Early Classification of Time Series

as a Non Myopic Sequential Decision Making Problem

A. Dachraoui[1,2], A. Bondu[1] and A. Cornuéjols[2]

[1] EDF R&D, [2]AgroParisTech-INRA MIA 518

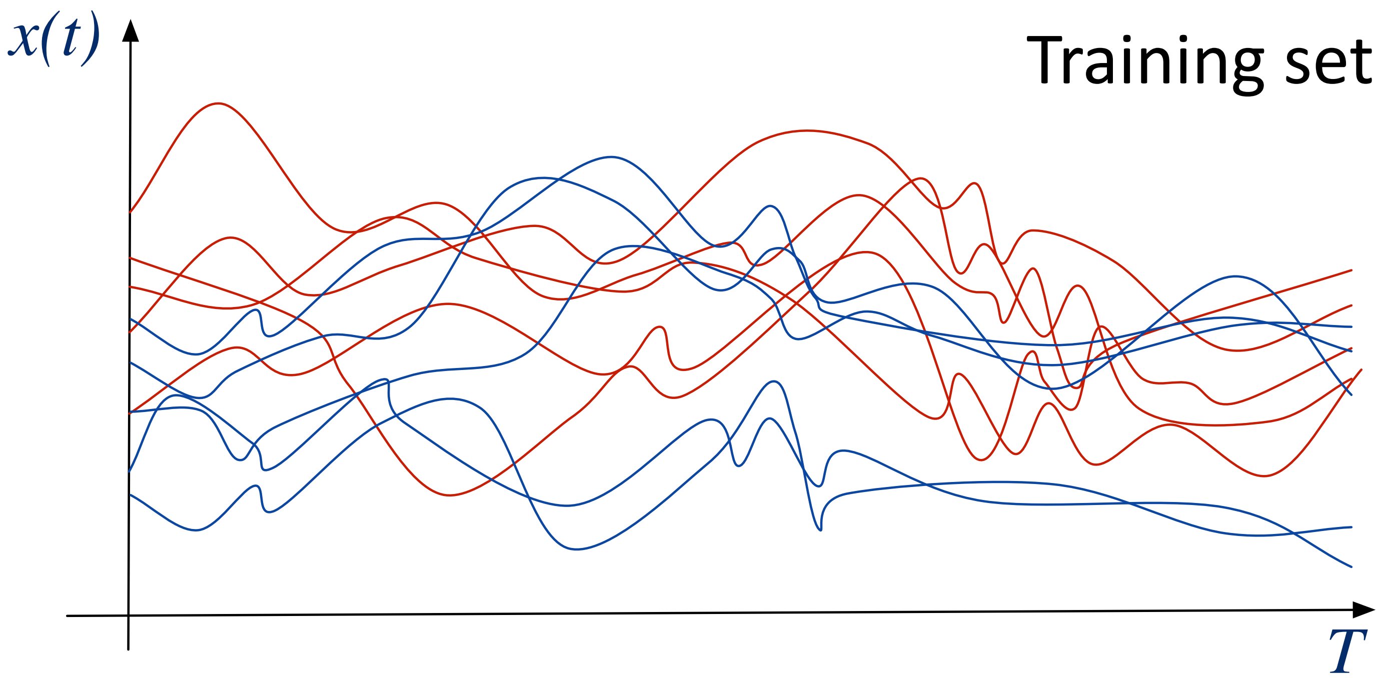
(Paris – France)

Outline

1. Introduction: a new set of problems
2. Formal statement and a solution
3. The proposed approach
4. Experimental results
5. Conclusions

