Early Classification of Time Series

How to solve the corresponding tradeoff

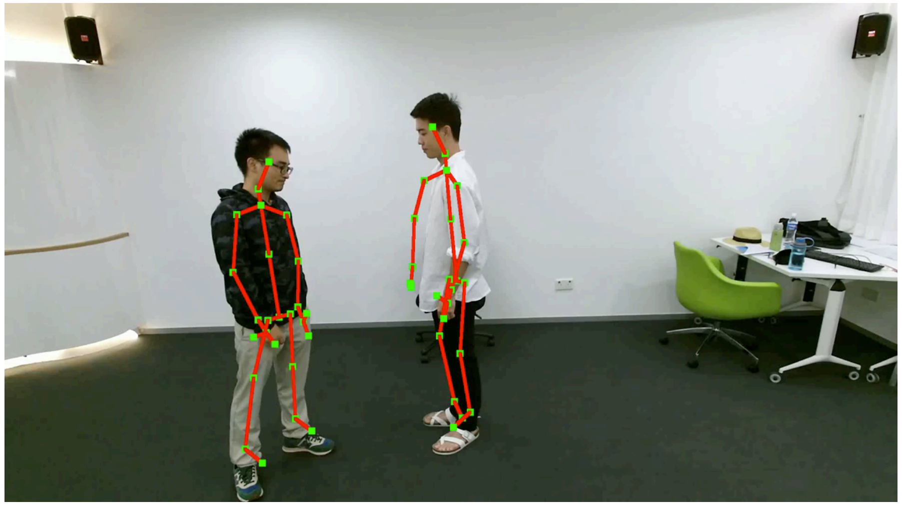
Antoine Cornuéjols

AgroParisTech – **INRAe** MIA Paris-Saclay

EKINOCS research group

- **Ekinocs** research team
  - Machine Learning for
    - **Life** science
      - Bioinformatics
    - **Agriculture**
      - Satellite image analysis: monitoring changes in the land uses
      - Control of irrigation
      - Predicting late frost
      - …
    - **Nutrition**
      - Changing the consumers habits to turn from the consumption of animal proteins to the consumption of vegetal ones

Human activity recognition



- Recognize what they are doing **as fast as possible** but with a **high accuracy**

  Are they

  - Playing?
  - Fighting?

Outline

Supervised learning

- From a **training** set

$$S = \{(x_1, y_1), (x_2, y_2), \ldots, (x_i \overset{f}{\longrightarrow} y_i), \ldots, (x_m, y_m)\}$$

Supervised learning

- From a **training** set

$$S = \{(x_1, y_1), (x_2, y_2), \ldots, (x_i, y_i), \ldots, (x_m, y_m)\}$$

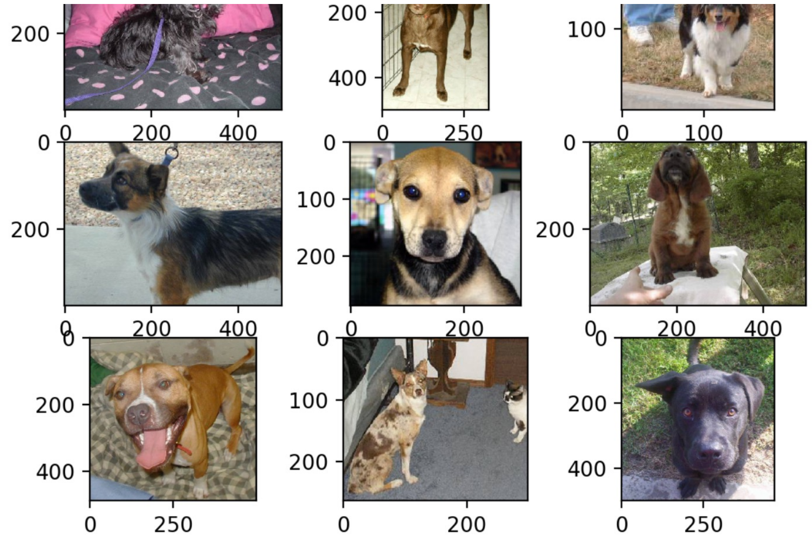
f

h

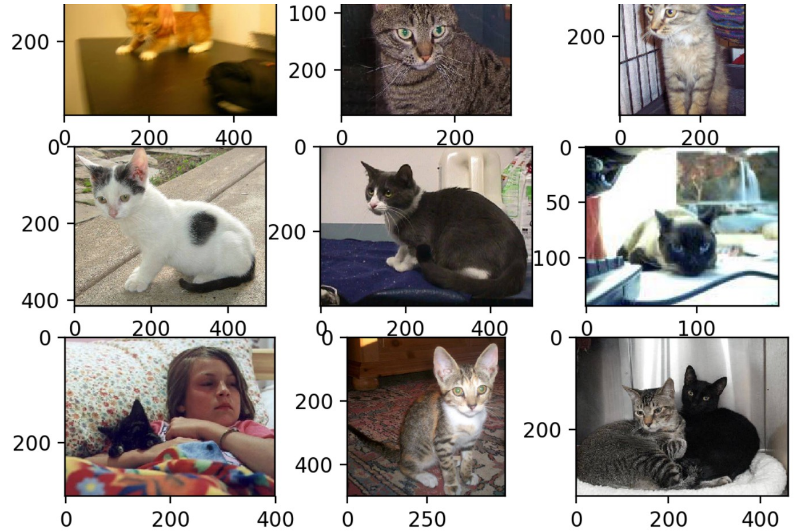
- **Learn** a function from the **input** space X to the **output** space Y

$$x - h \rightarrow y$$

From a **training** set



dogs



cats

Learn a function from an **input** space *X* to an **output** space *Y*

Cat or **dog?**

...

Dog
or
muffin?

One example that tells a lot …

- Examples described using:

  Number (1 or 2); ***size*** (small or large); ***shape*** (circle or square); ***color*** (red or green)

- They belong either to class '**+**' or to class '**-**'

- Examples described using:

  Number (1 or 2); ***size*** (small or large); ***shape*** (circle or square); ***color*** (red or green)

- They belong either to class '**+**' or to class '**-**'

Description	Your answer	True answer
1 large red square		

AgroParisTech

- Examples described using:

  Number (1 or 2); ***size*** (small or large); ***shape*** (circle or square); ***color*** (red or green)

- They belong either to class '**+**' or to class '**-**'

Description	Your answer	True answer
1 large red square		-

- Examples described using:

  Number (1 or 2); ***size*** (small or large); ***shape*** (circle or square); ***color*** (red or green)

- They belong either to class '**+**' or to class '**-**'

Description	Your answer	True answer
1 large red square		-
1 large green square		

- Examples described using:

  Number (1 or 2); ***size*** (small or large); ***shape*** (circle or square); ***color*** (red or green)

- They belong either to class '**+**' or to class '**-**'

Description	Your answer	True answer
1 large red square		-
1 large green square		+

- Examples described using:

  Number (1 or 2); ***size*** (small or large); ***shape*** (circle or square); ***color*** (red or green)

- They belong either to class '**+**' or to class '**-**'

Description	Your answer	True answer
1 large red square		-
1 large green square		+
2 small red squares		

- Examples described using:

  Number (1 or 2); ***size*** (small or large); ***shape*** (circle or square); ***color*** (red or green)

- They belong either to class '**+**' or to class '**-**'

Description	Your answer	True answer
1 large red square		-
1 large green square		+
2 small red squares		+

- Examples described using:

  Number (1 or 2); ***size*** (small or large); ***shape*** (circle or square); ***color*** (red or green)

- They belong either to class '**+**' or to class '**-**'

Description	Your answer	True answer
1 large red square		-
1 large green square		+
2 small red squares		+
2 large red circles		

- Examples described using:

  Number (1 or 2); ***size*** (small or large); ***shape*** (circle or square); ***color*** (red or green)

- They belong either to class '**+**' or to class '**-**'

Description	Your answer	True answer
1 large red square		-
1 large green square		+
2 small red squares		+
2 large red circles		-

- Examples described using:

  Number (1 or 2); ***size*** (small or large); ***shape*** (circle or square); ***color*** (red or green)

- They belong either to class '**+**' or to class '**-**'

Description	Your answer	True answer
1 large red square		-
1 large green square		+
2 small red squares		+
2 large red circles		-
1 large green circle		

- Examples described using:

  Number (1 or 2); ***size*** (small or large); ***shape*** (circle or square); ***color*** (red or green)

- They belong either to class '**+**' or to class '**-**'

Description	Your answer	True answer
1 large red square		-
1 large green square		+
2 small red squares		+
2 large red circles		-
1 large green circle		+

- Examples described using:

  Number (1 or 2); ***size*** (small or large); ***shape*** (circle or square); ***color*** (red or green)

- They belong either to class '**+**' or to class '**-**'

Description	Your answer	True answer
1 large red square		-
1 large green square		+
2 small red squares		+
2 large red circles		-
1 large green circle		+
1 small red circle		

- Examples described using:

  Number (1 or 2); ***size*** (small or large); ***shape*** (circle or square); ***color*** (red or green)

- They belong either to class '**+**' or to class '**-**'

Description	Your answer	True answer
1 large red square		-
1 large green square		+
2 small red squares		+
2 large red circles		-
1 large green circle		+
1 small red circle		+

- When would you be **certain** about your guess?

One example that tells a lot ...

- Examples described using:

  Number (1 or 2); ***size*** (small or large); ***shape*** (circle or square); ***color*** (red or green)

Description	**Your** prediction	**True** class
1 large red square		-
1 large green square		+
2 small red squares		+
2 large red circles		-
1 large green circle		+
1 small red circle		+

How many possible functions altogether from *X* to *Y* ? $2^{2^4} = 2^{16} = 65{,}536$

How many functions do remain after 9 training examples? $2^5 = 32$

AgroParisTech

Induction: an impossible game?

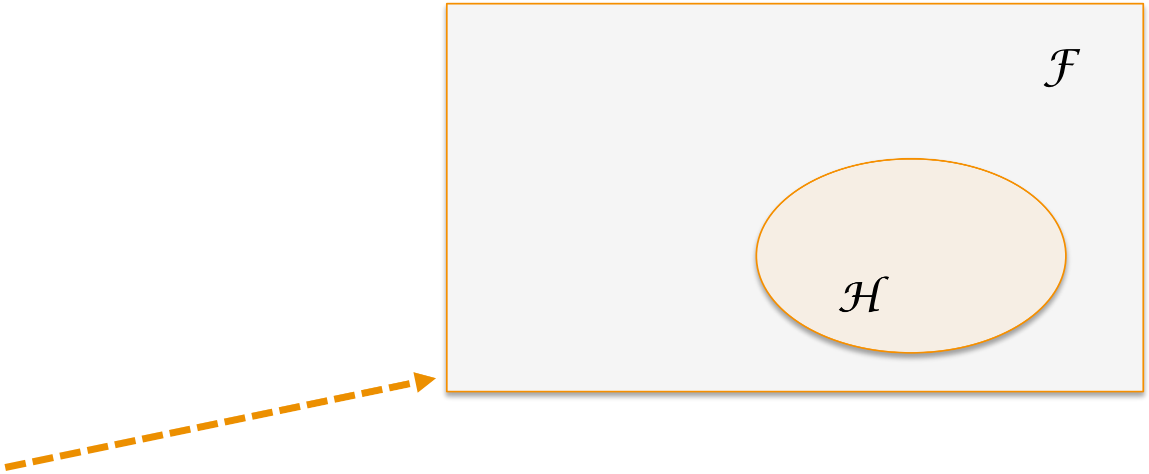
- **A bias is need**

Induction: an impossible game?

- **A bias is need**

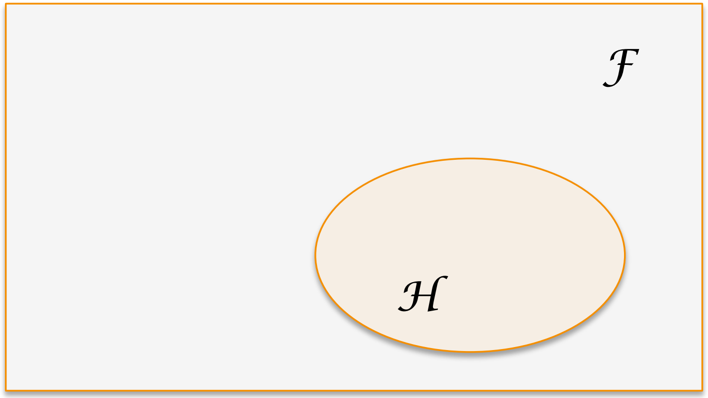
- **Types** of bias

  – **Representation** bias (declarative)

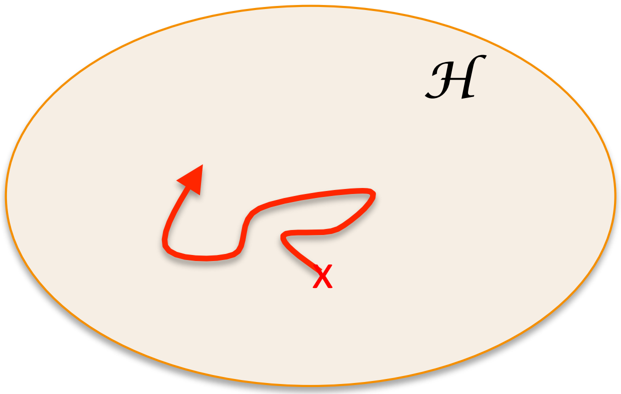


Induction: an impossible game?

