engineering 11(2), 141–144 (June 2003) 12

[44] Gay, D., Guigourès, R., Boullé, M., Clérot, F.: Feature extraction over multiple representations for time series classification. In: International Workshop NFMCP held at ECML/PKDD. pp. 18–34 (2013) 137

[45] Ghahramani, Z., Jordan, M.I.: Supervised learning from incomplete data via an em approach. In: Advances in Neural Information Processing Systems 6. pp. 120–127. Morgan Kaufmann (1994) 38

[46] Ghalwash, M.F., Radosavljevic, V., Obradovic, Z.: Extraction of interpretable multivariate patterns for early diagnostics. In: IEEE 13th International Conference on Data Mining. pp. 201–210 (2013) 63

[47] Ghalwash, M.F., Ramljak, D., Obradovic, Z.: Early classification of multivariate time series using a hybrid hmm/svm model. In: IEEE International Conference on Bioinformatics and Biomedicine (BIBM). pp. 1–6 (2012) 63

[48] Ghalwash, M., Radosavljevic, V., Obradovic, Z.: Utilizing temporal patterns for estimating uncertainty in interpretable early decision making. In: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 402–411. ACM (2014) 45, 64

[49] Ghalwash, M.F., Obradovic, Z.: Early classification of multivariate temporal observations by extraction of interpretable shapelets. BMC Bioinformatics 13(1), 1–12 (2012) 63

[50] Goldin, D.Q., Millstein, T.D., Kutlu, A.: Bounded similarity querying for time-series data. Information and Computation 194(2), 203 – 241 (2004) 17

[51] Gonzalez, C., Diez, J., Boström, H.: Time series classification by boosting interval based literals. Intelligent Data Analysis 5(3), 245–262 (2001) 55

[52] Graham, J.W.: Missing data analysis: Making it work in the real world. Annual Review of Psychology 60(1), 549–576 (2009), pMID: 18652544 38

[53] Grefenstette, Ramsey, Ramsey, C.L.: An approach to anytime learning. In: Proceedings of the Ninth International Conference on Machine Learning (ICML). pp. 189–195. Morgan Kaufmann (1992) 27

[54] Griffin, M.P., Moorman, J.R.: Toward the early diagnosis of neonatal sepsis and sepsis-like illness using novel heart rate analysis. Pediatrics 107(1), 97–104 (2001) 2

[55] Hatami, N., Chira, C.: Classifiers with a reject option for early time-series classification. In: Computational Intelligence and Ensemble Learning (CIEL), 2013 IEEE Symposium on. pp. 9–16. IEEE (2013) 36, 60, 64

[56] He, G., Duan, Y., Qian, T., Chen, X.: Early prediction on imbalanced multivariate time series. In: The 22nd ACM international conference on Conference on information & knowledge management. pp. 1889–1892. CIKM '13, ACM (2013) 63

[57] Higuchi, T.: Approach to an irregular time series on the basis of the fractal theory. Phys. D 31(2), 277–283 (Jun 1988) 12

[58] Hotelling, H.: Analysis of a complex of statistical variables into principal components, (1933) 13

[59] Ishiguro, K., Sawada, H., Sakano, H.: Multi-class boosting for early classification of sequences. In: Proceedings of the British Machine Vision Conference. pp. 24.1–24.10. BMVA Press (2010) 63

[60] Keogh, E., Chu, S., Hart, D., Pazzani, M.: An enhanced representation of time series which allows fast and accurate classification, clustering and relevance feedback. Proceedings of the Fourth conference on Knowledge Discovery in Databases and Data Mining, New York pp. 239–241 (1998) 137

[61] Keogh, E., Chu, S., Hart, D., Pazzani, M.: Locally adaptive dimensionality reduction for indexing large time series databases. In: In Proceedings of ACM SIGMOD Conference on Management of Data. pp. 151–162 (May 2001) 13

[62] Keogh, E., Xi, X., Wei, L., Ratanamahatana, C.A.: The UCR Time Series Classification/Clustering homepage. www.cs.ucr.edu/~eamonn/time_series_data/ (2006) 103, 120, 121

[63] Keogh, E., Wei, L., Xi, X., Vlachos, M., Lee, S.H., Protopapas, P.: Supporting exact indexing of arbitrarily rotated shapes and periodic time series under euclidean and warping distance measures. The VLDB Journal 18(3), 611–630 (Jun 2009) 12