Introduction

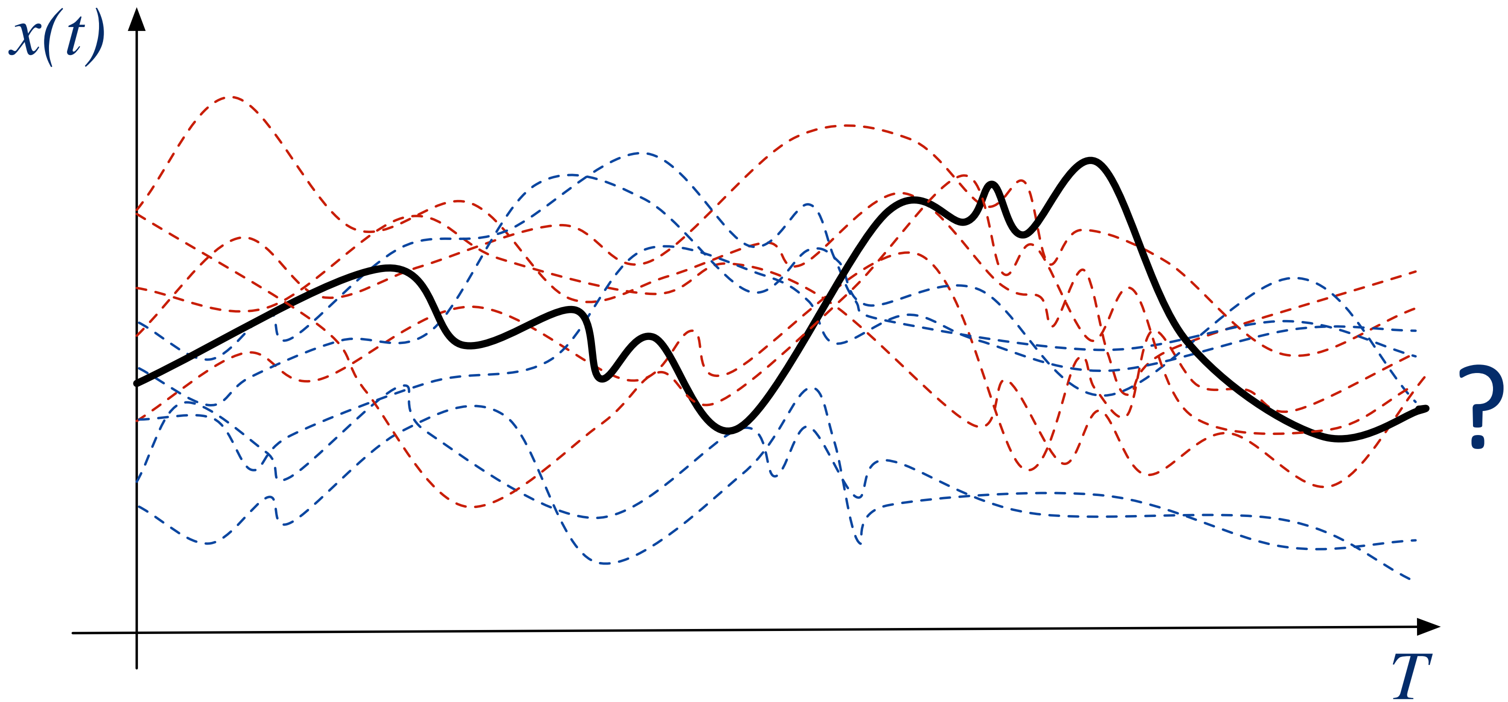
(Early) classification of time series



- Monitoring of ***consumer actions on a web site***: will buy or not
- Monitoring of a ***patient state***: critical or not
- Early prediction of daily ***electrical consumption***: high or low

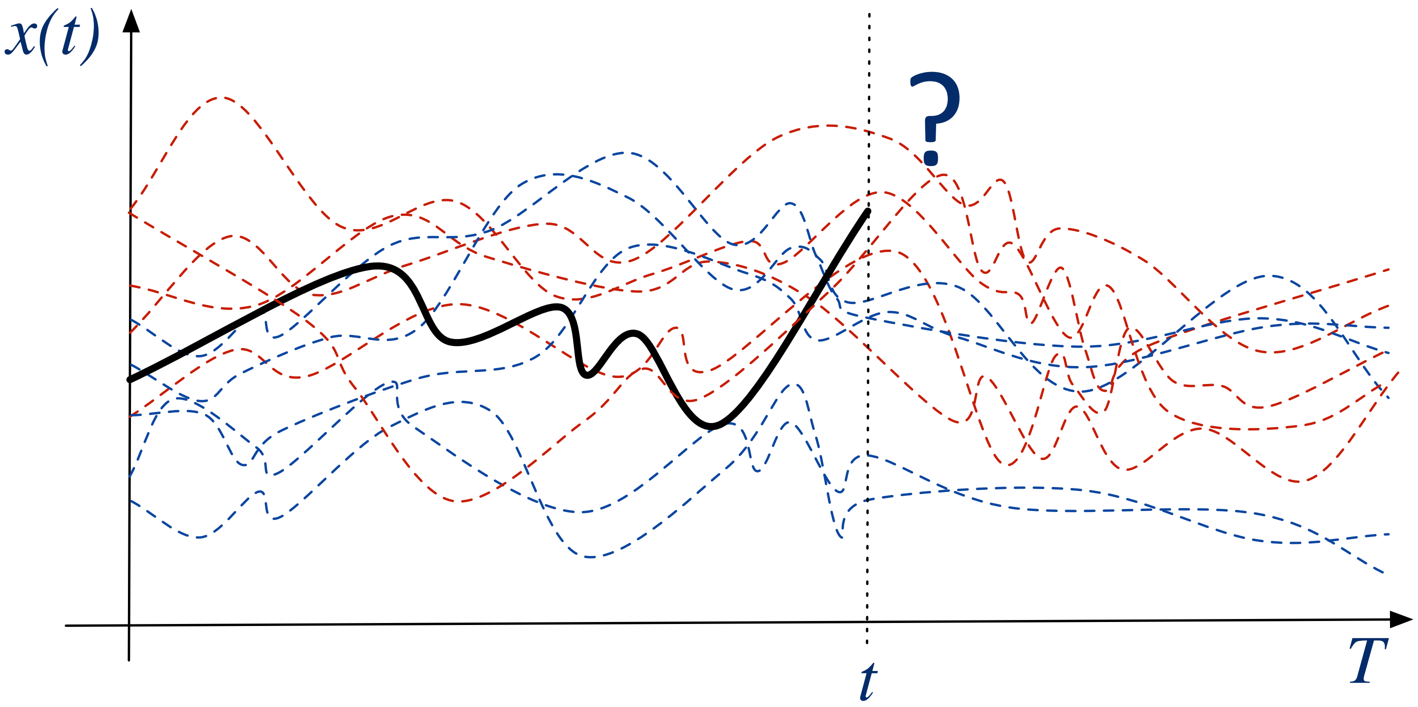
Standard classification of time series

- What is the class of the new time series x_T?