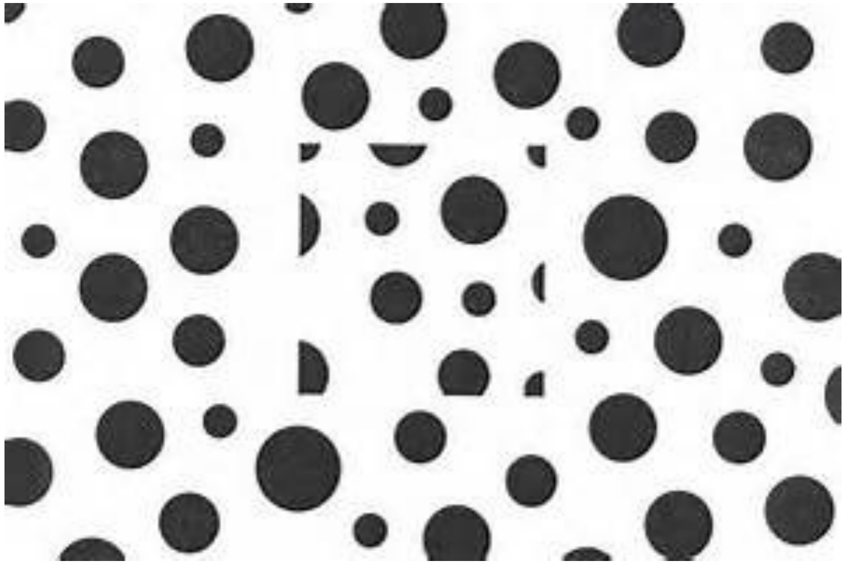
- **A bias is need**

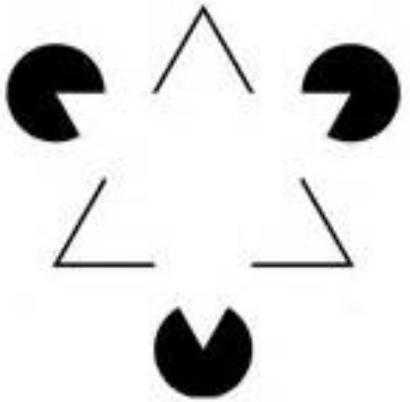


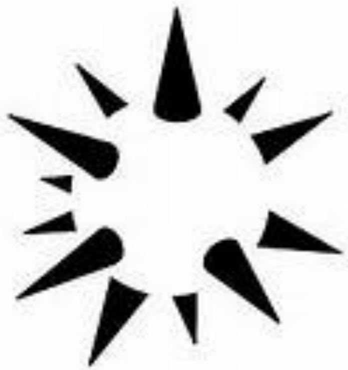
- **Types** of bias
  - **Representation** bias (declarative)
  - **Research** bias (procedural)



