Early Classification of Time Series as a Non Myopic Sequential Decision Making Problem

Asma Dachraoui[1,2], Alexis Bondu[1], and Antoine Cornuéjols[2](✉)

[1] EDF R & D, 1, Avenue du Général de Gaulle, 92141 Clamart Cedex, France
[2] Département MMIP Et INRA UMR-518, AgroParisTech, 16, Rue Claude Bernard,
75231 Paris Cedex 5, France
antoine.cornuejols@agroparistech.fr
http://www.agroparistech.fr/mia/equipes:membres:page:antoine

Abstract. Classification of time series as early as possible is a valuable goal. Indeed, in many application domains, the earliest the decision, the more rewarding it can be. Yet, often, gathering more information allows one to get a better decision. The optimization of this time vs. accuracy tradeoff must generally be solved online and is a complex problem.

This paper presents a formal criterion that expresses this trade-off in all generality together with a generic sequential meta algorithm to solve it. This meta algorithm is interesting in two ways. First, it pinpoints where choices can (*have to*) be made to obtain a computable algorithm. As a result a wealth of algorithmic solutions can be found. Second, it seeks online the earliest time in the future where a minimization of the criterion can be expected. It thus goes 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.

After this general setting has been expounded, we study one simple declination of the meta-algorithm, and we show the results obtained on synthetic and real time series data sets chosen for their ability to test the robustness and properties of the technique. The general approach is vindicated by the experimental results, which allows us to point to promising perspectives.

Keywords: Early classification of time series · Sequential decision making

1 Introduction

In many applications, it is natural to acquire the description of an object incrementally, with new measurements arriving sequentially. This is the case in medicine, when a patient undergoes 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 measurements are not controlled and just arrive over time, as when the behavior of a consumer on a web site is monitored on-line in order to predict what add to put on his/her screen.

A. Appice et al. (Eds.): ECML PKDD 2015, Part I, LNAI 9284, pp. 433–447, 2015.
DOI: 10.1007/978-3-319-23528-8_27

In these situations, one is interested in making a prediction as soon as possible, either because each measurement is costly or because it is critical to act as early as possible in order to yield higher returns. However, this generally induces a tradeoff as less measurements commonly entail more prediction errors that can be expensive. The question is therefore how to decide on-line that now is the optimal time to make a prediction.

The problem of deciding when enough information has been gathered to make a reliable decision has historically been studied under the name of *sequential decision making* or *Optimal statistical decisions* [1,2]. One foremost technique being Wald's Sequential Probability Ratio Test [3] which applies to two-classes classification problems and uses the likelihood ratio:

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

where $\langle x_1^i, \ldots, x_t^i \rangle$ is the sequence of t measurements so far that must be classified to either class -1 or class $+1$. As the number of measurements t increases, this ratio is compared to 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*). The main difficulty lies in the estimation of the conditional probabilities $P(\langle x_1^i, \ldots, x_t^i \rangle \mid y)$. (See also [4], a modern implantation of this idea).

A prominent limitation of this general approach is that the cost of delaying the decision is not taken into account. More recent techniques include the two components of the cost of early classification problems: the cost associated with *the quality* of the prediction and the cost of *the delay* before a prediction is made about the incoming sequence. However, most of them compute an optimal decision time from the learning set, which is then applied to any incoming example whatever their characteristics are. The decision is therefore not adaptive since the delay before making a prediction is independent on the input sequence.

The originality of the method presented here is threefold. *First*, the problem of early classification of time series is formalized as a sequential decision problem involving the two costs: quality and delay of the prediction. *Second*, the method is adaptive, in that the properties of the incoming sequence 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 algorithm presented is not myopic. At each time step, it computes 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. It would never try to estimate in advance the best time to make the prediction. The capacity of conjecturing when in the future an optimal prediction should be made with regard to the quality and delay of the prediction is however important and offers valuable opportunities compared to myopic sequential decisions. Indeed, 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.

The paper is organized as follows. We first review some related work in Section (2). The formal statement of the early classification problem (Section (3)) leads to a generic sequential decision making meta algorithm. Our early decision making proposed approach and its optimal decision rule are formalized in Section (4). In Section (5), we propose one simple implementation of this meta algorithm to illustrate our approach. Experiments and results on synthetic data as well as on real data are presented and discussed in Section (6). The conclusion, in Section (7), underlines the promising features of the approach presented and discusses future works.

2 A Generic Framework and Positions of Related Works

In the following, we will assume that we have a set $\mathcal{S}$ of m training sequences with each training sequence being a couple $(\mathbf{x}_T^i, y_i) \in \mathbb{R}^T \times \mathcal{Y}$, meaning that it is composed of T real valued measurements $\langle x_1^i, \ldots, x_T^i \rangle$, and an associated label $y_i \in \mathcal{Y}$, where $\mathcal{Y}$ is a finite set of classes. The question is to choose the earliest time t^* at which a new incoming and still incomplete sequence $\mathbf{x}_{t^*} = \langle x_1, x_2, \ldots, x_{t^*} \rangle$ can be optimally labeled. Algorithm (1) provides a generic description of early classification methods.