Early classification of time series

- What is the class of the new **incomplete** time series $\boldsymbol{x_t}$?



New set of decision problems : early classification

- Data **stream**
- **Classification** task
- As **early** as possible
- A **trade-off**
  - Classification **performance** (better if $t \nearrow$)
  - Cost of **delaying** prediction (better if $t \searrow$)

Previous works

- *Sequential decision making (1933, 1948)*
  - ***Wald's sequential probability ratio test***

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

$$h(\mathbf{x}_t) = \begin{cases} +1 & \text{if} \quad R_t > \alpha \\ -1 & \text{if} \quad R_t < \beta \\ \text{continue monitoring} & \text{if} \quad \beta < R_t < \alpha \end{cases}$$

Previous works

- *Sequential decision making (1933, 1948)*
  - ***Wald's sequential probability ratio test***

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

    - Difficult to estimate
    - Have to set the thresholds α and β

$$h(\mathbf{x}_t) = \begin{cases} +1 & \text{if } R_t > \alpha \\ -1 & \text{if } R_t < \beta \\ \text{continue monitoring} & \text{if } \beta < R_t < \alpha \end{cases}$$

    - No explicit cost of delaying decision
    - Myopic decision criterion

Previous works

- *Sequential decision making (1933, 1948)*
  - ***Wald's sequential probability ratio test***

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

    - Difficult to estimate
    - Have to set the thresholds α and β
    - No explicit cost of delaying decision
    - Myopic decision criterion

- Other **(modern) approaches**
    - Sophisticated heuristics
    - No explicit cost of delaying decision
    - Myopic decision criterion

[Ishiguro et al. 2000]
[Sochman & Matas, 2005]
[Xing et al., 2009, 2011]
[Anderson et al., 2012]
[Parrish et al., 2013]
[Hatami & Chira, 2013]
[Ghalwash et al., 2014]
…

Our contribution

1. **Formalize** the problem as a sequential decision making problem

2. Explicit **trade-off**
   - Classification **performance**
   - **Cost of delaying** the decision

3. **Adaptive**
   - Takes into account the peculiarities of x_t

4. **Non myopic**
   - At each time step, estimates the expected future time for optimal decision

*An **algorithm** that realizes all these items*

Formal analysis and first attempt

Decision making (1)

- Given an incoming sequence $\mathbf{x}_t = \langle x_1, x_2, \ldots, x_t \rangle \quad \text{where } x_t \in \mathbb{R}$
- And given:
  - A *miss-classification cost function* $C_t(\hat{y}|y) : \mathcal{Y} \times \mathcal{Y} \longrightarrow \mathbb{R}$
  - A *delaying decision cost function* $C(t) : \mathbb{N} \longrightarrow \mathbb{R}$
- What is the **optimal time** to make a **decision?**

Decision making (1)

- Given an incoming sequence $\mathbf{x}_t = \langle x_1, x_2, \ldots, x_t \rangle$ where $x_t \in \mathbb{R}$
- And given:
  - A *miss-classification cost function* $C_t(\hat{y}|y) : \mathcal{Y} \times \mathcal{Y} \longrightarrow \mathbb{R}$
  - A *delaying decision cost function* $C(t) : \mathbb{N} \longrightarrow \mathbb{R}$
- **What is the optimal time to make a decision?**

Expected cost for a decision at time *t*

$$f(\mathbf{x}_t) = \underbrace{\sum_{y \in \mathcal{Y}} P(y|\mathbf{x}_t) \sum_{\hat{y} \in \mathcal{Y}} P(\hat{y}|y, \mathbf{x}_t)\, C_t(\hat{y}|y)}_{\text{expected miss-classification cost given } \mathbf{x}_t} + C(t)$$

Decision making (1)

- Given an incoming sequence $\mathbf{x}_t = \langle x_1, x_2, \ldots, x_t \rangle$ where $x_t \in \mathbb{R}$
- And given:
  - A *miss-classification cost function* $C_t(\hat{y}|y) : \mathcal{Y} \times \mathcal{Y} \longrightarrow \mathbb{R}$
  - A *delaying decision cost function* $C(t) : \mathbb{N} \longrightarrow \mathbb{R}$
- **What is the optimal time to make a decision?**

Expected cost for a decision at time *t*

$$f(\mathbf{x}_t) = \underbrace{\sum_{y \in \mathcal{Y}} P(y|\mathbf{x}_t) \sum_{\hat{y} \in \mathcal{Y}} P(\hat{y}|y, \mathbf{x}_t)\, C_t(\hat{y}|y)}_{\text{expected miss-classification cost given } \mathbf{x}_t} + C(t)$$

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

Decision making (1)

- Given an incoming sequence $\mathbf{x}_t = \langle x_1, x_2, \ldots, x_t \rangle$ where $x_t \in \mathbb{R}$
- And given:
  - A *miss-classification cost function* $C_t(\hat{y}|y) : \mathcal{Y} \times \mathcal{Y} \longrightarrow \mathbb{R}$
  - A *delaying decision cost function* $C(t) : \mathbb{N} \longrightarrow \mathbb{R}$
- **What is the optimal time to make a decision?**

?