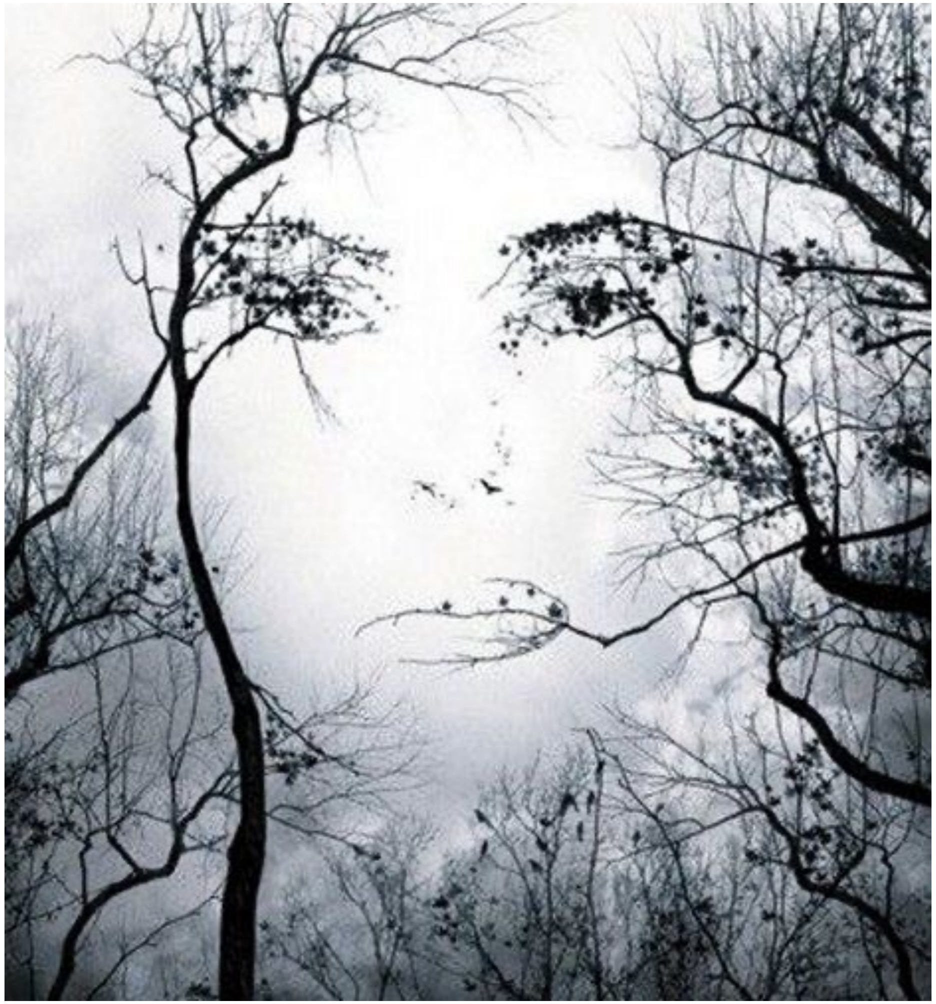
Interpretation – completion of percepts





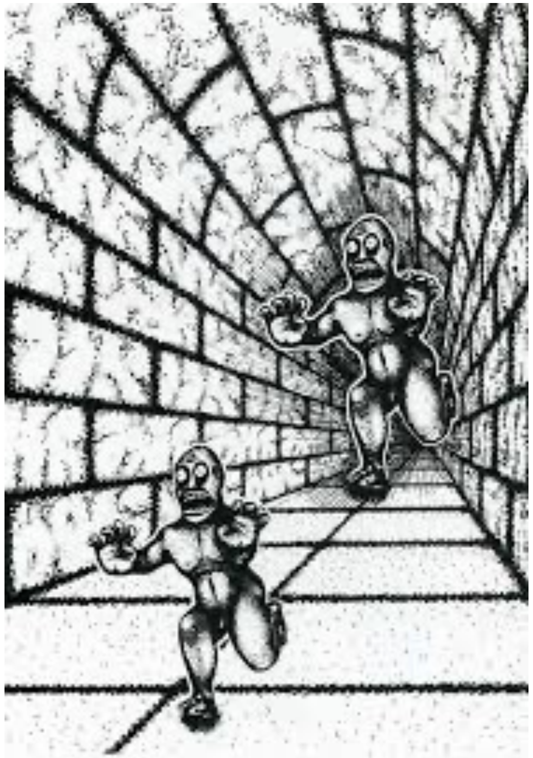


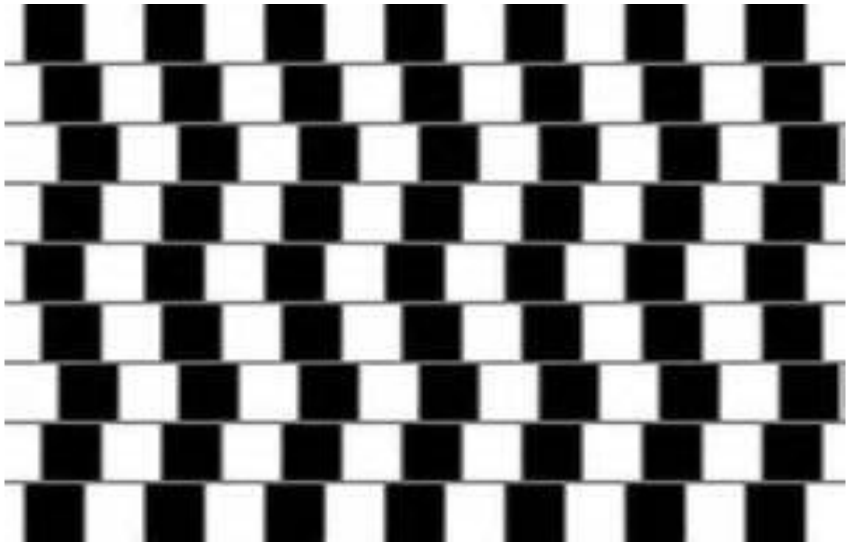
Interpretation – completion of percepts



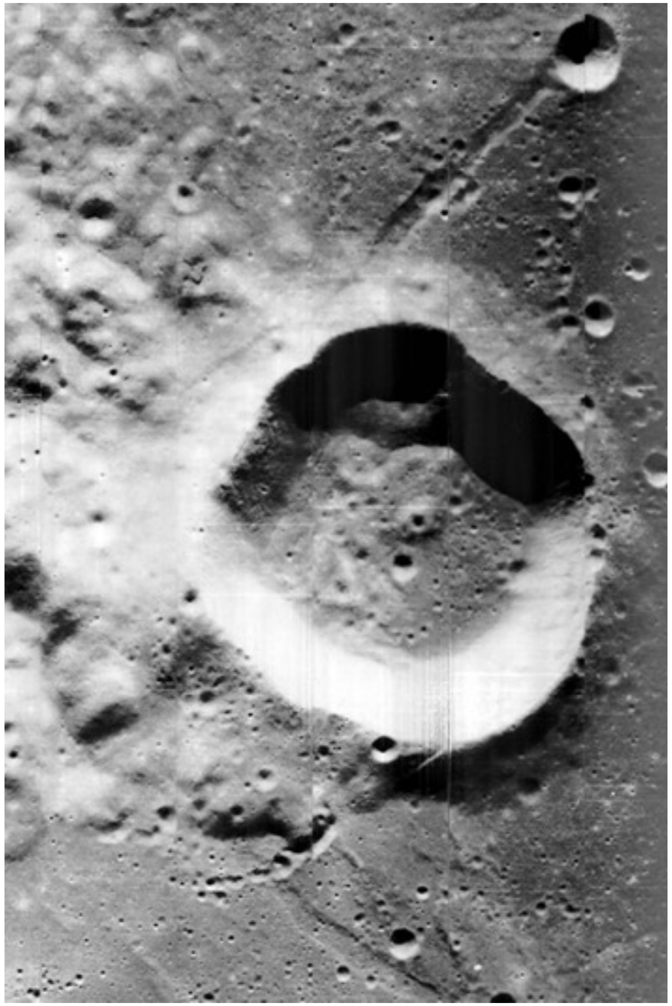
Interprétation – complétion de percepts

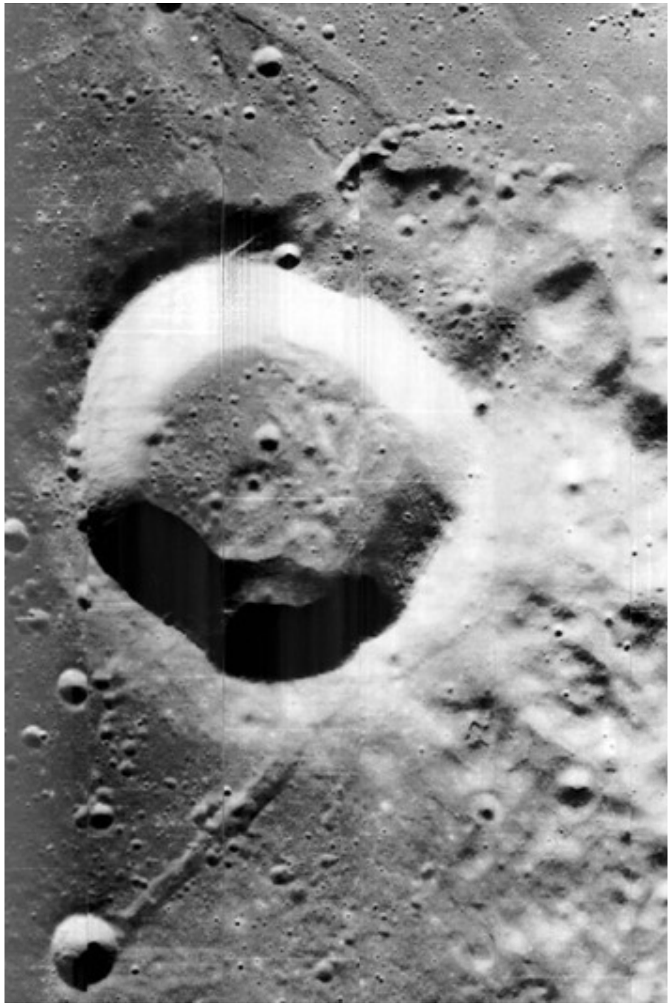
Optical illusions





Induction and its illusions

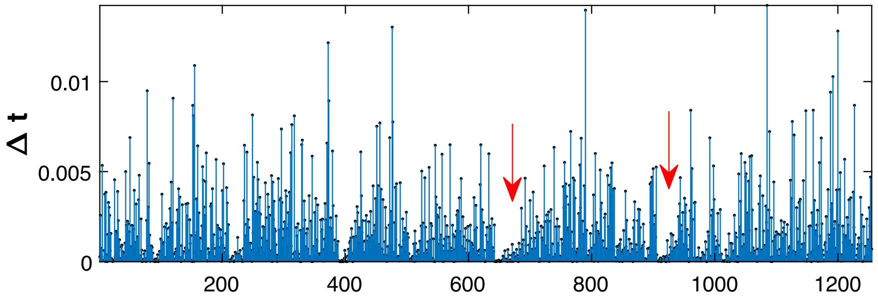


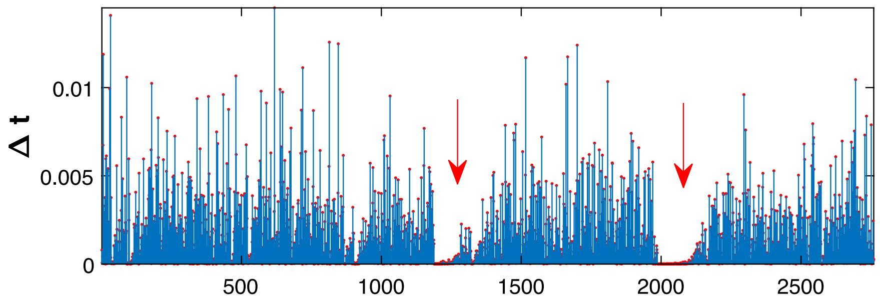


Illustration

Back to time series

Time Series



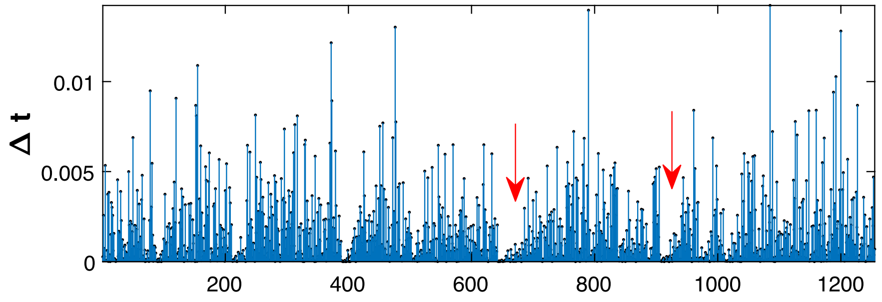


Subsequences obtained from Sumatra-Andaman earthquake time-series

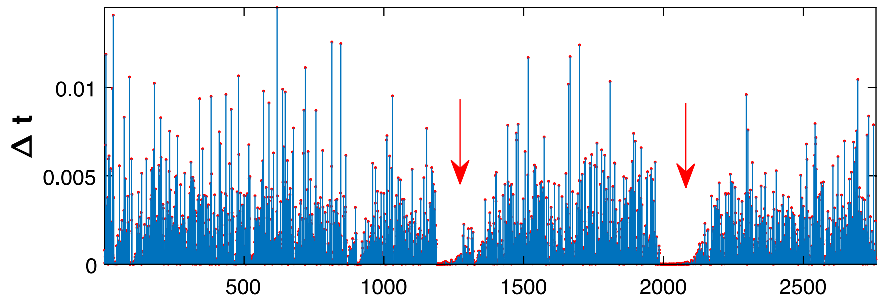
[Vijay, R. K., & Nanda, S. J. (2023). **Earthquake pattern analysis using subsequence time series clustering**. *Pattern Analysis and Applications*, 26(1), 19-37.]

Classification of Time Series

Suppose:



Not indicative of an **incoming earthquake**

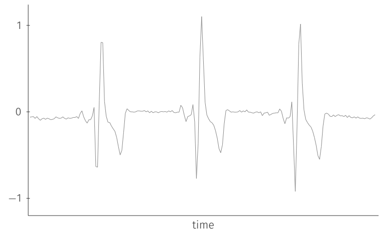


Indicative of an **incoming earthquake**

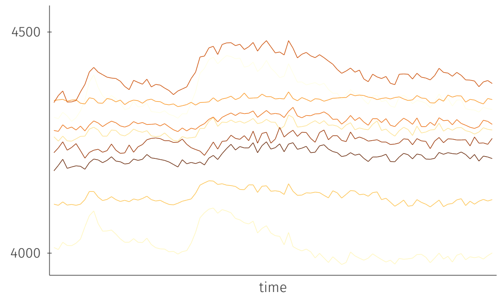
Learn a predictive **function**

Characteristiques

- We will be interested in **real valued** time series
  - Stock market values, electrical consumption, temperature, …



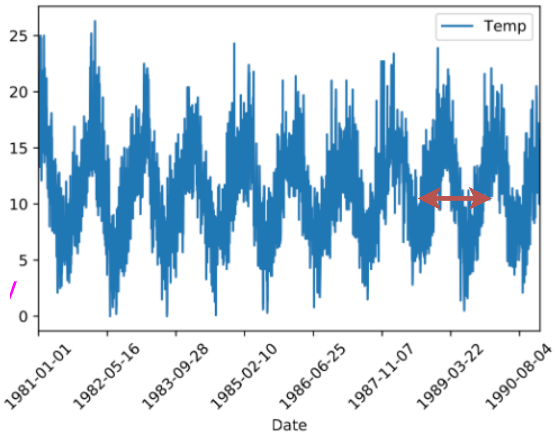
- **Uni**variate vs. **Multi**variate
  - Electrocardiogram vs. Electroencephalogram

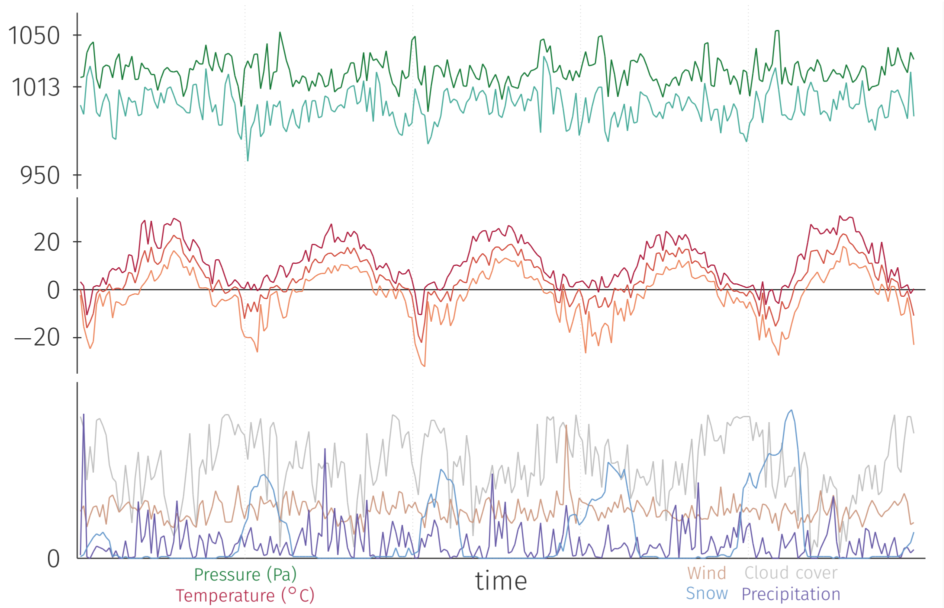


- **Periodic** sampling vs. **Irregular** sampling
  - Stock market values vs. On-line purchases

Univariate vs. multivariate

Single instance





...

Lots of phenomena are **temporal**

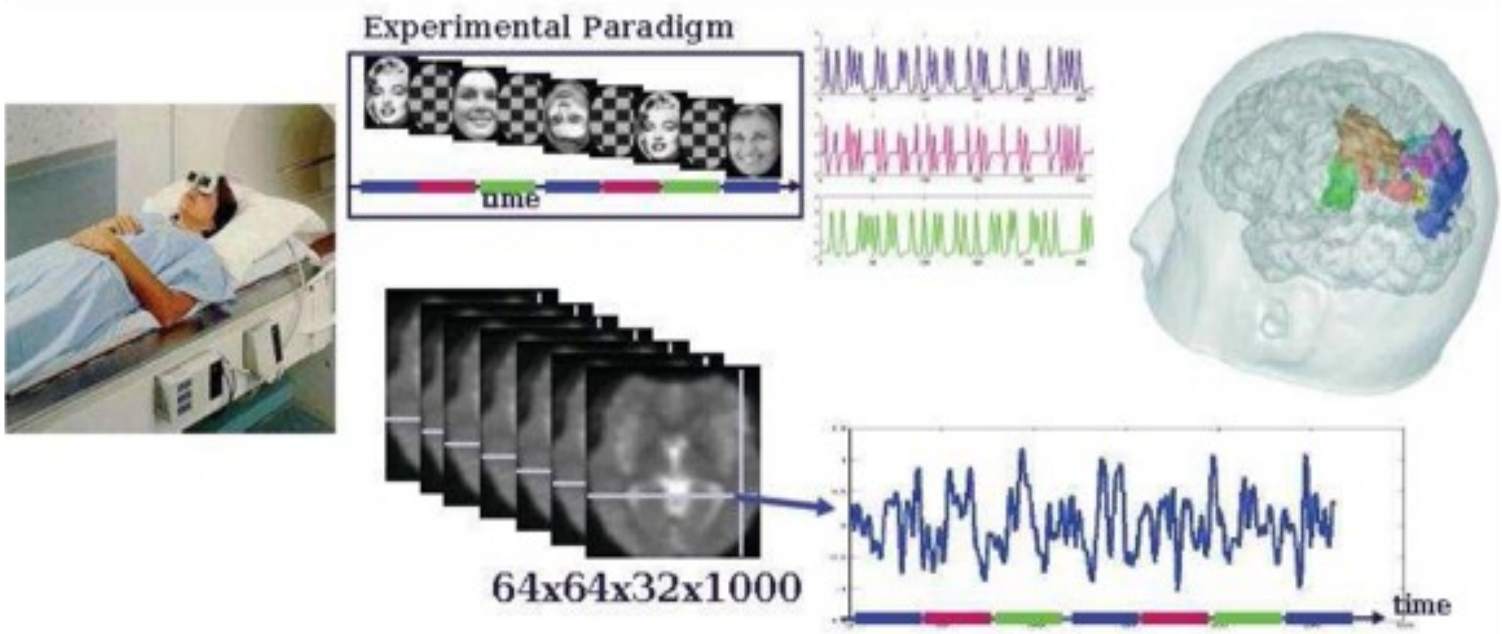
Lots of applications involve **identifying the class** of the phenomenon

(i.e. the **class** of the **time series**)

Outline

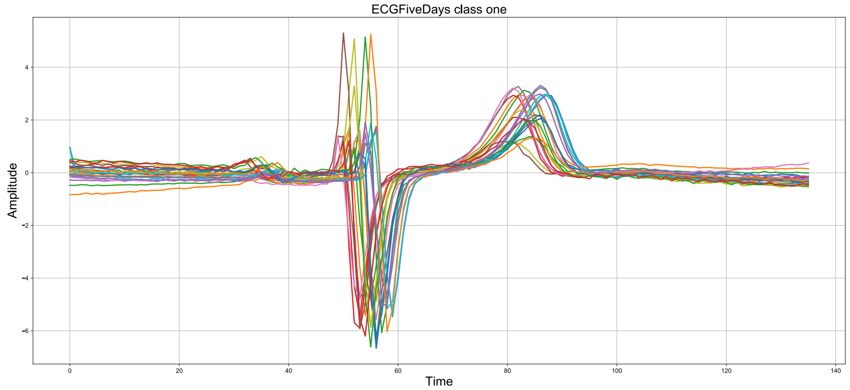
1. Introduction
2. Classification of time series: the standard setting
3. Early Classification of Times Series (ECTS)
4. A detour: the LUPI framework
5. Anticipation-based ECTS
6. Experiments and comparisons
7. Conclusions

Prosopagnosia

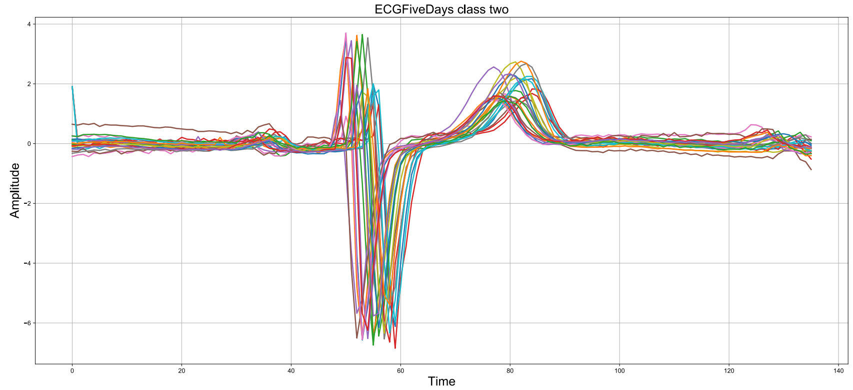


- Can we identify from the IRM measurements whether the patient suffers from **prosopagnosia** or not?