Algorithm 1. Framework of early classification methods

Input:

- $\mathbf{x}_t \in \mathbb{R}^t$, $t \in \{1, \ldots, T\}$, an incoming time series;
- $\{h_t\}_{t \in \{1 \ldots, T\}} : \mathbb{R}^t \longrightarrow \mathcal{Y}$, a set of predictive functions h_t learned from the training set;
- $x_t \in \mathbb{R}$, a new incoming real measurement;
- $\mathcal{T}rigger : \mathbb{R}^t \times h_t \longrightarrow \mathcal{B}, t \in \{1, \ldots, T\}, \mathcal{B} \in \{true, false\}$, 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
```

In the framework outlined above, we suppose that the training set $\mathcal{S}$ has been used in order to learn a series of hypotheses $h_t (t \in \{1, \ldots, T\})$, each hypothesis h_t being able to classify examples of length t: $\mathbf{x}_t = \langle x_1, x_2, \ldots, x_t \rangle$.

Then, the various existing methods for early classification of time series can be categorized according to the $\mathcal{T}rigger$ function which decides when to stop measuring additional information and output a prediction $h_t(\mathbf{x}_t)$ for the class of $\mathbf{x}_t$.

Several papers that are openly motivated by the problem of early classification turn out indeed to be concerned with the problem of classifying from incomplete sequences rather than with the problem of optimizing a tradeoff between the precision of the prediction and the time it is performed. (see for instance [5] where clever classification schemes are presented, but no explicit cost for delaying the decision is taken into account). Therefore there is stricto sensu no $\mathcal{T}rigger$ function used in these algorithms.

In [6], the $\mathcal{T}rigger$ function relies on an estimate of the earliest time at which the prediction $h_t(\mathbf{x}_t)$ should be equal to the one that would be made if the complete example $\mathbf{x}_T$ was known: $h_T(\mathbf{x}_T)$. The so-called *minimum prediction length* (MPL) is introduced, and is estimated using a one nearest neighbor classifier.

In a related work [7,8], the $\mathcal{T}rigger$ function is based on a very similar idea. It outputs *true* when the probability that the assigned label $h_t(\mathbf{x}_t)$ will match the one that would be assigned using the complete time series $h_T(\mathbf{x}_T)$ exceeds some given threshold. To do so, the authors developed a *quadratic discriminant analysis* that estimates a reliability bound on the classifier's prediction at each time step.

In [9], the $\mathcal{T}rigger$ function outputs *true* if the classification function h_t has a sufficient confidence in its prediction. In order to estimate this confidence level, the authors use an ensemble method whereby the level of agreement is translated into a confidence level.

In [10], an early classification approach relying on uncertainty estimations is presented. It extends the *early distinctive shapelet classification* (EDGC) [11] method to provide an uncertainty estimation for each class at each time step. Thus, an incoming time series 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.

It is remarkable that even if the earliness of the decision is mentioned as a motivation in these papers, 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.

In this paper, we present a method that explicitly balance the expected gain in the precision of the decision *at all future time steps* with the cost of delaying the decision. In that way, the optimizing criterion is explicitly a function of both aspects of the early decision problem, and, furthermore, it allows one to estimate, and update if necessary, the future optimal time step for the decision.

3 A Formal Analysis and a Naïve Approach

The question is to learn a decision procedure in order to determine the earliest time t^* at which a new incoming sequence $\mathbf{x}_{t^*} = \langle x_1, x_2, \ldots, x_{t^*} \rangle$ 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* $C_t(\hat{y}|y) : \mathcal{Y} \times \mathcal{Y} \longrightarrow \mathbb{R}$ 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* $C(t)$ 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.

$$f(\mathbf{x}_t) \;=\; \sum_{y \in \mathcal{Y}} P(y|\mathbf{x}_t) \sum_{\hat{y} \in \mathcal{Y}} P(\hat{y}|y, \mathbf{x}_t)\, C_t(\hat{y}|y) \;+\; C(t) \tag{1}$$

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 :

$$t^* \;=\; \underset{t \in \{1,\ldots,T\}}{\text{ArgMin}}\; f(\mathbf{x}_t) \tag{2}$$

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 sequence* (hence the function now denoted $f(t)$):

$$f(t) \;=\; \sum_{y \in \mathcal{Y}} P(y) \sum_{\hat{y} \in \mathcal{Y}} P(\hat{y}|y)\, C_t(\hat{y}|y) \;+\; C(t) \tag{3}$$

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 that the confusion matrix associated with the considered classifier. One gets then the optimal time for prediction as:

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