Expected cost for a decision at time *t*

$$f(\mathbf{x}_t) = \underbrace{\sum_{y \in \mathcal{Y}} P(y|\mathbf{x}_t) \sum_{\hat{y} \in \mathcal{Y}} P(\hat{y}|y, \mathbf{x}_t)\, C_t(\hat{y}|y)}_{\text{expected miss-classification cost given } \mathbf{x}_t} + C(t)$$

Optimal time: $t^* = \underset{t \in \{1,\ldots,T\}}{\text{ArgMin}}\ f(\mathbf{x}_t)$

Decision making (2)

- Expected cost for any sequence of length *t*

$$f(t) \;=\; \underbrace{\sum_{y \in \mathcal{Y}} P(y) \sum_{\hat{y} \in \mathcal{Y}} P_t(\hat{y}|y)\, C_t(\hat{y}|y)}_{\text{expected miss-classification cost given } t} \;+\; C(t)$$

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

Decision making (2)

- Expected cost for any sequence of length *t*

$$f(t) = \underbrace{\sum_{y \in \mathcal{Y}} P(y) \sum_{\hat{y} \in \mathcal{Y}} P_t(\hat{y}|y)\, C_t(\hat{y}|y)}_{\text{expected miss-classification cost given } t} + C(t)$$

Computable from a training set

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

Decision making (2)

- Expected cost for any sequence of length *t*

$$f(t) = \underbrace{\sum_{y \in \mathcal{Y}} P(y) \sum_{\hat{y} \in \mathcal{Y}} P_t(\hat{y}|y) \, C_t(\hat{y}|y)}_{\text{expected miss-classification cost given } t} + C(t)$$

Computable from a training set

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

- **Not** adaptive!!

Challenge

- How can we get an optimization criterion st.
  - It takes into account the incoming sequence x_t (**adaptive**)
  - It can be **easily estimated** (from the training set)

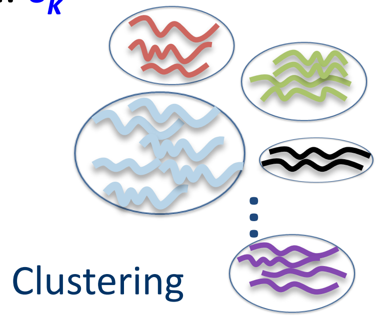
The proposed approach

The principle $P(y|\mathbf{x}_t) \rightarrow P(y|\mathfrak{c}_k)$

The principle $P(y|\mathbf{x}_t) \rightarrow P(y|\mathfrak{c}_k)$

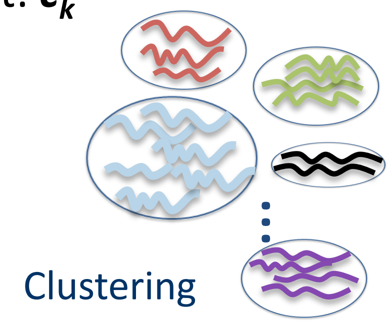
1. During **training**:

→ – identify **meaningful subsets** of time sequences in the training set: c_k

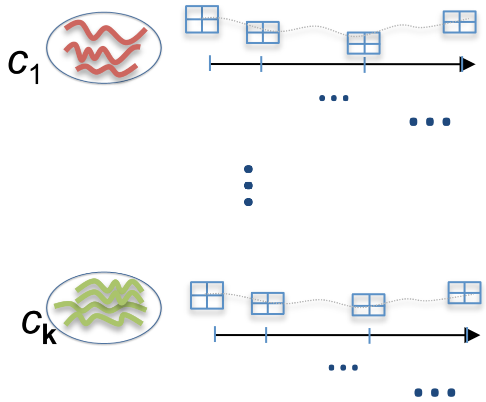


The principle

1. During **training**:
   - identify **meaningful subsets** of time sequences in the training set: c_k
   - **For each of these subsets** c_k, and for each time step t
     - Estimate the **confusion matrices** $P_t(\hat{y}|y, \mathfrak{c}_k)$

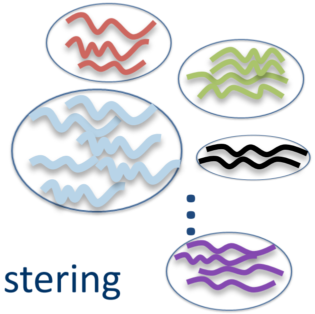


- T classifiers are **learnt** $h_t(\mathbf{x}_t) : \mathcal{X}_t \rightarrow \mathcal{Y}$
- And their confusion matrices $P_t(\hat{y}|y, \mathfrak{c}_k)$ are **estimated** on a test set



The principle

1. During **training**:
   - identify **meaningful subsets** of time sequences in the training set: c_k
   - For each of these subsets c_k, and for each time step t
     - Estimate the **confusion matrices** $P_t(\hat{y}|y, \mathfrak{c}_k)$



Clustering

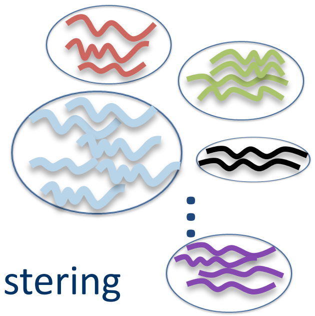
2. **Testing**: For any new incomplete incoming sequence x_t
   - Identify the **most likely subset**: the closer class of shapes to x_t