ECG signals



(c) *ECGFiveDays* class one



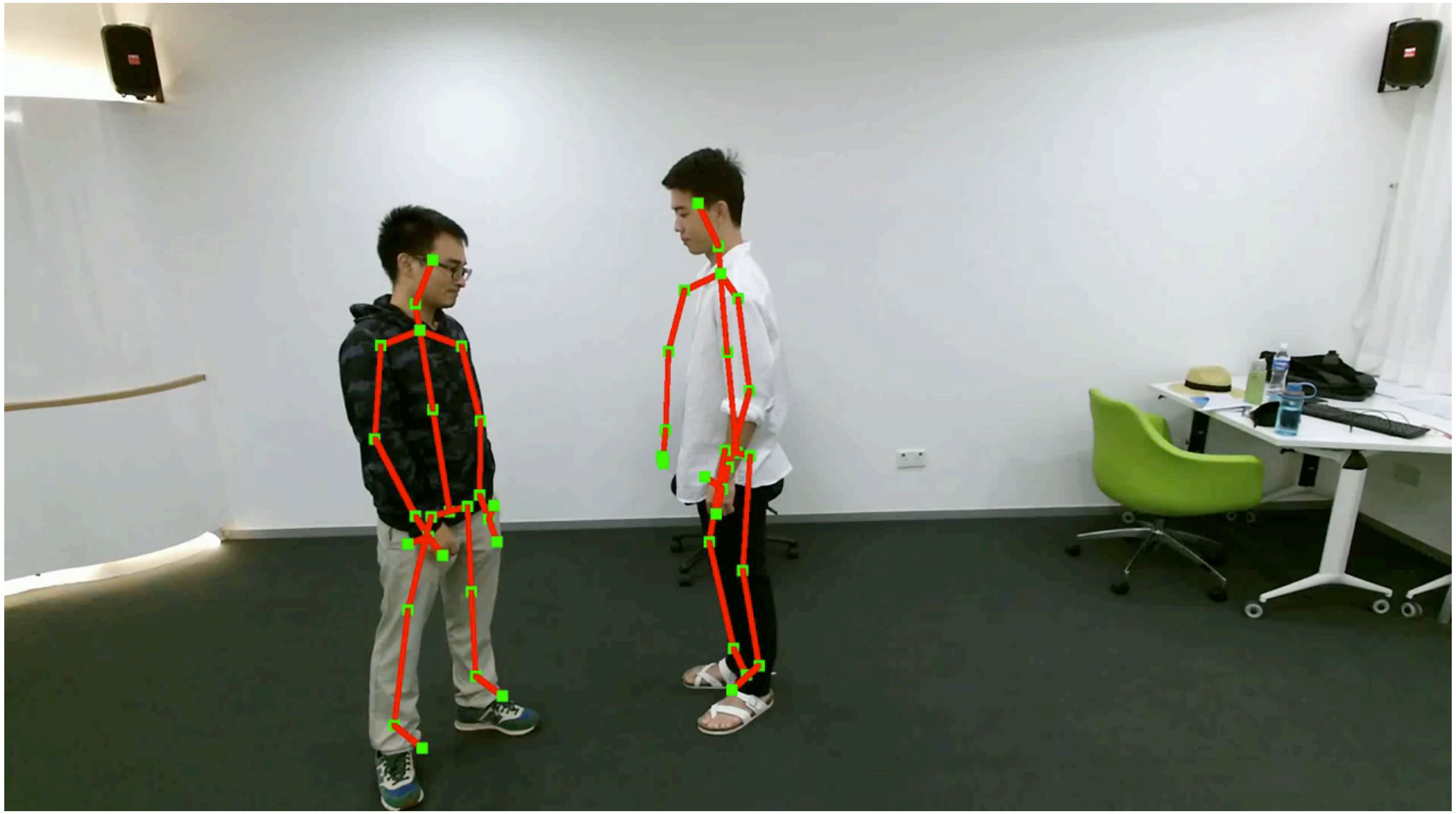
(d) *ECGFiveDays* class two

Terefe, T., Devanne, M., Weber, J., Hailemariam, D., & Forestier, G. (2023). **Estimating time series averages from latent space of multi-tasking neural networks**. *Knowledge and Information Systems*, 65(11), 4967-5004.

…

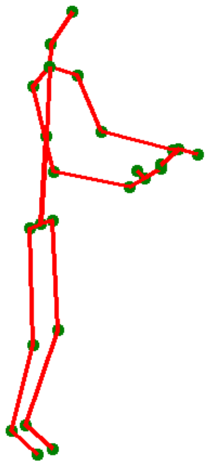
AgroParisTech

Human activity recognition



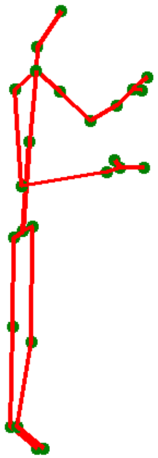
- NTU

Human activity recognition



Reading

Same class?!