This can be computed before any new incoming sequence, and, indeed, t^* is independent on the input sequence. Of course, this is intuitively unsatisfactory as one could feel, regarding a new sequence, 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^*. This is why we propose an adaptive approach.

4 The Proposed Approach

The goal is to estimate the conditional probability $P(\hat{y}|y, \mathbf{x}_t)$ in Equation (1) by taking into account the incoming time series $\mathbf{x}_t$ in order to determine the optimal time t^*. There are several possibilities for this.

In this paper, the idea is to identify a set $\mathcal{C}$ of K clusters $\mathfrak{c}_k$ ($k \in \{1, \ldots, K\}$) of complete sequences using a training set so that, later, an (incomplete) input sequence $\mathbf{x}_t = \langle x_1, \ldots, x_t \rangle$ can have a membership probability assigned to each of these clusters: $P(\mathfrak{c}_k \mid \mathbf{x}_t)$, and therefore will be recognized as more or less close to each of the prototype sequences corresponding to the clusters. A complete explanation is given below in Section 5.

The set $\mathcal{C}$ of clusters should obey two constraints as well as possible.

1. Different clusters should correspond to different confusion matrices. Otherwise, Equation (1) will not be able to discriminate the cost between clusters.
2. Clusters should contain similar time series, and be dissimilar to other clusters, so that an incoming sequence will generally be assigned markedly to one of the clusters.

For each time step $t \in [1, \ldots, T]$, a classifier h_t is trained using a learning set $\mathcal{S}'$. One can then estimate the associated confusion matrix for each cluster and classifier h_t: $\mathfrak{c}_k$: $P_t(\hat{y}|y, \mathfrak{c}_k)$ over a distinct learning set $\mathcal{S}$".

When a new input sequence $\mathbf{x}_t$ of length t is considered, it is compared to each cluster $\mathfrak{c}_k$ (of complete time series) and is given a probability membership $P_t(\hat{y}|y, \mathfrak{c}_k)$ for each of them (as detailed in Section (5)). In a way, this compares the input sequence to all families of its possible continuations.

Given that, at time t, $T - t$ measurements are still missing on the incoming sequence, it is possible to compute the expected cost of classifying $\mathbf{x}_t$ at each future time step $\tau \in \{0, \ldots, T - t\}$:

$$f_\tau(\mathbf{x}_t) \;=\; \sum_{\mathfrak{c}_k \in \mathcal{C}} P(\mathfrak{c}_k|\mathbf{x}_t) \sum_{y \in \mathcal{Y}} \sum_{\hat{y} \in \mathcal{Y}} P_{t+\tau}(\hat{y}|y, \mathfrak{c}_k)\, C(\hat{y}|y) + C(t + \tau) \tag{4}$$

Perhaps not apparent at first, this equation expresses two remarkable properties.

First, it is computable, which was not the case of Equation (1). Indeed, each of the terms $P(\mathfrak{c}_k|\mathbf{x}_t)$ and $P_{t+\tau}(\hat{y}|y, \mathfrak{c}_k)$ can now be estimated through frequencies observed in the training data (see Figure (1)). Second, the cost depends on the incoming sequence because of the use of the probability memberships $P(\mathfrak{c}_k|\mathbf{x}_t)$. It is therefore not computed beforehand, once for all.

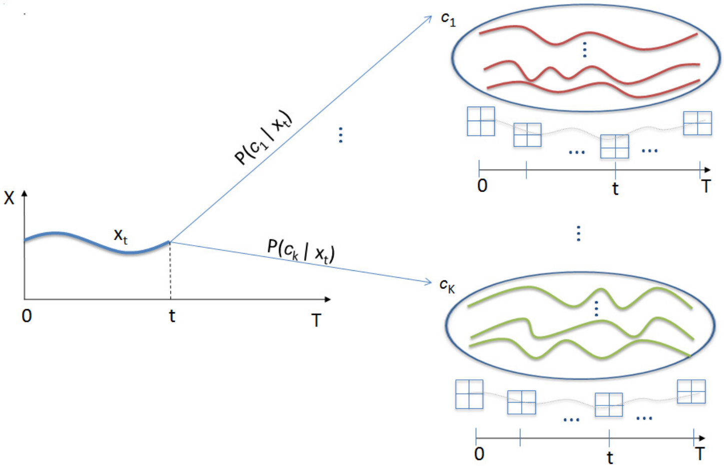

Fig. 1. An incoming (incomplete) sequence is compared to each cluster $\mathfrak{c}_k$ obtained from the training set of complete time series. The confusion matrices for each time step t and each cluster $\mathfrak{c}_k$ are computed as explained in the text.

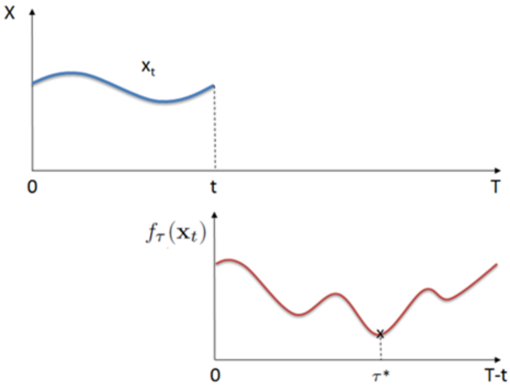

Fig. 2. The first curve represents an incoming time series $\mathbf{x}_t$. The second curve represents the expected cost $f_\tau(\mathbf{x}_t)$ given $\mathbf{x}_t$, $\forall \tau \in \{0, \ldots, T - t\}$. It shows the balance between the gain in the expected precision of the prediction and the cost of waiting before deciding. The minimum of this tradeoff is expected to occur at time $\tau^\star$. New measurements can modify the curve of the expected cost and the estimated $\tau^\star$.

In addition, the fact that the expected cost $f_\tau(\mathbf{x}_t)$ can be computed for each of the remaining τ time steps allows one to forecast what should be the optimal horizon $\tau^\star$ for the classification of the input sequence (see Figure (2)):

$$\tau^* = \underset{\tau \in \{0,\ldots,T-t\}}{\text{ArgMin}} \; f_\tau(\mathbf{x}_t) \tag{5}$$

Of course, these costs, and the expected optimal horizon $\tau^\star$, can be re-evaluated when a new measurement is made on the incoming sequence. At any time step t, if the optimal horizon $\tau^\star = 0$, then the sequential decision process stops and a prediction is made about the class of the input sequence $\mathbf{x}_t$ using the classifier h_t^k:

$$\hat{y} \;=\; h_t^k(\mathbf{x}_t), \quad \text{where } k = \underset{k \in \{1,\ldots,K\}}{\text{ArgMax}} \; P(\mathfrak{c}_k|\mathbf{x}_t)$$

Returning to the general framework outlined for the early classification problem in Section (3), the proposed function that triggers a prediction for the incoming sequence is given in Algorithm (2):

Algorithm 2. Proposed $\mathcal{T}rigger(\mathbf{x}_t, h_t)$ function.