Membership probability

$$P(\mathfrak{c}_k|\mathbf{x}_t) = \frac{s_k}{\sum_i^K s_i}, \quad \text{where } s_k = \frac{1}{1 + \exp^{-\lambda(\bar{D} - d_k)/\bar{D}}}$$

The principle

1. During **training**:
   - identify **meaningful subsets** of time sequences in the training set: $\mathfrak{c}_k$
   - For each of these subsets c_k, and for each time step t
     - Estimate the **confusion matrices** $P_t(\hat{y}|y, \mathfrak{c}_k)$



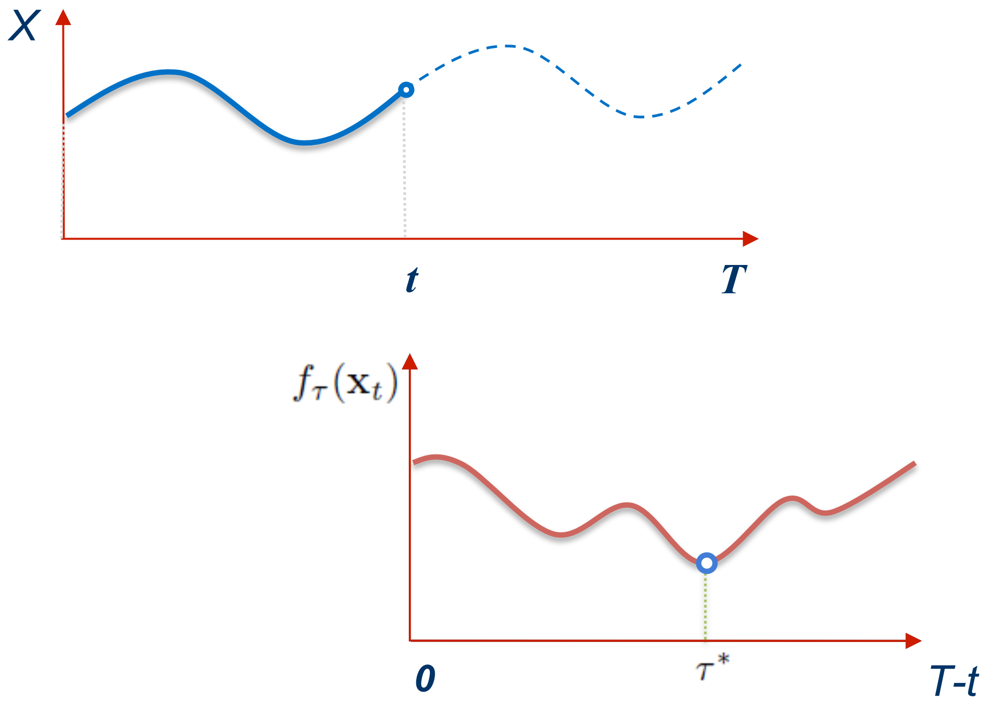
2. **Testing**: For any new incomplete incoming sequence x_t
   - Identify the most likely subset: the closer shape to x_t
   - Compute the **expected cost of decision** for all future time steps

$$f_\tau(\mathbf{x}_t) \; = \; \underbrace{\sum_{\mathfrak{c}_k \in \mathcal{C}} P(\mathfrak{c}_k|\mathbf{x}_t) \sum_{y \in \mathcal{Y}} P(y|\mathfrak{c}_k) \sum_{\hat{y} \in \mathcal{Y}} P_{t+\tau}(\hat{y}|y, \mathfrak{c}_k) \, C(\hat{y}|y)}_{\text{expected miss-classification cost given } \mathbf{x}_t} \; + \; C(t+\tau)$$