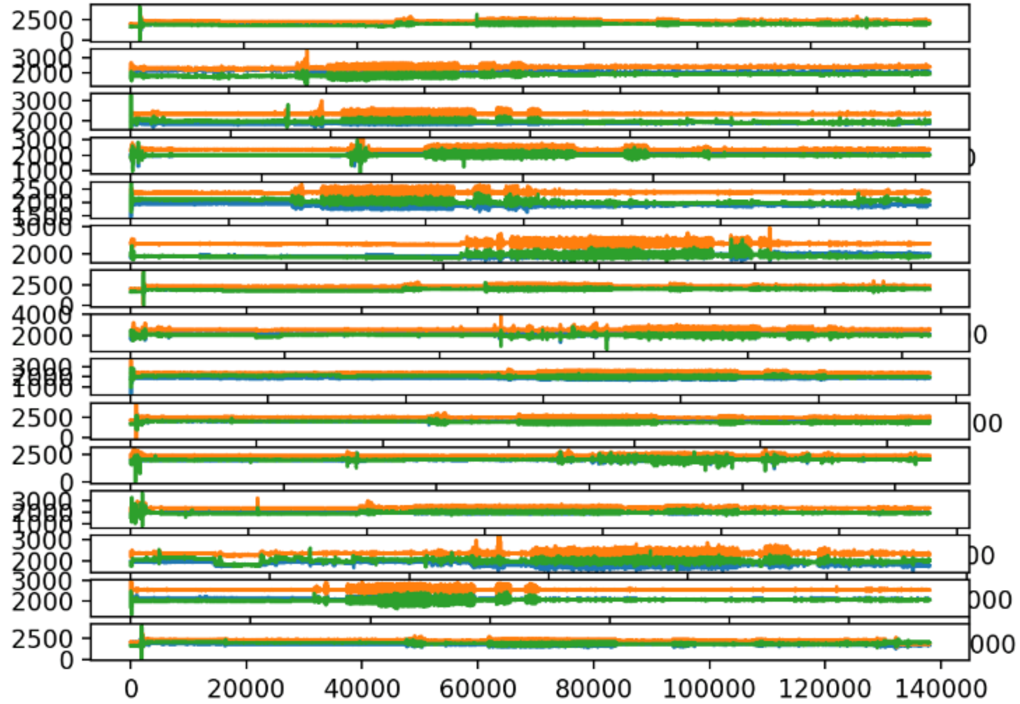


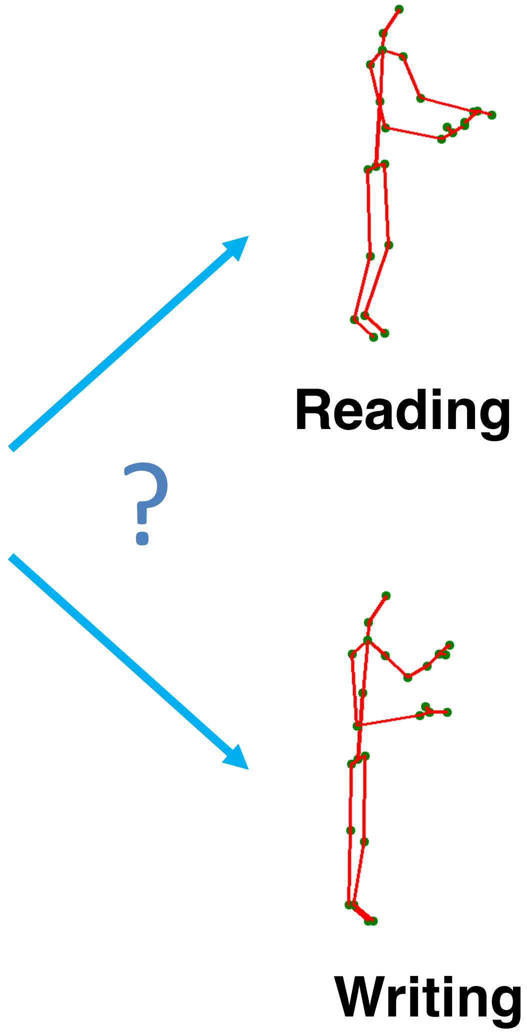
Writing

Articulated pose alone is not sufficient

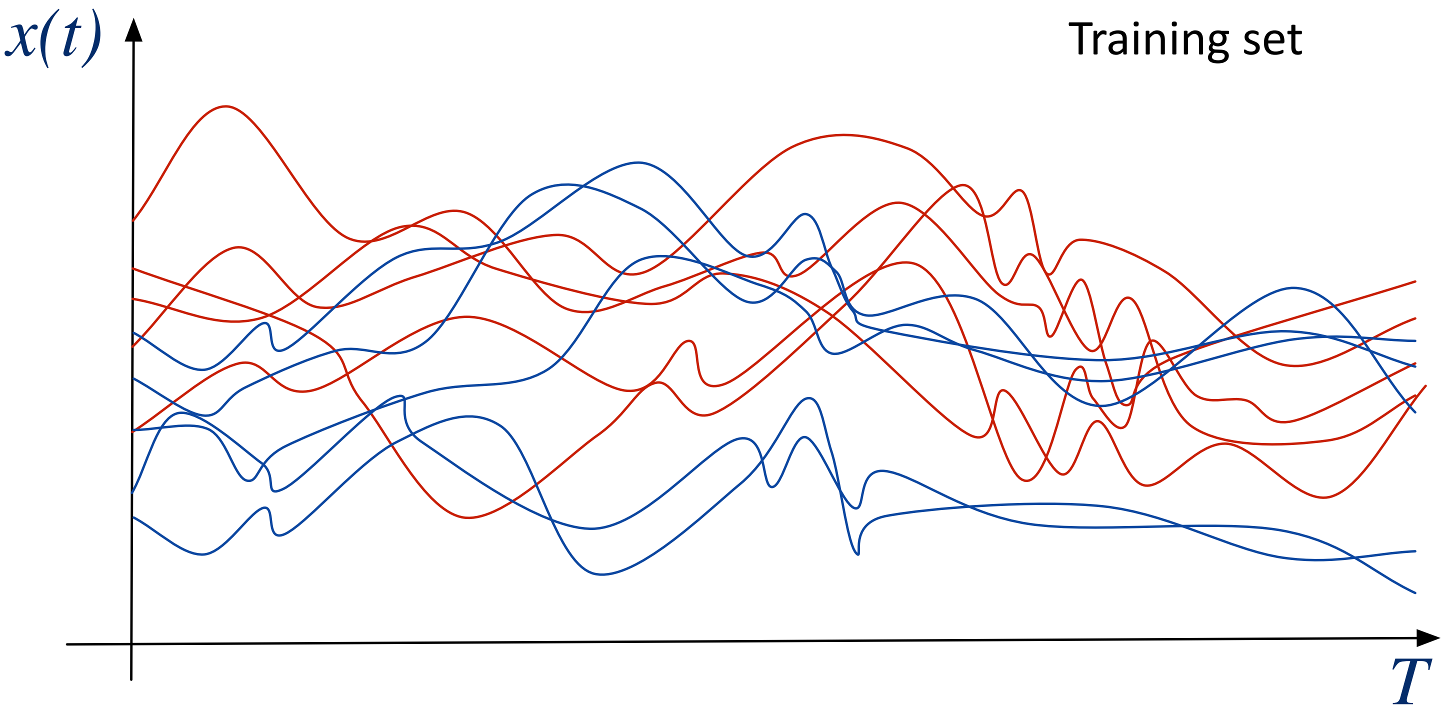
Human activity recognition

Measurements on the joints





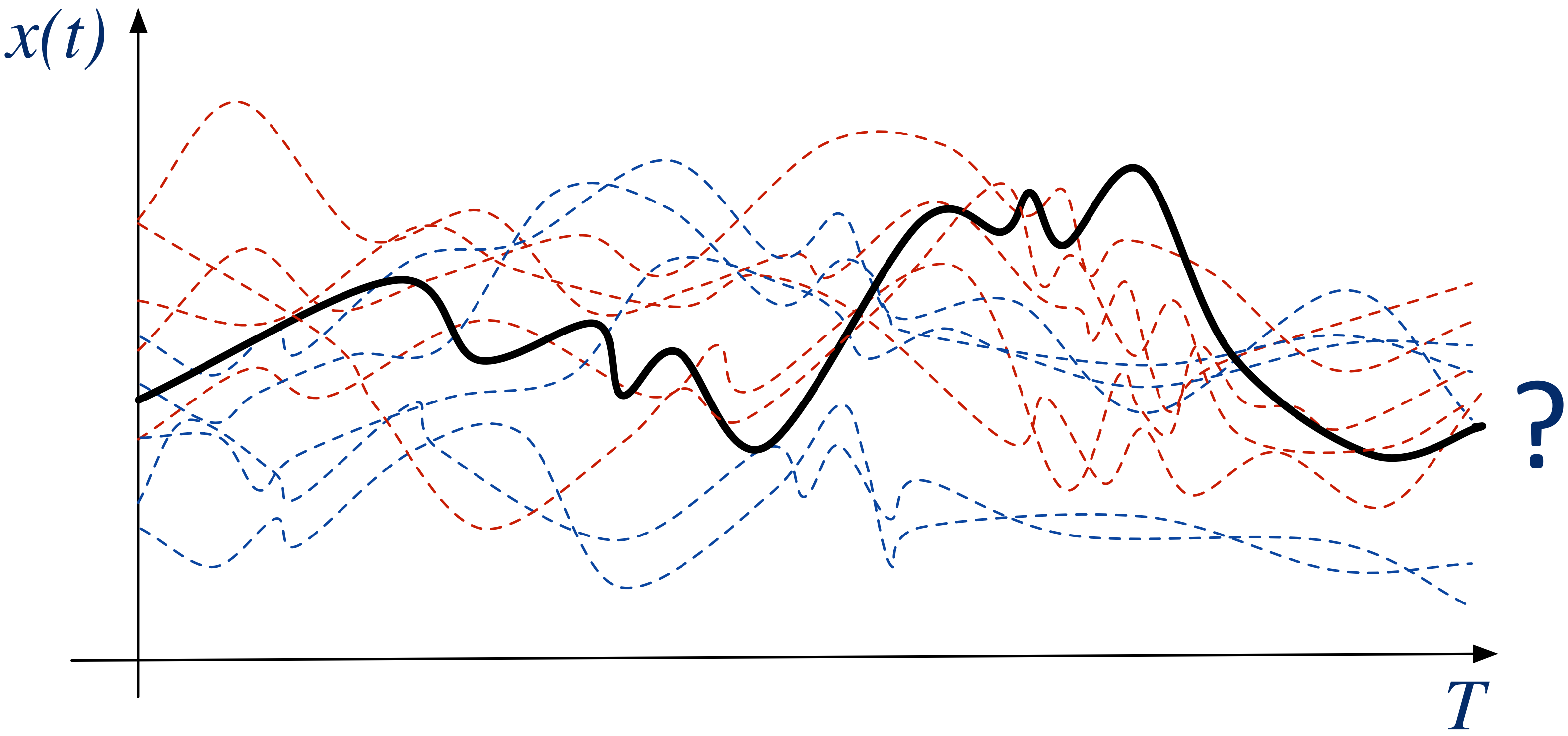
Classification of time series



- Monitoring of ***consumer actions on a web site***: will buy or not
- Monitoring of a ***patient state***: critical or not
- Evening ***electrical consumption*** *(prediction each day at 6pm)*: high or low

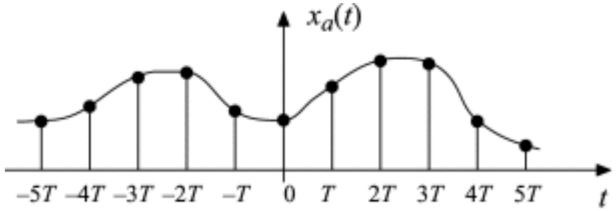
Standard classification of time series

- What is the class of the new time series $\boldsymbol{x_T}$?

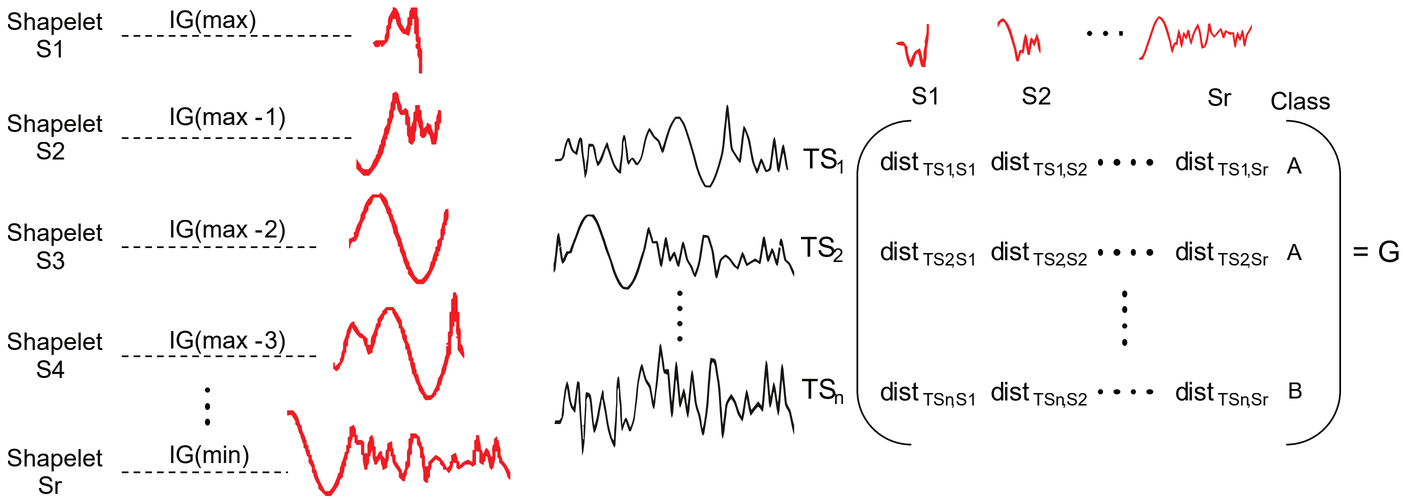


1. Representing the time series

- **Periodic sampling** of the time series



- As a set of **shapelets**



Arul, M., & Kareem, A. (2021). **Applications of shapelet transform to time series classification of earthquake, wind and wave data.** *Engineering Structures*, 228, 111564.

1. Representing the time series

- Computing **new descriptors**
  - **tsfresh** (scikit-learn)
    - Vector of 76 values
    - Independent of the length of the TS
  - Other libraries
    - **Rocket**
    - **MiniRocket**

`tsfresh.feature_extraction.feature_calculators`	This module contains the feature calculators that take time series as input and calculate the values of the feature.

The following list contains all the feature calculations supported in the current version of *tsfresh*:

`abs_energy` (x)	Returns the absolute energy of the time series which is the sum over the squared values
`absolute_maximum` (x)	Calculates the highest absolute value of the time series x.
`absolute_sum_of_changes` (x)	Returns the sum over the absolute value of consecutive changes in the series x
`agg_autocorrelation` (x, param)	Descriptive statistics on the autocorrelation of the time series.
`agg_linear_trend` (x, param)	Calculates a linear least-squares regression for values of the time series that were aggregated over chunks versus the sequence from 0 up to the number of chunks minus one.
`approximate_entropy` (x, m, r)	Implements a vectorized Approximate entropy algorithm.
`ar_coefficient` (x, param)	This feature calculator fits the unconditional maximum likelihood of an autoregressive AR(k) process.
`augmented_dickey_fuller` (x, param)	Does the time series have a unit root?
`autocorrelation` (x, lag)	Calculates the autocorrelation of the specified lag, according to the formula [1]
`benford_correlation` (x)	Useful for anomaly detection applications [1][2]. Returns the correlation from first digit distribution when

76 new descriptors (nov. 2023)

Libraries

- Libraries of functions that code time series into sets of features
  - Barandas M, Folgado D, Fernandes L, Santos S, Abreu M, Bota P, Liu H, Schultz T, Gamboa H (2020) **Tsfel**: Time series feature extraction library. SoftwareX 11:100456, https://github.com/fraunhoferportugal/tsfel

2. Classifying the time series

- **Distance**-based methods
  - E.g. kNN
  - Needs a **distance**
    - Euclidian
    - Time Warping

- **Decomposition**-based (dictionary approaches)
  - Choosing a set of **descriptors**
    - E.g. Fourier functions, shapelets, …
  - **Representing** the time series as vectors of descriptors
  - **Using** all methods based on vectors
    - SVM
    - Decision trees
    - …

- **Deep** neural networks

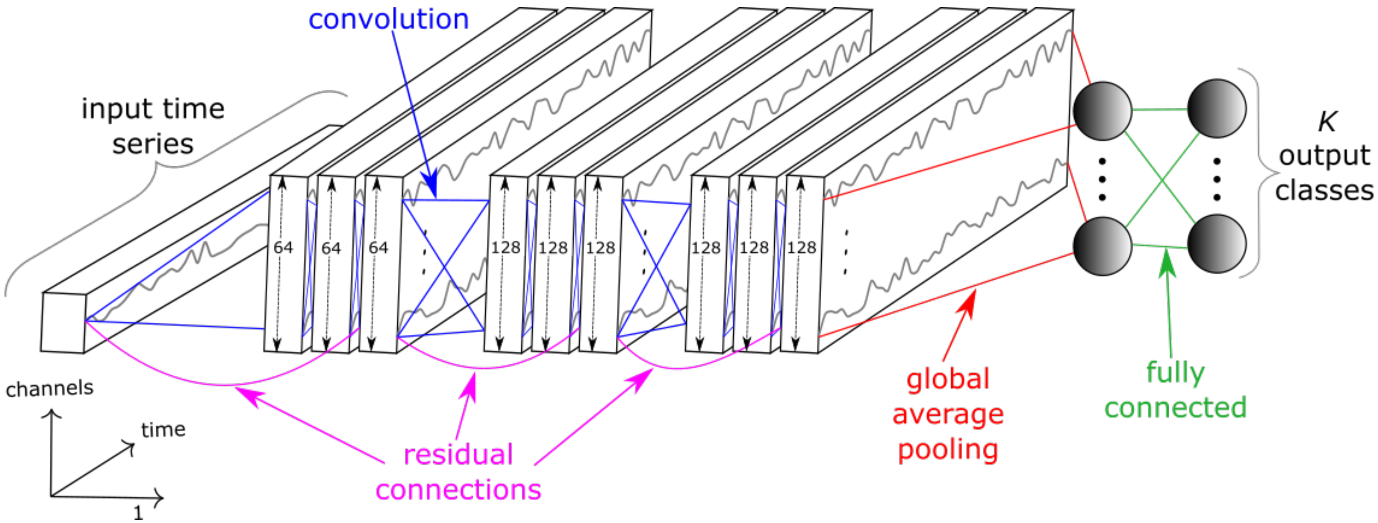


Figure 11: ***InceptionTime architecture.*** *The InceptionTime artificial neural network consists of several Inception modules with residual connections, followed by a global average pooling layer and a fully connected layer. Reproduced from (Ismail Fawaz et al, 2020).*