```
Input:  $\mathbf{x}_t$, $t \in \{1, \ldots, T\}$, an incoming time series;
$\mathcal{T}rigger \longleftarrow false$
for all $\tau \in \{0, \ldots, T-t\}$ do
    compute $f_\tau(\mathbf{x}_t)$                         /* see Equation (4)*/
end for
$\tau^* = \underset{\tau \in \{0,\ldots,T-t\}}{\text{ArgMin}} \; f_\tau(\mathbf{x}_t)$
if $(\tau^* = 0)$ then
    $\mathcal{T}rigger \longleftarrow true$
end if
```

5 Implementation

Section (3) has outlined the general framework for the early classification problem while Section (4) has presented our proposed approach where the problem is cast as a sequential decision problem with three properties: (i) both the quality of the prediction and the delay before prediction are taken into account in the total criterion to be optimized, (ii) the criterion is adaptive in that it depends upon the incoming sequence $\mathbf{x}_t$, and (iii) the proposed solution leads to a non myopic scheme where the system forecasts the expected optimal horizon τ^* instead of just deciding that now is, or is not, the time to make a prediction.

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

1. The type of *classifiers* used. For each time step $t \in \{1, \ldots, T\}$, the input dimension of the classifier is t.
2. The *clustering method*, which includes the technique (e.g. k-means), the distance used (e.g. the euclidean distance, the time warping distance, ...), 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 paper, 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) for the classifiers, we have used Naïve Bayes classifiers and Multi-layer Perceptrons with one hidden layer of $\lfloor t + 2/2 \rfloor$ neurons. In Section (6), we only show results obtained using the Multi-Layer Perceptron since both classifiers give similar results. (2) The clustering over complete time series is performed using k-means with euclidean distance. The number K_y of clusters for each of the target classes $y = -1$ and $y = +1$ corresponds to the maximum *silhouettes* factor [12]. (3) The membership probabilities $P(\mathfrak{c}_k|\mathbf{x}_t)$ are computed using the following equation:

$$P(\mathfrak{c}_k|\mathbf{x}_t) \; = \; \frac{s_k}{\sum_i^K s_i}, \quad \text{where } s_k \; = \; \frac{1}{1 + \exp^{-\lambda \Delta_k}} \tag{6}$$

The constant λ used in the sigmoid function s_k is empirically learned from the training set, while $\Delta_k \; = \; \overline{D} - d_k$ is the 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 Euclidian distance between the first t components of the two series.

6 Experiments

Our experiments aimed at checking the validity of the proposed method and at exploring its capacities for various conditions. To this end, we devised controlled experiments with artificial data sets for which we could vary the control parameters: difference between the two target classes, noise level, number of different time series shapes in each class and the cost of waiting before decision $C(t)$. We also applied the method to the real data set TwoLeadECG from UCR Time Series Classification/Clustering repository [13].

6.1 Controlled Experiments

We devised our experiments so that there should be a gain, that we can control, in the prediction accuracy if more measurements are made (increasing t). We have also devised the target classes so that they are composed of several families of time sequences, with, possibly, families that share a strong resemblance between different target classes.

In the reported experiments, the time series in the training set and the testing set are generated according to the following equations:

$$\mathbf{x}_t \; = \; a \, \sin(\omega_i \, t + \mathit{phase}) + b \, t \; + \; \varepsilon(t) \tag{7}$$