A non myopic decision process

- Optimal estimated time relative to current time t

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

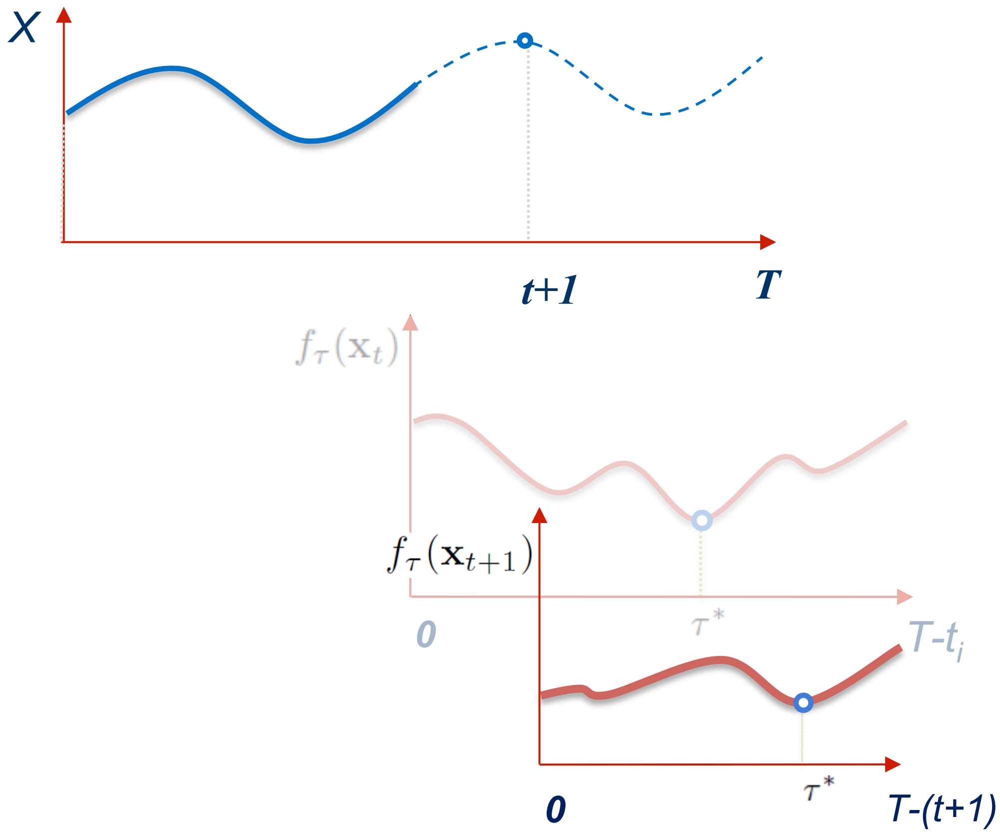


→ Continue monitoring

A non myopic decision process

- Optimal estimated time relative to current time *t*

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

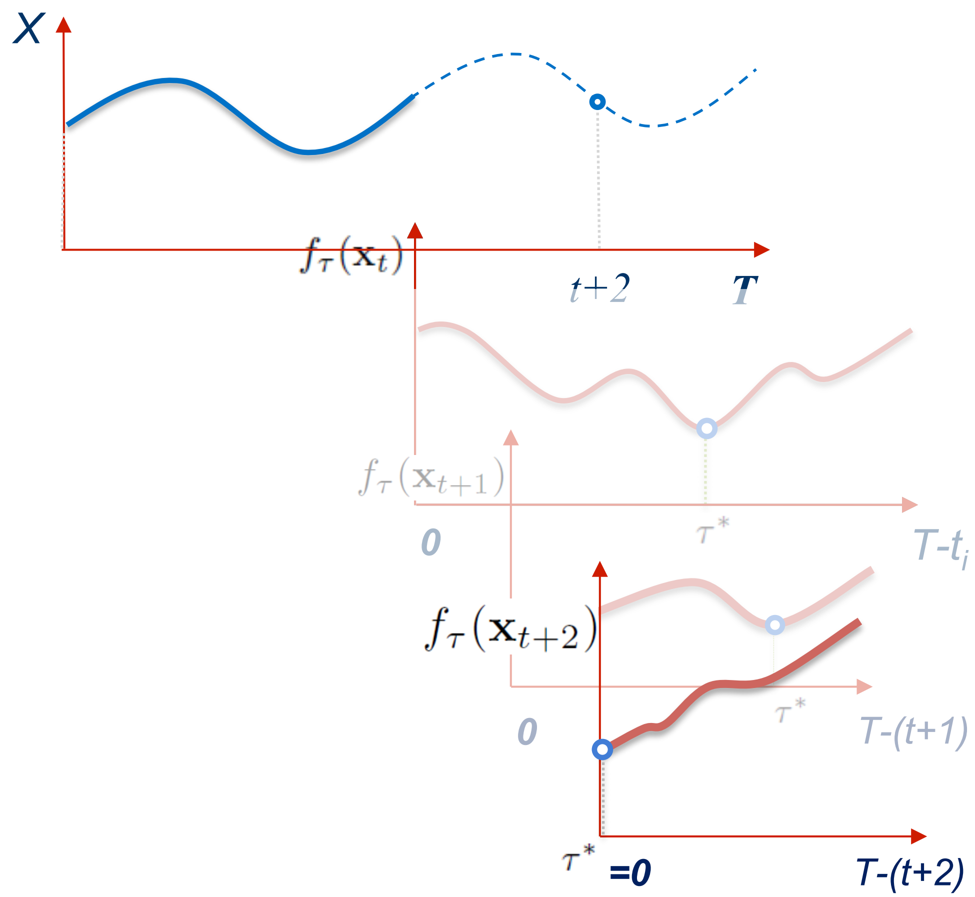


→ Continue monitoring

A non myopic decision process

- Optimal estimated time relative to current time *t*

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



→ Take **decision**

Properties

- The decision criterion **naturally incorporates**
  - The quality of the decision
  - The cost of delaying decision

- **Adaptive**: the decision depends upon x_t

- **Non** myopic:
  - At each time step the expected best time for decision is estimated

A simple implementation

A baseline implementation: simple and direct

- The **distance** used to measure proximity between sequences is the Euclidian distance
  - Clustering of sequences
  - Clustering membership $$P(\mathfrak{c}_k|\mathbf{x}_t) = \frac{s_k}{\sum_i^K s_i}, \quad \text{where } s_k = \frac{1}{1+\exp^{-\lambda(\bar{D}-d_k)/\bar{D}}}$$