2. Classifying the time series

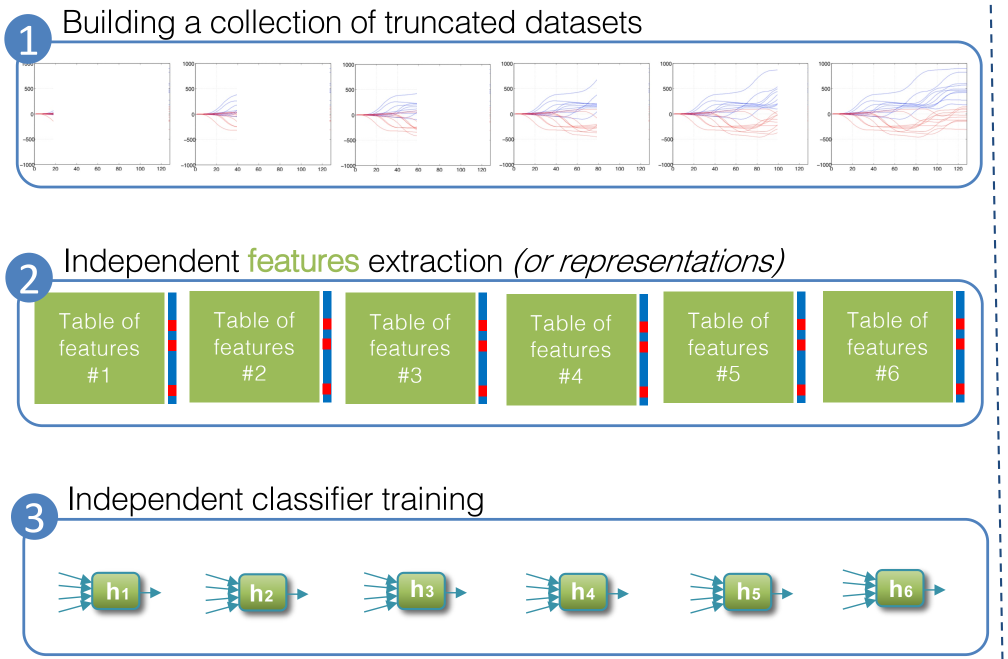
- **Single** classifiers
  - XGBoost
    - Generally a very good choice
  - Neural Networks

- **Ensemble** of classifiers
  - HIVE-COTE
    - a **collection** of classification models that each perform their own class discrimination on the data set
    - Take the **majority vote**

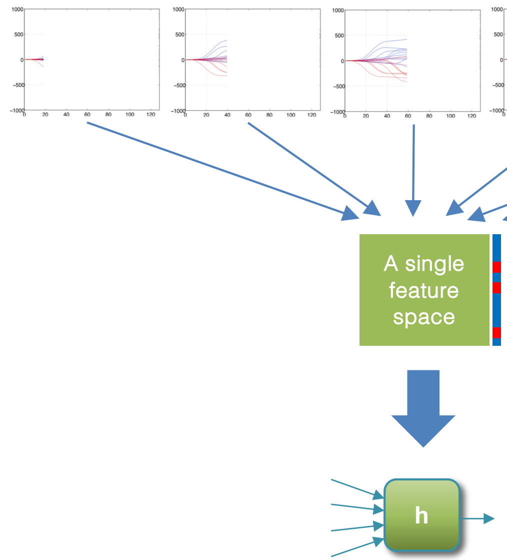
Classification function

Training a collection of time-indexed classifiers

- The most used in the literature.



Training a single classifier



- Specific feature engineering
- Or learning a representation (deep)

- Training a single classifier

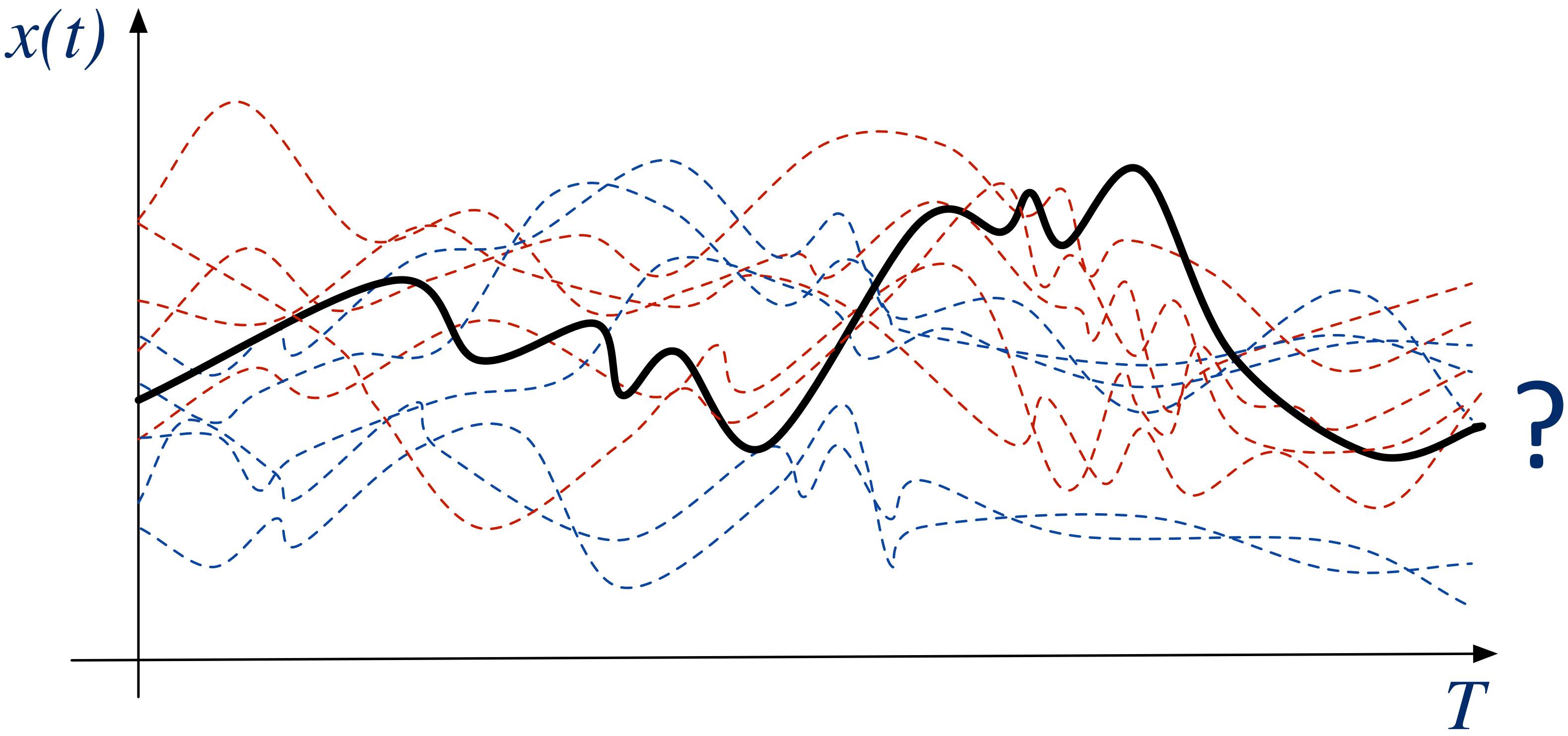
AgroParisTech

Outline

1. Introduction
2. Classification of time series: the standard setting
3. Early Classification of Times Series (ECTS)
4. A detour: the LUPI framework
5. Anticipation-based ECTS
6. Experiments and comparisons
7. Conclusions

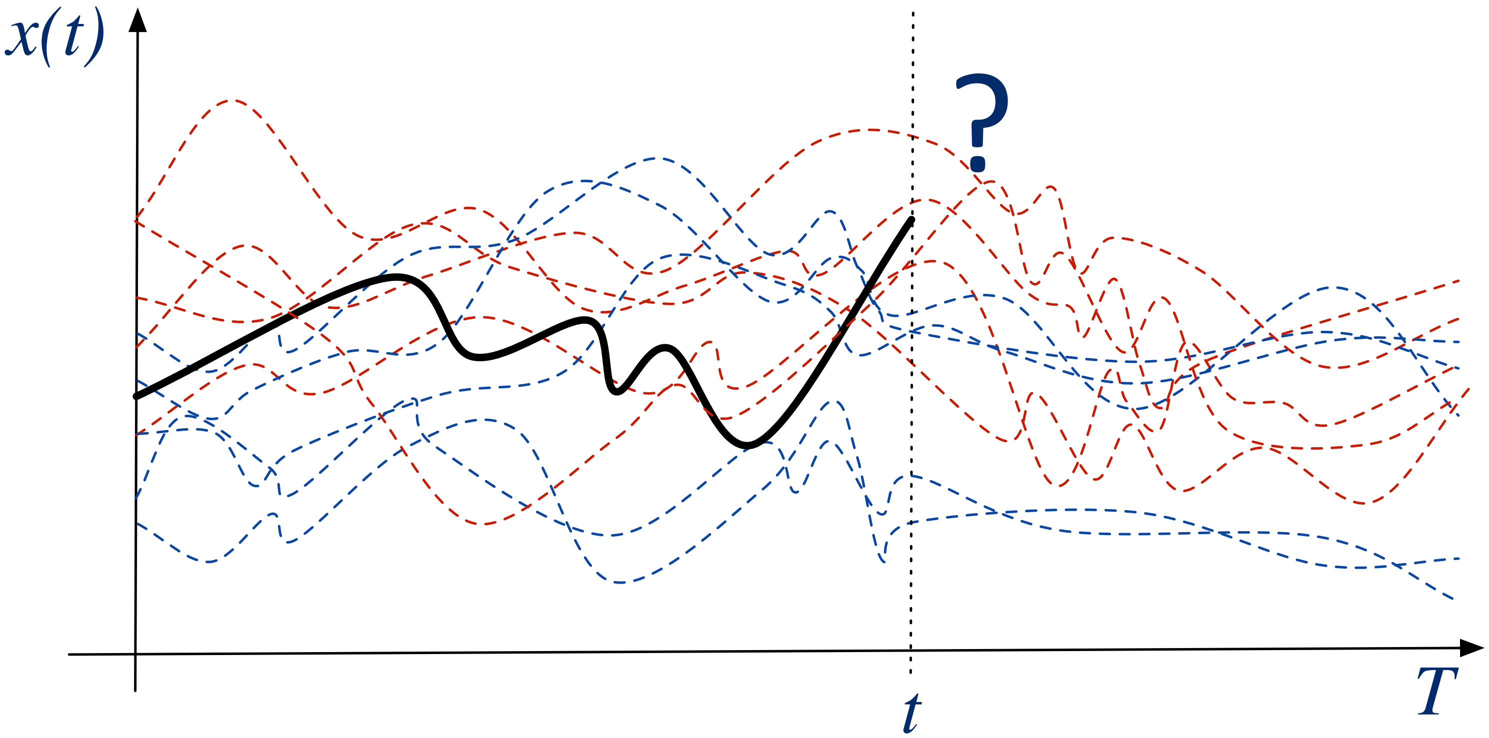
Standard classification of time series

- What is the class of the new time series $\boldsymbol{x}_T$?



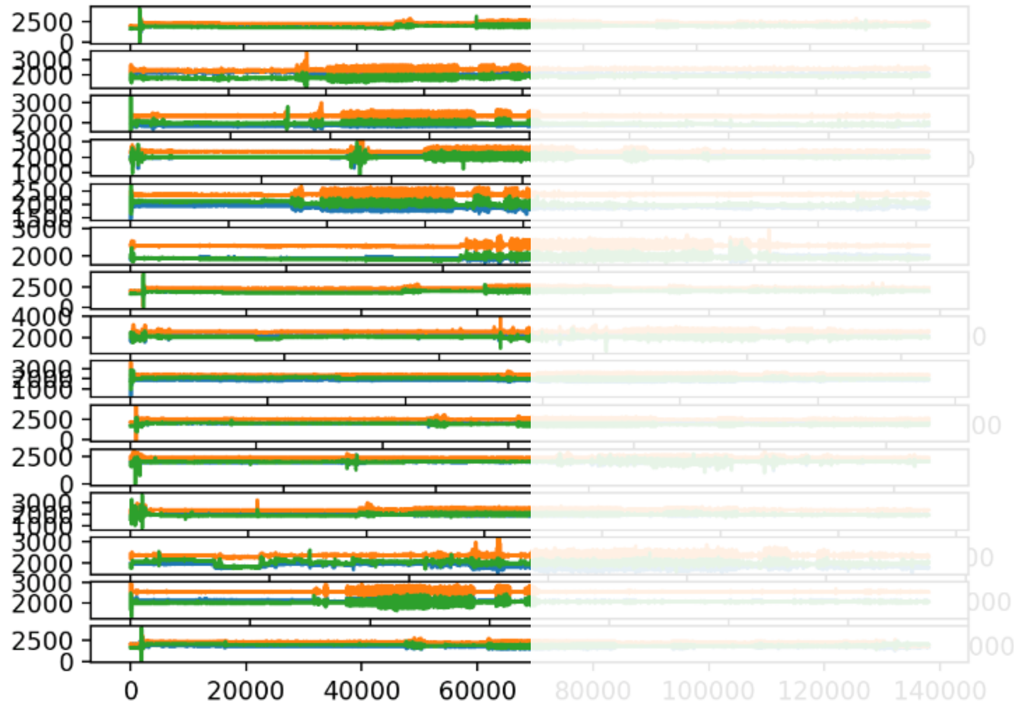
Early classification of time series

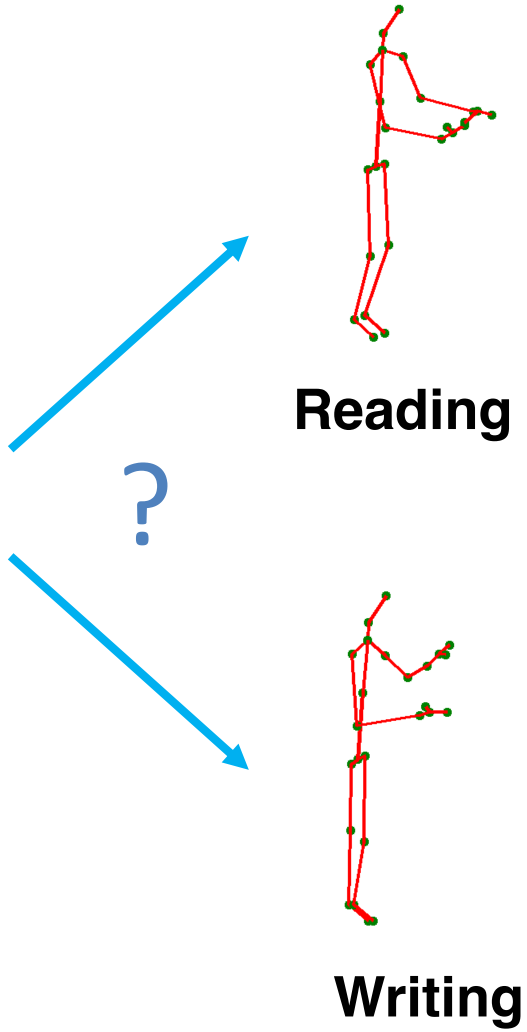
- What is the class of the new **incomplete** time series $\boldsymbol{x_t}$?