[64] Korn, F., Jagadish, H., Faloutsos, C.: Efficiently supporting ad hoc queries in large datasets of time sequences. In Proceedings of ACM SIGMOD International Conference on Management of Data pp. 289–300 (1997) 13

[65] Laxman, S., Sastry, P.S.: A survey of temporal data mining. Sadhana 31(2), 173–198 (2006) 12, 13

[66] Li, S., Wang, Z., Zhou, G., Lee, S.Y.M.: Semi-supervised learning for imbalanced sentiment classification. International Joint Conference on Artificial Intelligence (IJCAI) pp. 826–1831 (2011) 63

[67] Lin, J., Keogh, E., Lonardi, S., Chiu, B.: A Symbolic Representation of Time Series, with Implications for Streaming Algorithms. In: In 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery, San Diego. pp. 2–11 (2003) 13, 137

[68] Lin, Y.F., Chen, H.H., Tseng, V.S., Pei, J.: Advances in Knowledge Discovery and Data Mining: 19th Pacific-Asia Conference, PAKDD 2015, Proceedings, Part I, chap. Reliable Early Classification on Multivariate Time Series with Numerical and Categorical Attributes, pp. 199–211. Springer International Publishing, Cham (2015) 63

[69] Little, R.J.A., Rubin, D.B.: Statistical Analysis with Missing Data. John Wiley & Sons, Inc., New York, NY, USA (1986) 36

[70] Mauá, D.D., Antonucci, A., de Campos, C.P.: Hidden markov models with set-valued parameters. Neurocomputing 180, 94 – 107 (2016) 63

[71] Mitsa, T.: Temporal Data Mining. Chapman & Hall/CRC, 1st edn. (2010) 12

[72] Mori, A., Uchida, S., Kurazume, R., Taniguchi, R.: Early recognition and prediction of gestures. In: Proceedings of International Conference on Pattern Recognition. pp. 560–563. The MIT Press (2006) 36

[73] Myers, C., Rabiner, L., Rosenberg, A.: Performance tradeoffs in dynamic time warping algorithms for isolated word recognition. IEEE Transactions on Acoustics, Speech, and Signal Processing 28, 623–635 (1980) 15

[74] Needleman, S.B., Wunsch, C.D.: A general method applicable to the search for similarities in the amino acid sequence of two proteins. Journal of Molecular Biology 48, 443–453 (1970) 15

[75] Ng, A.Y., Jordan, M.I.: On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In: Advances in Neural Information Processing Systems, pp. 841–848. MIT Press (2002) 7

[76] Novikoff, A.B.: On convergence proofs on perceptrons. In: Proceedings of the Symposium on the Mathematical Theory of Automata. vol. 12, pp. 615–622. Polytechnic Institute of Brooklyn, New York, NY, USA (1962) 12

[77] Paramanathan, P., Uthayakumar, R.: Detecting patterns in irregular time series with fractal dimension. Proceedings of the International Conference on Computational Intelligence and Multimedia Applications pp. 323–237 (2007) 12

[78] Parrish, N., Anderson, H.S., Gupta, M.R., Hsiao, D.Y.: Classifying with confidence from incomplete information. The Journal of Machine Learning Research 14(1), 3561–3589 (2013) 39, 59, 64

[79] Pearson, K.: On lines and planes of closest fit to systems of points in space. Philosophical Magazine 2, 559–572 (1901) 13

[80] Pigott, T.D.: A review of methods for missing data. Educational Research and Evaluation 7(4), 353–383 (2001) 36

[81] Platt, J.C.: Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods. In: ADVANCES IN LARGE MARGIN CLASSIFIERS. vol. 10, pp. 61–74 (1999) 89

[82] Ramirez, J.C., Peterson, L.L., Cook, D.J., Peterson, D.M.: Discovery of temporal patterns in sparse course-of-disease data. IEEE Engineering in Medicine and Biology 19(4), 63–71 (2000) 12

[83] Rebbapragada, U., Protopapas, P., Brodley, C.E., Alcock, C.: Finding anomalous periodic time series. Machine Learning 74(3), 281–313 (2009) 12

[84] Rodríguez, J.J., Guez, J.J.R., Alonso, C.J.: Boosting interval-based literals: Variable length and early classification. Data Mining in Time Series Databases (2004) 51, 52, 53, 64

[85] Rosenblatt, F.: The perceptron: A probabilistic model for information storage and organization in the brain. Psychological Review pp. 65–386 (1958) 11