A baseline implementation: simple and direct

- The **distance** used to measure proximity between sequences is the Euclidian distance
  - Clustering of sequences
  - Clustering membership

$$P(\mathfrak{c}_k|\mathbf{x}_t) = \frac{s_k}{\sum_i^K s_i}, \quad \text{where } s_k = \frac{1}{1 + \exp^{-\lambda(\bar{D} - d_k)/\bar{D}}}$$

- **Classifiers**

$$\hat{y} = h_t(\mathbf{x}_t)$$

  - Naïve Bayes
  - → Multi-Layer Perceptrons

A baseline implementation: simple and direct

- The **distance** used to measure proximity between sequences is the Euclidian distance
  - Clustering of sequences
  - Clustering membership $$P(\mathfrak{c}_k|\mathbf{x}_t) = \frac{s_k}{\sum_i^K s_i}, \quad \text{where } s_k = \frac{1}{1 + \exp^{-\lambda(\bar{D} - d_k)/\bar{D}}}$$

- **Classifiers** $$\hat{y} = h_t(\mathbf{x}_t)$$
  - Naïve Bayes
  - Multi-Layer Perceptrons

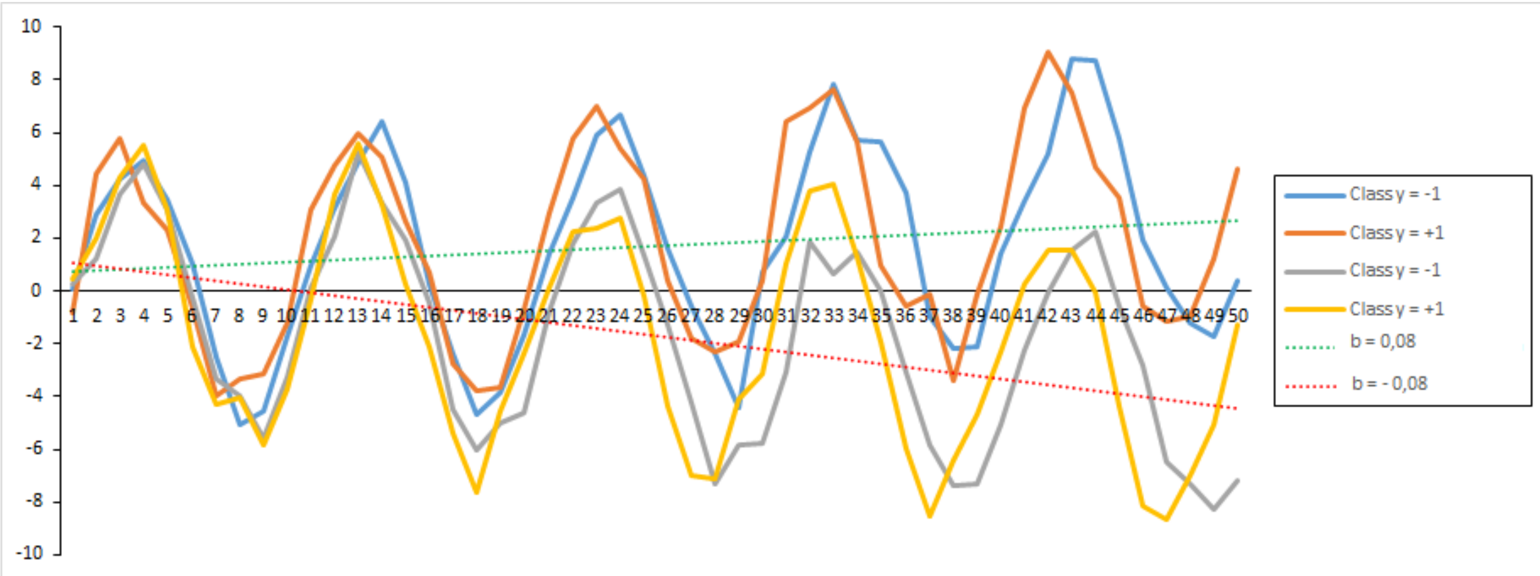
→ **Other choices are possible** within the general approach

Experiments

Controlled data

- Control of
  - The **proximity** between the **classes**
  - The number and shapes of **clusters** within each class
  - The **noise level**

$$\mathbf{x}_t = a \sin(\omega_i t + phase) + bt + \varepsilon(t)$$



Results

$C(t)$	$\pm b$ / $\varepsilon(t)$	0.02 $\overline{\tau}^\star$	0.02 $\sigma(\tau^\star)$	0.02 AUC	0.05 $\overline{\tau}^\star$	0.05 $\sigma(\tau^\star)$	0.05 AUC	0.07 $\overline{\tau}^\star$	0.07 $\sigma(\tau^\star)$	0.07 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

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.

Results: effect of the noise level

Increasing the **noise level** increases the waiting time, and then it's no longer worth it

$C(t)$	$\pm b$	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

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.

Results: effect of the waiting cost

Increasing the **waiting cost** **reduces** the waiting time

$C(t)$	$\pm b$ / $\varepsilon(t)$	0.02 $\tau^\star$	0.02 $\sigma(\tau^\star)$	0.02 AUC	0.05 $\tau^\star$	0.05 $\sigma(\tau^\star)$	0.05 AUC	0.07 $\tau^\star$	0.07 $\sigma(\tau^\star)$	0.07 AUC
	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
0.01	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
	0.2	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
0.05	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
	0.2	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
0.10	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

Table 2. 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.