The constant b is used to set a general trend, for instance either ascending ($b > 0$) or descending ($b < 0$), while the first term $a \, \sin(\omega_i \, t + \mathit{phase})$ provides

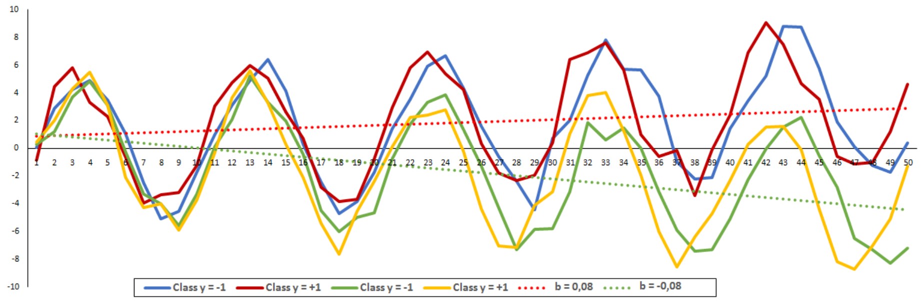

Fig. 3. Subgroups of sequences generated for classes $y = +1$ and $y = -1$, when the trend parameter $b = -0.08$ or $b = +0.08$, and the noise level $\varepsilon(t) = 0.5$.

a shape for this particular family of time series. The last term is a noise factor that makes the overall prediction task more or less difficult.

For instance, Figure (3) shows a set of time series (one for each shape) where:

- $b = -0.08$ or $b = +0.08$
- $a = 5$ and $phase = 0$
- $\omega_1 = 10$ or $\omega_2 = 10.3$ (here, there are 2 groups of time sequences per class)
- $\varepsilon(t)$ is a gaussian term of mean $= 0$ and standard deviation $= 0.5$
- $T = 50$

In this particular setting, it is apparent that it is easy to mix up the two classes $y = -1$ and $y = +1$ until intermediate values of t. However, the waiting cost $C(t)$ may force the system to make a decision before there is enough measurements to make a reasonably sure guess on the class y.

In our experiments, the training set $\mathcal{S}$ contained 2,500 examples, and the testing set $\mathcal{T}$ contained 1000 examples, equally divided into the two classes $y = -1$ and $y = +1$. (Nota: In case of imbalanced classes, it is easy to compensate this by modifying the misclassification cost function $C_t(\hat{y}|y)$). Each class was made of several subgroups: K_{-1} ones for class -1 and K_{+1} ones for class $+1$. The misclassification costs were set as: $C(\hat{y}|y) = 1, \ \forall \ \hat{y}, y$, and the *time cost function* $C(t) = d \times t$, where $d \in \{0.01, 0.05, 0.1\}$.

We varied:

- The level of *distinction between the classes* controlled by b
- The *number of subgroups* in each class and their shape (given by the term $a \, \sin(\omega_i \, t + phase)$)
- The *noise* level $\varepsilon(t)$
- The *cost of waiting* before decision $C(t)$

The results for various combinations of these parameters are shown in Table (1) as obtained on the time series of the testing set. It reports $\overline{\tau}^{\star}$, the average of computed optimal times of decision and its associated standard deviation $\sigma(\tau^{\star})$.

Table 1. Experimental results in function of the waiting cost $C(t) = \{0.01, 0.05, 0.1\} \times t$, the noise level $\varepsilon(t)$ and the trend parameter b.

$C(t)$	$\pm b$		0.02			0.05			0.07	
	$\varepsilon(t)$	$\overline{\tau}^\star$	$\sigma(\tau^\star)$	AUC	$\overline{\tau}^\star$	$\sigma(\tau^\star)$	AUC	$\overline{\tau}^\star$	$\sigma(\tau^\star)$	AUC
0.01	0.2	9.0	2.40	0.99	9.0	2.40	0.99	10.0	0.0	1.00
	0.5	13.0	4.40	0.98	13.0	4.40	0.98	15.0	0.18	1.00
	1.5	24.0	10.02	0.98	32.0	2.56	1.00	30.0	12.79	0.99
	5.0	26.0	7.78	0.84	30.0	18.91	0.87	30.0	19.14	0.88
	10.0	38.0	18.89	0.70	48.0	1.79	0.74	46.0	5.27	0.75
	15.0	23.0	15.88	0.61	32.0	13.88	0.64	29.0	17.80	0.62
	20.0	7.0	8.99	0.52	11.0	11.38	0.55	4.0	1.22	0.52
0.05	0.2	8.0	2.00	0.98	8.0	2.00	0.98	9.0	0.0	1.00
	0.5	10.0	2.80	0.96	8.0	4.0	0.98	14.0	0.41	0.99
	1.5	5.0	0.40	0.68	20.0	0.42	0.95	14.0	4.80	0.88
	5.0	8.0	3.87	0.68	6.0	1.36	0.64	5.0	0.50	0.65
	10.0	4.0	0.29	0.56	4.0	0.25	0.56	4.0	0.34	0.57
	15.0	4.0	0.0	0.54	4.0	0.25	0.56	4.0	0.0	0.55
	20.0	4.0	0.0	0.52	4.0	0.0	0.52	4.0	0.0	0.52
0.10	0.2	6.0	0.80	0.95	7.0	1.60	0.94	8.0	0.40	0.96
	0.5	6.0	0.80	0.84	9.0	2.40	0.93	10.0	0.0	0.95
	1.5	4.0	0.0	0.67	5.0	0.43	0.68	6.0	0.80	0.74
	5.0	4.0	0.07	0.64	4.0	0.05	0.64	4.0	0.11	0.64
	10.0	4.0	0.0	0.56	48.0	1.79	0.74	4.0	0.22	0.56
	15.0	4.0	0.0	0.55	4.0	0.0	0.55	4.0	0.0	0.55
	20.0	4.0	0.0	0.52	11.0	11.38	0.55	4.0	0.0	0.52