[86] Rousseeuw, P.J.: Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. Journal of computational and applied mathematics 20, 53–65 (1987) 80

[87] Rubin, D.B.: Multiple Imputation for Nonresponse in Surveys. Wiley & Sons (1987) 38

[88] Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Parallel distributed processing: Explorations in the microstructure of cognition, vol. 1. chap. Learning Internal Representations by Error Propagation, pp. 318–362. MIT Press, Cambridge, MA, USA (1986), http://dl.acm.org/citation.cfm?id=104279.104293 11

[89] Russell, S.J., Zilberstein, S.: Composing real-time systems. In: Proceedings of the 12th International Joint Conference on Artificial Intelligence - Volume 1. pp. 212–217. IJCAI'91, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1991) 27

[90] Saar-Tsechansky, M., Provost, F.: Handling missing values when applying classification models. Journal of machine learning research 8(Jul), 1623–1657 (2007) 38

[91] Salvador, S., Chan, P.: Toward accurate dynamic time warping in linear time and space. Intell. Data Anal. 11, 561–580 (2007) 15

[92] Sankoff, D., Kruskal, J.B.: Time Warps, String Edits and Macromolecules: The Theory and Practice of Sequence Comparison. Addison-Wesley Publishing Co, Reading, Massachusetts (1983) 15

[93] Shatkay, H., Zdonik, S.B.: Approximate Queries and Representations for Large Data Sequences. In: In Proc. of the 12th International Conference on Data Engineering (ICDE). pp. 536–545 (1996) 13

[94] Smith, T.F., Waterman, M.S.: Identification of common molecular subsequences. Journal of molecular biology 147, 195–197 (1981) 15

[95] Smola, A.J., Vishwanathan, S.V.N., Hofmann, T.: Kernel methods for missing variables. In: In Proceedings of the Tenth International Workshop on Artificial Intelligence and Sstatistics. pp. 325–332 (2005) 38

[96] Tavenard, R., Malinowski, S.: Cost-Aware Early Classification of Time Series. In: European Conference on Machine Learning and Principles and Practice of Knowledge Discovery. Riva del Garda, Italy (Sep 2016) 79

[97] Tsoumakas, G., Katakis, I.: Multi-label classification: An overview. Dept. of Informatics, Aristotle University of Thessaloniki, Greece (2006) 140

[98] Turney, P.D.: Types of cost in inductive concept learning. Workshop on Cost-Sensitive Learning at the Seventeenth International Conference on Machine Learning (WCSL at ICML-2000), Stanford University, California. (2000) 2

[99] Vapnik, V., Vashist, A.: A new learning paradigm: Learning using privileged information. Neural Networks 22(5), 544–557 (2009) 29

[100] Viola, P., Jones, M.J.: Robust real-time face detection. Int. J. Comput. Vision 57(2), 137–154 (May 2004) 69

[101] Vlachos, M., Kollios, G., Gunopulos, D.: Discovering Similar Multidimensional Trajectories. In: IEEE International Conference on Data Engineering (ICDE). pp. 673–684 (2002) 15

[102] Wald, A., Wolfowitz, J.: Optimum character of the sequential probability ratio test. The Annals of Mathematical Statistics pp. 326–339 (1948) 69

[103] Wang, X., Mueen, A., Ding, H., Trajcevski, G., Scheuermann, P., Keogh, E.: Experimental comparison of representation methods and distance measures for time series data. Data Mininig Knowledge Discovery 26(2), 275–309 (Mar 2013) 13

[104] Wasito, I., Mirkin, B.: Nearest neighbour approach in the least-squares data imputation algorithms. Information Sciences 169(1), 1–25 (2005) 38

[105] Wolpert, D.H., Macready, W.G.: No free lunch theorems for optimization. IEEE Transactions on Evolutionary Computation 1(1), 67–82 (April 1997) 32

[106] Xing, Z., Pei, J., Dong, G., Philip, S.: Mining sequence classifiers for early prediction. In: SDM. pp. 644–655. SIAM (2008) 23, 55, 57, 64

[107] Xing, Z., Pei, J., Philip, S.: Early prediction on time series: A nearest neighbor approach. In: IJCAI. pp. 1297–1302. Citeseer (2009) 34, 41, 57, 64

[108] Xing, Z., Pei, J., Philip, S., Wang, K.: Extracting interpretable features for early classification on time series. In: SDM. vol. 11, pp. 247–258. SIAM (2011) 45, 58, 62, 64