Results: effect of the difference between classes

Increase of the difference between classes

The **performance** increases (AUC)

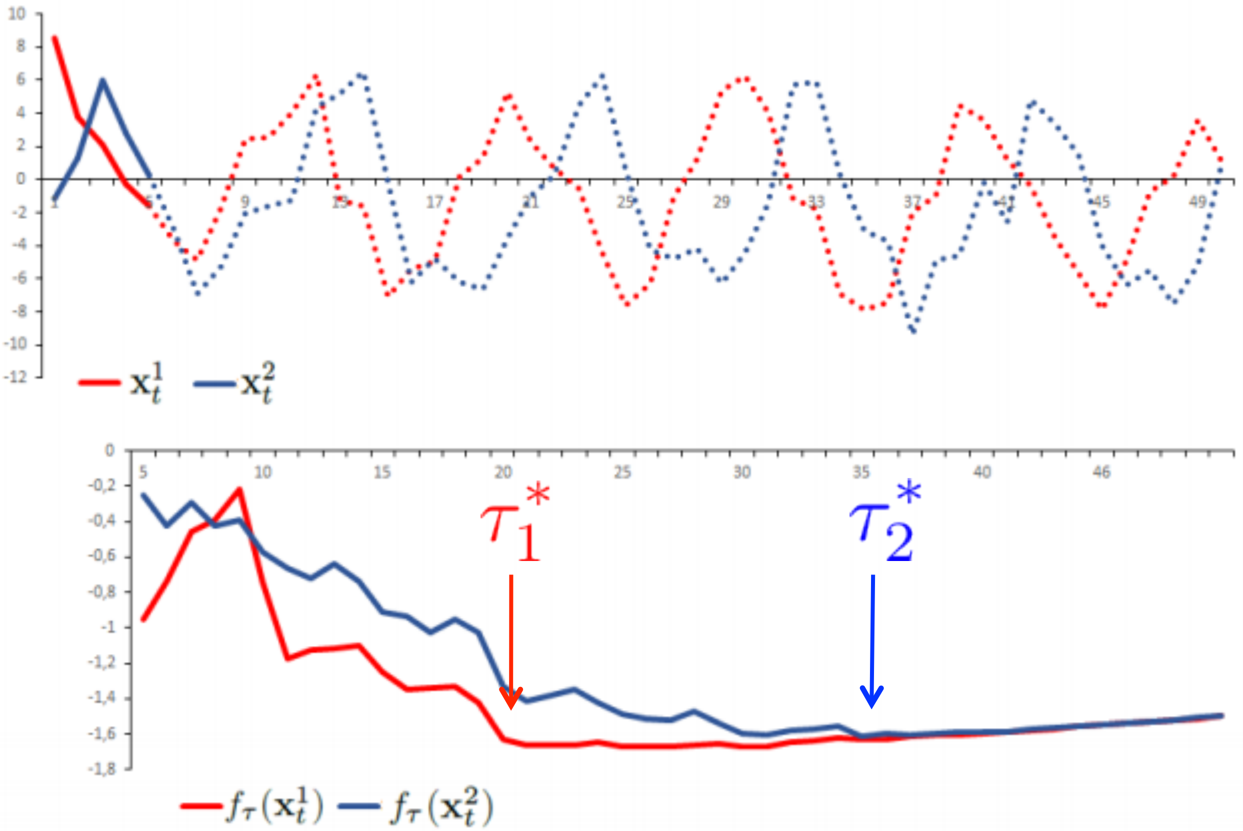
The *waiting time* is not much changed in these experiments

$C(t)$	$\pm b$ / $\varepsilon(t)$	0.02			0.05			0.07		
		$\overline{\tau}^\star$	$\sigma(\tau^\star)$	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

Table 3. 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.

Results: the method is adaptive

- The expected optimal decision time depends on the incoming sequence



Real dataset

- **TwoLeadECG** (UCI repository)
- 1,162 signals of **81** measurements each (81 minutes)
- Two classes
- We arbitrarily varied the waiting cost:
  - C(*t*) = 0.01 . *t* (cheap)
  - C(*t*) = 0.05 . *t* (costly)
  - C(*t*) = 0.10 . *t* (very costly)

$C(t)$	**0.01**	**0.05**	**0.1**
$\overline{\tau}^{\star}$	**22.0**	**24.0**	**10.0**
$\sigma(\tau^{\star})$	6.1	15.7	9.8
AUC	**0.99**	**0.99**	**0.91**

Adapts to keep a good performance with fewer measurements

Conclusions

Conclusions

- **Online classification** of data streams is increasingly important
- **Contribution**
  - A **new optimization criterion** incorporating
    - Classification performance
    - Cost of delaying decision
  - A **baseline method**
    - Adaptive
    - Non myopic
    - A spectrum of different implementations is possible
  - **Experimental results** show the promise of the method

Perspectives

- **Exploration** of the spectrum of variations

  1. **Better clustering** method of the training sequences
     - More informed distance
  2. A **more direct approach** without clustering on sequences
  3. **Better classifier** of incomplete sequences

- **Application** to electrical grid data