Early human activity recognition

Measurements on the joints





Applications

- Decide **chirurgical operation**
  - Do not operate if not necessary
  - But, the earliest the decision, the better the outcome

Applications

- Decide **chirurgical operation**
  - Do not operate if not necessary
  - But, the earliest the decision, the better the outcome
- Predictive **maintenance**
  - Early maintenance is unnecessarily costly
  - But, waiting too long can be very costly

Applications

- Decide **chirurgical operation**
  - Do not operate if not necessary
  - But, the earliest the decision, the better the outcome

- Predictive **maintenance**
  - Early maintenance is unnecessarily costly
  - But, waiting too long can be very costly

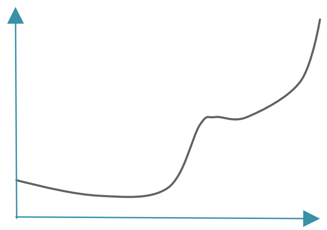
- Decide operation only with enough certainty
- Do not wait too long before taking decision

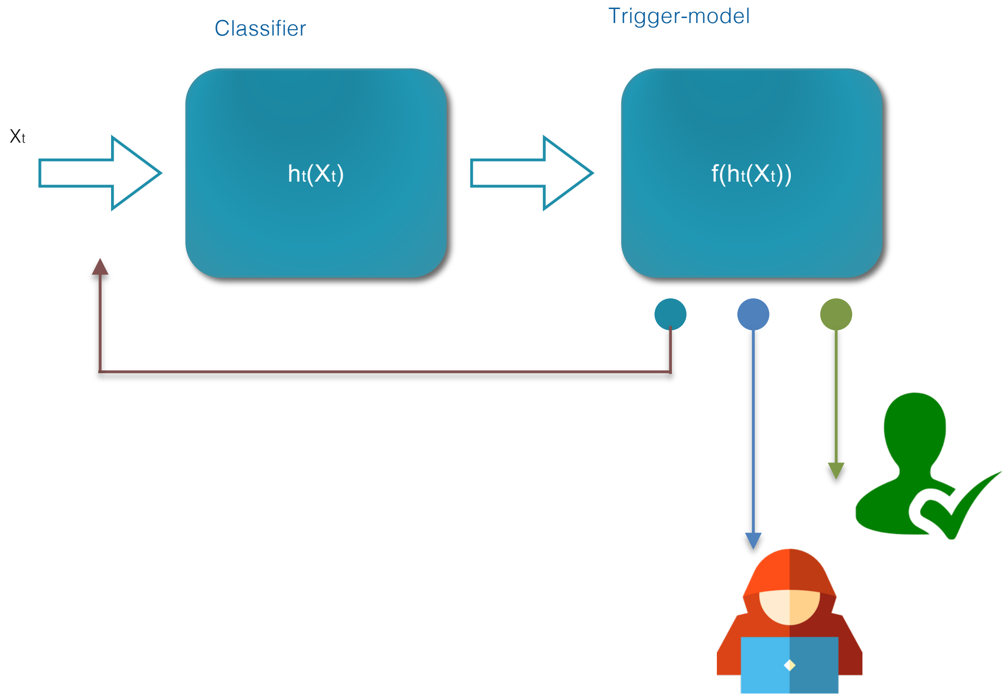
New decision problems: early classification

- A **trade-off**
  - Classification **performance** (better if $t \nearrow$)
  - Cost of **delaying** prediction (better if $t \searrow$)

Formalization

How would you approach the problem?

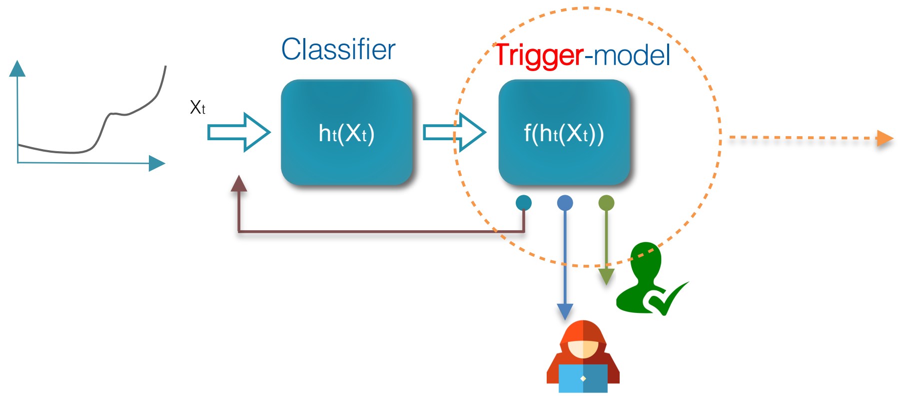



Classifier
Trigger-model
Xt
ht(Xt)
f(ht(Xt))

- …

A natural approach: confidence-based

1. Input = x_t at time step t
2. Compute the **confidence** of the prediction $h(x_t)$
3. Make a prediction when **confidence** > threshold



$$f\left(h_t(\mathbf{x}_t)\right) = \begin{cases} 0 & \text{if } \max_{y \in Y}(\hat{p}(y|X_t)) \leq \gamma \\ 1 & \text{otherwise} \end{cases}$$

A natural approach: confidence-based

Question: **How** to set the threshold?

…

AgroParisTech

A natural approach: confidence-based

- The threshold is a parameter that is **optimized** on a training set

Experimental setting

- Data sets
  - The UCR archive for time series classification: 77 data sets
- Classifier
  - MiniRocket
- Performance

Number of test data sets

$$AvgCost = \frac{1}{M} \sum_{i=1}^{M} \underbrace{C_m(\hat{y}_i|y_i)}_{\text{Misclassification cost}} + \underbrace{C_d(\hat{t}_i)}_{\text{Delay cost}}$$

Misclassification cost **Delay** cost

$$C_m(\hat{y}|y) : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R} \qquad C_d(t) : \mathbb{R}^+ \rightarrow \mathbb{R}$$

- **Misclassification** cost

	p_+	p_-
True class / **Predicted** class	+	-
+	0	FP **cost = 10**
-	FN **cost = 5**	0

Cost matrix

- **Misclassification** cost

True class / Predicted class	+ (p_+)	- (p_-)
+	0	FP **cost = 10**
-	FN **cost = 5**	0

Cost matrix

- **Expected** misclassification cost

True class / Predicted class	+	-
+	TP **= 0.82**	FP **= 0.23**
-	FN **= 0.18**	TN **= 0.77**

Confusion matrix of the classifier at time *t*

p_+ p_-

- **Misclassification** cost

True class / Predicted class	+	-
+	0	**FP cost = 10**
-	**FN cost = 5**	0

Cost matrix

- **Expected** misclassification cost

True class / Predicted class	+	-
+	**TP = 0.82**	**FP = 0.23**
-	**FN = 0.18**	**TN = 0.77**

Confusion matrix of the classifier at time *t*

$$\mathbb{E}^{t}_{(\hat{y},y)\in\mathcal{Y}^2}\left[\mathrm{C}_m(\hat{y}_t|y)\right]$$

$$= (p_+ \times 0.82 \times 0) + (p_- \times 0.23 \times 10)$$

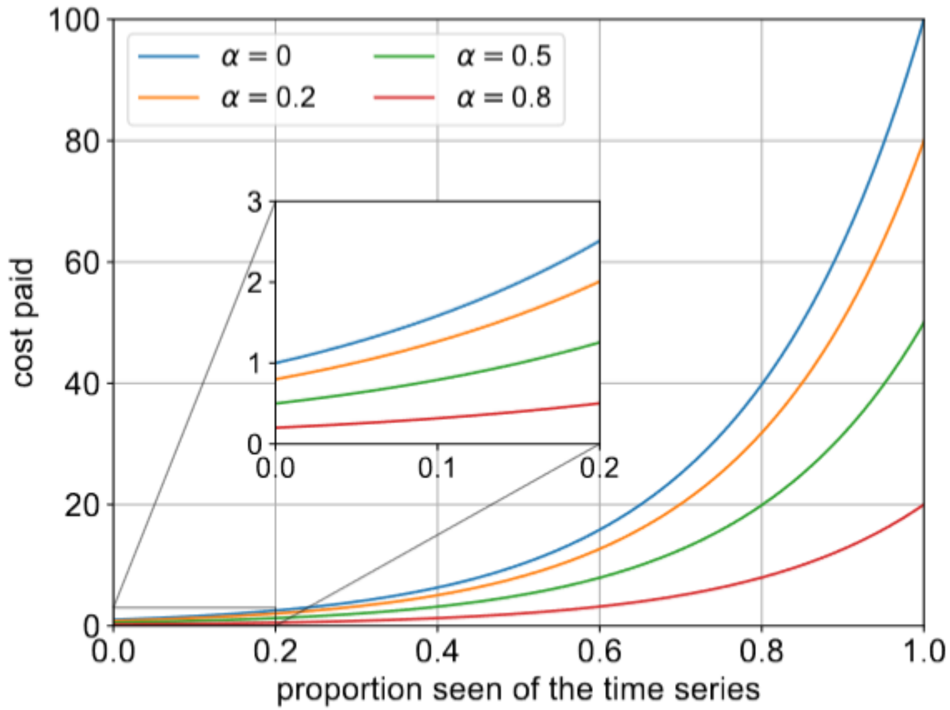
$$+ (p_+ \times 0.18 \times 5) + (p_- \times 0.77 \times 0)$$

$$= 3.2$$

Delay cost $C_d(t)$

Often considered as **linear** with time

Here, **exponential** costs



…

A natural approach: confidence-based

- The threshold is a parameter that is **optimized** on a training set
  - By **minimizing the AvgCost** on the training data sets

Number of test data sets

$$AvgCost = \frac{1}{M} \sum_{i=1}^{M} \mathrm{C}_m(\hat{y}_i | y_i) + \mathrm{C}_d(\hat{t}_i)$$

Misclassification cost

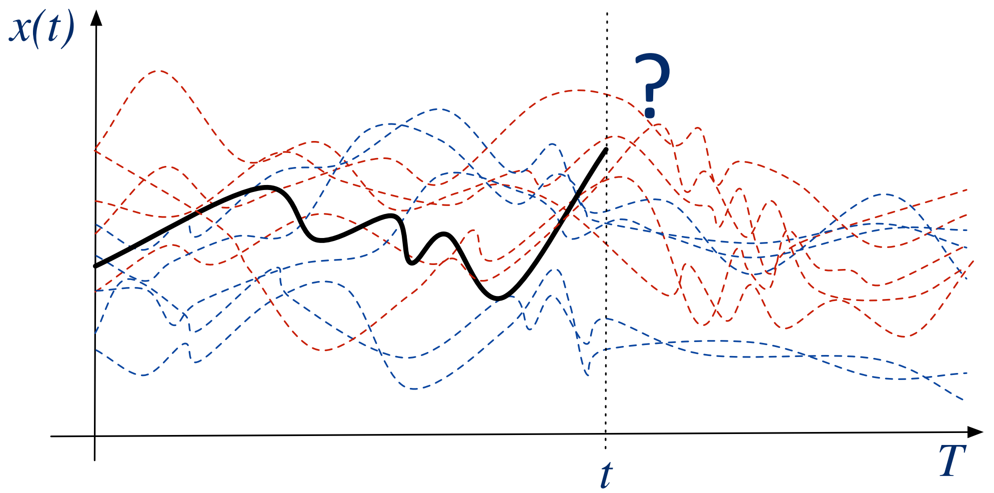
Delay cost

AgroParisTech

Optimal decision time

- The **time** optimizing the tradeoff for a given time series *i*

$$t_i^\star \;=\; \underset{t\in[1,T]}{\text{ArgMin}} \left\{ \mathrm{C}_m(\hat{y}_i|y_i) + \mathrm{C}_d(\hat{t}_i) \right\}$$



A more **sophisticated** confidence-based

The "stopping rule"

$$f\left(h_t(\mathbf{x}_t)\right) = \begin{cases} 0 & \text{if } \gamma_1 \boxed{p_1} + \gamma_2 \boxed{p_2} + \gamma_3 \boxed{\frac{t}{T}} \leq 0 \\ 1 & \text{otherwise} \end{cases}$$

Reliability

- the largest posterior probability estimated by the classifier
- the difference between the two largest posterior probabilities

Earliness

- the proportion of the incoming time series that is available at time *« t »*

Mori, U., Mendiburu, A., Dasgupta, S., & Lozano, J. A. (2017). **Early classification of time series by simultaneously optimizing the accuracy and earliness**. *IEEE transactions on neural networks and learning systems*, 29(10), 4569-4578.