Additionally, the Area Under the ROC Curve AUC evaluates the quality of the prediction at the optimal decision time $\tau^\star$ computed by the system.

Globally, one can see that when the noise level is low ($\varepsilon \leq 1.5$) and the waiting cost is low too ($C(t) = c_t \times t$, with $c_t \leq 0.05$), the system is able to reach a high level of performance by waiting increasingly as the noise level augments. When the waiting cost is high ($C(t) = 0.1 \times t$), on the other hand, the system takes a decision earlier at the cost of a somewhat lower prediction performance. Indeed, with rising levels of noise, the system decides that it is not worth waiting and makes a prediction early on, often at the earliest possible moment, which was set to 4 in our experiments[1].

More specifically:

- **Impact of the noise level** $\varepsilon(t)$: As expected, up to a certain value, rising levels of noise $\varepsilon(t)$ entails increasing delays before a decision is decided upon by the system. Then, a decrease of $\overline{\tau}^\star$ is observed, which corresponds to the

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

fact that there is no gain to be expected by waiting further. Accordingly, the performance, as measured with the AUC, decreases as well when $\varepsilon(t)$ rises.

- **Impact of the waiting cost** $C(t)$: The role of the waiting cost $C(t)$ appears clearly. When $C(t)$ is very low, the algorithm tends to wait longer before making a decision, often waiting the last possible time. On the other hand, with rising $C(t)$, the optimal decision time $\overline{\tau}^{\star}$ decreases sharply, converging to the minimal possible value of 4.
- **Impact of the trend parameter** b: While the value of b, which controls the level of distinction of the classes $y = +1$ and $y = -1$, is not striking on the average time of decision $\overline{\tau}^{\star}$, one can notice however the decrease of the standard deviation when b increases from $b = 0.02$ to $b = 0.05$. At the same time, the AUC increases as well. For small values of the noise level, the decrease of the standard deviation is further observed when $b = 0.07$.
- **Impact of the number of subgroups 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 (7). For instance, Table (2) reports the results obtained when the number of subgroups of class $y = -1$ was set to $K_{-1} = 3$ while it was set to $K_{+1} = 5$ for class $y = +1$.
  When the waiting cost is very low ($C(t) = 0.01$), the number of subgroups in each class, and hence the complexity of the classes, does not influence the results. However, when the waiting cost increases ($C(t) = 0.05 \times t$), the decision task becomes harder, and the decision time increases while the AUC decreases.

The above results, in Table (1) and Table (2), aggregate the measures on the whole testing set. It is interesting to look as well at individual behaviors. For instance, Figure (4) shows the expected costs $f_\tau(\mathbf{x}_t^1)$ and $f_\tau(\mathbf{x}_t^2)$ for two different incoming sequences $\mathbf{x}_t^1$ and $\mathbf{x}_t^2$, for each of the potentially remaining τ time steps. First, one can notice the overall shape of the cost function $f_\tau(\mathbf{x}_t)$ with a decrease followed by a rise. Second, the *dependence on the incoming sequence* appears clearly, with different optimal times $t^{\star}$. This confirms that the algorithm takes into account the peculiarities of the incoming sequence.

6.2 Experiments on a Real Data Set