[109] Ye, L., Keogh, E.: Time series shapelets: a new primitive for data mining. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2009) 45

[110] Ye, L., Keogh, E.: Time series shapelets: a new primitive for data mining. In: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 947–956. ACM (2009) 45

[111] Yi, B., Faloutsos, C.: Fast time sequence indexing for arbitary Lp norms. 26th Intl. Conference On very large databases, Cairo pp. 285–394 (2000) 13

[112] Zaffalon, M., Fagiuoli, E.: Tree-based credal networks for classification. Reliable Computing 9(6), 487–509 (2003) 64

[113] Zliobaite, I.: Learning under concept drift: an overview. CoRR (2010) 27

université PARIS-SACLAY
ÉCOLE DOCTORALE
Agriculture, alimentation, biologie, environnement, santé (ABIES)

Titre : Classification précoce de séries temporelles lorsque reporter la décision est coûteux

Mots clefs : Classification précoce, séries temporelles, décision adaptative et non myope, coût d'attente de décision

Résumé : Dans de nombreux domaines dans lesquels les mesures ou les données sont disponibles séquentiellement, il est important de savoir décider le plus tôt possible, même si c'est à partir d'informations encore incomplètes. C'est le cas par exemple en milieu hospitalier où l'apprentissage de règles de décision peut se faire à partir de cas complètement documentés, mais où, devant un nouveau patient, il peut être crucial de prendre une décision très rapidement. Dans ce type de contextes, un compromis doit être optimisé entre la possibilité d'arriver à une meilleure décision en attendant des mesures supplémentaires, et le coût croissant associé à chaque nouvelle mesure.
Nous considérons dans cette thèse un nouveau cadre général de classification précoce de séries temporelles où le coût d'attente avant de prendre une décision est explicitement pris en compte lors de l'optimisation du compromis entre la qualité et la précocité de prédictions. Nous proposons donc un critère formel qui exprime ce compromis, ainsi que deux approches différentes pour le résoudre. Ces approches sont intéressantes et apportent deux propriétés désirables pour décider en ligne : (i) elles estiment en ligne l'instant optimal dans le futur où une minimisation du critère peut être prévue. Elles vont donc au-delà des approches classiques qui décident d'une façon myope, à chaque instant, d'émettre une prédiction ou d'attendre plus d'information, (ii) ces approches sont adaptatives car elles prennent en compte les propriétés de la série temporelle en entrée pour estimer l'instant optimal pour la classifier.
Des expériences extensives sur des données contrôlées et sur des données réelles montrent l'intérêt de ces approches pour fournir des prédictions précoces, fiables, adaptatives et non myopes, ce qui est indispensable dans de nombreuses applications.

Title: Cost-Sensitive Early Classification of Time Series

Keywords: Early classification, time series, adaptive and non-myopic decision making, costly delaying decision

Abstract: Early classification of time series is becoming increasingly a valuable task for assisting in decision making process in many application domains.
In this setting, information can be gained by waiting for more evidences to arrive, thus helping to make better decisions that incur lower misclassification costs, but, meanwhile, the cost associated with delaying the decision generally increases, rendering the decision less attractive. Making early predictions provided that are accurate requires then to solve an optimization problem combining two types of competing costs.
This thesis introduces a new general framework for time series early classification problem. Unlike classical approaches that implicitly assume that misclassification errors are cost equally and the cost of delaying the decision is constant over time, we cast the the problem as a cost-sensitive online decision making problem when delaying the decision is costly. We then propose a new formal criterion, along with two approaches that estimate the optimal decision time for a new incoming yet incomplete time series. In particular, they capture the evolutions of typical complete time series in the training set thanks to a segmentation technique that forms meaningful groups, and leverage these complete information to estimate the costs for all future time steps where data points still missing.
These approaches are interesting in two ways: (i) they estimate, online, the earliest time in the future where a minimization of the criterion can be expected. They thus go beyond the classical approaches that myopically decide at each time step whether to make a decision or to postpone the call one more time step, and (ii) they are adaptive, in that the properties of the incoming time series are taken into account to decide when is the optimal time to output a prediction.
Results of extensive experiments on synthetic and real data sets show that both approaches successfully meet the behaviors expected from early classification systems.

Université Paris-Saclay
Espace Technologique / Immeuble Discovery
Route de l'Orme aux Merisiers RD 128 / 91190 Saint-Aubin, France