In order to test the ability of the method to solve real problems, we have realized experiments using the real data set TwoLeadECG from the UCR repository. This data set contains 1162 ECG signals all together, that we randomly and disjointedly re-sampled and split into a training set of 70% of examples and the remainder for the test set. Each signal is composed of 81 data point representing the electrical activity of the heart from two different leads. The goal is to detect an abnormal activity in the heart. Our experiments show that it is indeed possible to make an informed decision before all measurements are made.

Table 2. Experimental results in function of the noise level $\varepsilon(t)$, the trend parameter b, and the number of subgroups k_{+1} and k_{-1} in each class. The waiting cost $C(t)$ is fixed to 0.01.

(K_{-1}, K_{+1})	$\pm b$	0.02			0.05			0.07		
	$\varepsilon(t)$	$\overline{\tau}^\star$	$\sigma(\tau^\star)$	AUC	$\overline{\tau}^\star$	$\sigma(\tau^\star)$	AUC	$\overline{\tau}^\star$	$\sigma(\tau^\star)$	AUC
(3,2)	0.2	9.0	2.40	0.99	9.0	2.40	0.99	10.0	0.0	1.00
	0.5	13.0	4.40	0.98	13.0	4.40	0.98	15.0	0.18	1.00
	1.5	24.0	10.02	0.98	32.0	2.56	1.00	30.0	12.79	1.00
	5.0	26.0	7.78	0.84	30.0	18.90	0.87	30.0	19.14	0.88
	10.0	38.0	18.89	0.70	48.0	1.79	0.74	46.0	5.27	0.75
	15.0	23.0	15.88	0.61	32.0	13.88	0.64	29.0	17.80	0.62
	20.0	7.0	8.99	0.52	11.0	11.38	0.55	4.0	1.22	0.52
(3,5)	0.2	7.0	2.47	0.86	7.0	2.15	0.89	7.0	3.00	0.85
	0.5	11.0	5.10	0.87	10.0	4.87	0.88	14.0	7.07	0.91
	1.5	20.0	12.69	0.85	18.0	11.80	0.87	26.0	16.33	0.89
	5.0	44.0	4.75	0.83	46.0	2.81	0.87	38.0	11.49	0.81
	10.0	42.0	6.34	0.67	39.0	7.59	0.68	25.0	8.57	0.61
	15.0	28.0	5.99	0.58	32.0	6.51	0.59	19.0	10.12	0.58
	20.0	17.0	11.72	0.50	13.0	10.72	0.56	17.0	5.93	0.55

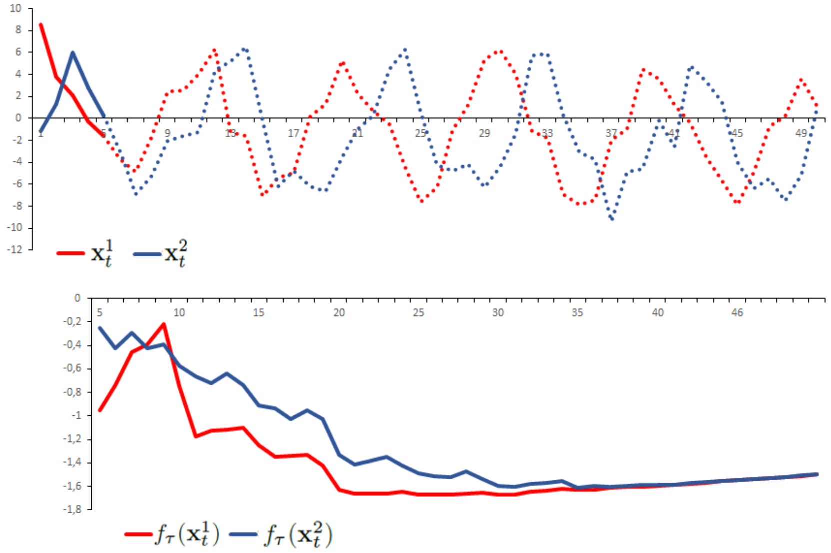

Fig. 4. For two different incoming sequences (top figure), the expected costs (bottom figure) are different. The minima have different values and occur at different instants. These differences confirm that deciding to make a prediction depends on the incoming sequence. (Here, $b = 0.05$, $C(t) = 0.01 \times t$ and $\varepsilon = 1.5$).

Table 3. Experimental results on real data in function of the waiting cost $C(t)$.

$C(t)$	0.01	0.05	0.1
$\overline{\tau}^{\star}$	22.0	24.0	10.0
$\sigma(\tau^{\star})$	6.1214	15.7063	9.7506
AUC	0.9895	0.9918	0.9061

Since the costs involving quality and delay of decision are not provided with this data set, we arbitrarily set these costs to $C(\hat{y}|y) = 1, \ \forall\, \hat{y}, y$, and $C(t) = d \times t$, where $d \in \{0.01, 0.05, 0.1\}$. The question here is whether the method is able to make reliable prediction early and provide reasonable results.

Table (3) reports the average of optimal times of decision $\overline{\tau}^{\star}$ of test time series, its associated standard deviation $\sigma(\tau^{\star})$, and the performance of the prediction AUC. It is remarkable that *a very good performance*, as measured by the AUC, *can be obtained from a limited set of measurements*: E.g. 22 out of 81 if $C(t) = 0.01$, 24 out of 81 if $C(t) = 0.05$, and 10 out of 81 if $C(t) = 0.1$.

We therefore see that the baseline solution proposed here is able to (1) adapt to each incoming sequence and (2) to predict an estimated optimal time of prediction that yields very good prediction performances while controlling the cost of delay.

7 Conclusion and Future Works

The problem of online decision making has been known for decades, but numerous new applications in medicine, electric grid management, automatic transportation, and so on, give a new impetus to research works in this area. In this paper, we have formalized a generic framework for early classification methods that underlines two critical parts: (i) the optimization criterion that governs the $\mathcal{T}rigger$ boolean function, and (ii) the manner by which the current information about the incoming time sequence is taken into account.

Within this framework, we have proposed an optimization criterion that balances the expected gain in the classification cost in the future with the cost of delaying the decision. One important property of this criterion is that it can be computed at each time step for all future instants. This prediction of the future gains is updated given the current observation and is therefore never certain, but this yields a non myopic sequential decision process.

In this paper, we have sought to determine the baseline properties of our proposed framework. Thus, we have used simple techniques as: (i) clustering of time series in order to compare the incoming time sequence 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: naïve Bayes or a simple implementation of Multi-Layer Perceptrons.

In this baseline setting, it is a remarkable feat that the experiments exhibit a remarkable fit with desirable properties for an early decision classification algorithm, as stated in Section 6. The system indeed controls the decision time so as to ensure a high level of prediction performance as best as possible given the level of difficulty of the task and the cost of delaying the decision. It is also adaptive by taking into account the peculiarities of the incoming time sequence.

While we have obtained quite satisfying and promising results in the experiments carried out on controlled data and on a real data set, one direction for future work is to boost up this baseline implementation. In particular, we have ideas about how to use training sequences in order to predict the future decision cost of an incoming time sequence without using a clustering approach. Besides, dedicated methods for classifying time sequences should be used rather than naïve Bayes or simple MLP.

Still, even as it is, the method presented here should prove a useful tool for many early classification tasks.

References

1. DeGroot, M.H.: Optimal statistical decisions, vol. 82. John Wiley & Sons (2005)
2. Berger, J.O.: Statistical decision theory and Bayesian analysis. Springer Science & Business Media (1985)
3. Wald, A., Wolfowitz, J.: Optimum character of the sequential probability ratio test. The Annals of Mathematical Statistics, 326–339 (1948)
4. Sochman, J., Matas, J.: Waldboost-learning for time constrained sequential detection. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005, vol. 2, pp. 150–156. IEEE (2005)
5. Ishiguro, K., Sawada, H., Sakano, H.: Multi-class boosting for early classification of sequences. Statistics **28**, 337–407 (2000)
6. Xing, Z., Pei, J., Philip, S.Y.: Early prediction on time series: A nearest neighbor approach. In: IJCAI, pp. 1297–1302. Citeseer (2009)
7. Anderson, H.S., Parrish, N., Tsukida, K., Gupta, M.: Early time-series classification with reliability guarantee. Sandria Report (2012)
8. Parrish, N., Anderson, H.S., Gupta, M.R., Hsiao, D.Y.: Classifying with confidence from incomplete information. J. of Mach. Learning Research **14**, 3561–3589 (2013)
9. Hatami, N., Chira, C.: Classifiers with a reject option for early time-series classification. In: 2013 IEEE Symposium on Computational Intelligence and Ensemble Learning (CIEL), pp. 9–16. IEEE (2013)
10. Ghalwash, M.F., 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)
11. Xing, Z., Pei, J., Philip, S.Y., Wang, K.: Extracting interpretable features for early classification on time series. In: SDM, vol. 11, pp. 247–258. SIAM (2011)
12. Rousseeuw, P.J.: Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. J. of computational and applied mathematics **20**, 53–65 (1987)
13. Keogh, E., Xi, X., Wei, L., Ratanamahatana, C.A.: The ucr time series classification/clustering homepage (2006). www.cs.ucr.edu/eamonn/time_series_data/