Limits …

- … of the confidence-based methods

?

AgroParisTech

Limits …

- … of the confidence-based methods?

  - **Do not** take into account the costs !!!

Only indirectly in the optimization process

Limits …

Question: **Can** we find a decision

criterion that **uses** the costs?

…

Cost-based methods

- For each time step *t*, compute the expected cost

Incoming time series at *t*

$$f(\mathbf{x}_t) = \mathbb{E}^{\,t}_{(\hat{y},y)\in\mathcal{Y}^2}\left[\mathrm{C}_m(\hat{y}|y)|\mathbf{x}_t\right] + \mathrm{C}_d(t)$$

Delay cost at *t*

Expectancy of the **misclassification cost** making the prediction $\hat{y}$ at *t*

$$= \sum_{y\in\mathcal{Y}} P_t(y|\mathbf{x}_t) \sum_{\hat{y}\in\mathcal{Y}} P_t(\hat{y}|y,\mathbf{x}_t)\,\mathrm{C}_m(\hat{y}|y) + \mathrm{C}_d(t)$$

Expected misclassification cost for a given target *y*

- Okay, you have an expected cost at **each time step** *t*
  - And you want the time t^* where it is minimal

But when would you **stop**?

- And make a **prediction**

AgroParisTech

Outline

1. Introduction
2. Classification of time series: the standard setting
3. Early Classification of Times Series (ECTS)
4. A detour: the LUPI framework
5. Anticipation-based ECTS
6. Experiments and comparisons
7. Conclusions

Learning Using Privileged Information

Inspired by learning at school

- The goal is to learn a function $h : \mathbf{x} \in \mathcal{X} \rightarrow y \in \{-1, +1\}$

- Suppose that at learning time there is more available information than at test time

$$\mathcal{S}^* = \{(\mathbf{x}_i, \mathbf{x}_i^*, y_i)\}_{1 \leq i \leq m}$$

$$\underbrace{\mathbf{x}_i, \mathbf{x}_i^*}_{\mathcal{X}'}$$

- **Can we then improve** the **generalization** performance wrt. the one obtained with *S* only?

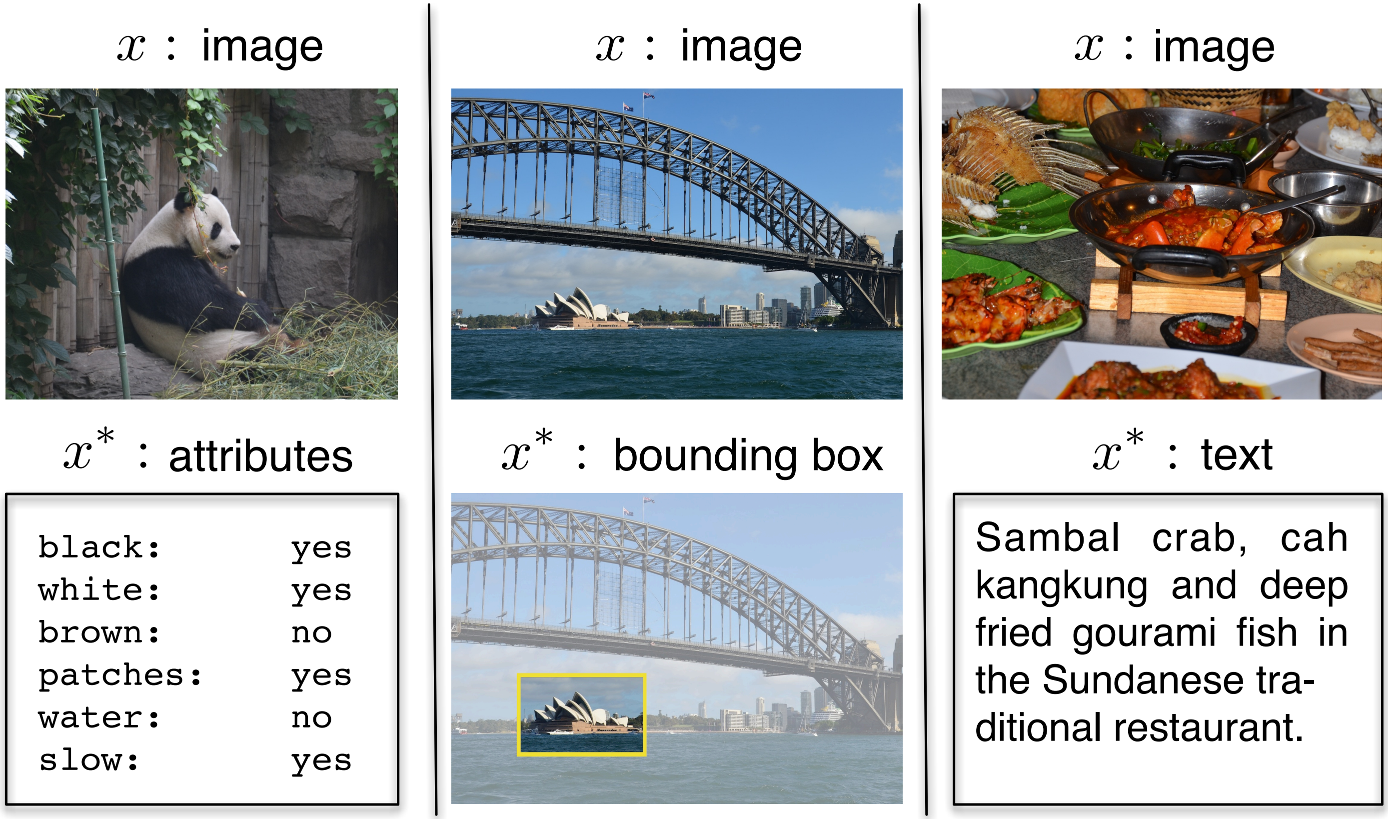
V. Vapnik and A. Vashist (2009) "A new learning paradigm: Learning using privileged information". *Neural Networks*, vol. 22, no. 5, pp. 544–557, 2009

Can you imagine **applications** where privileged information

could be available at *training* time (and not at *testing* time)?

Learning Using Privileged Information

Illustration in computer vision



V. Sharmanska, N. Quadrianto, and Ch. Lamper (2014) “Learning to transfer privileged information”. *ArXiv preprint arXiv:1410.0389,* 2014

Two general approaches to LUPI

- **Learning** a hypothesis in the "augmented" input space

$$h' : \mathcal{X}' \rightarrow y$$

$$\mathcal{X}' = \mathcal{X} \cup \mathcal{X}^{\star}$$

- **Testing**

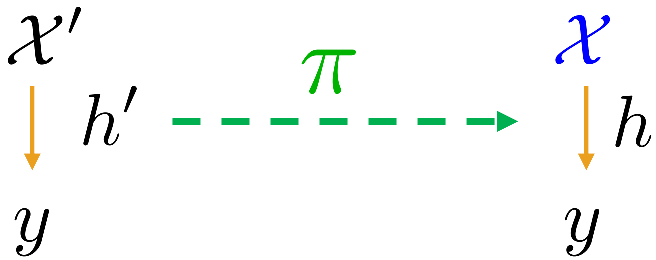
1. 1st approach: learn to "**complete**" the **description** in $\mathcal{X}$

   then use *h'*

$$\mathcal{X} \rightarrow \mathcal{X}^{\star}$$

$$h' : \mathcal{X}' \rightarrow y$$

2. 2nd approach: **project** back *h'*, the **learnt hypothesis**



Two general approaches to LUPI

- **Learning** a hypothesis in the "augmented" input space $h' : \mathcal{X}' \rightarrow y$

  $\mathcal{X}' = \mathcal{X} \cup \mathcal{X}^{\star}$

- **Testing**

  1. 1st approach: learn to "**complete**" the **description** in $\mathcal{X}$

     then use *h'*

     $\mathcal{X} \rightarrow \mathcal{X}^{\star}$

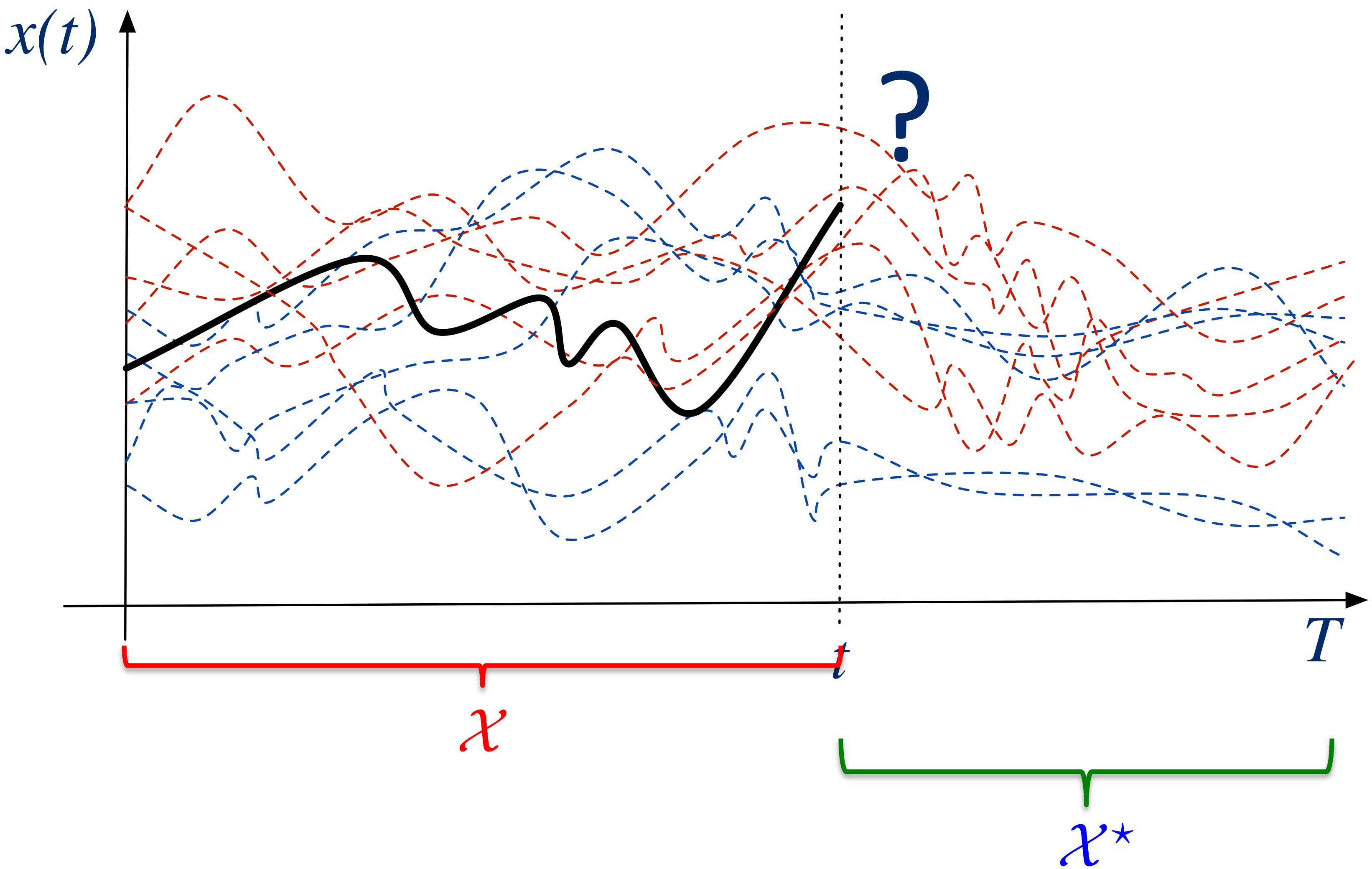
     $h' : \mathcal{X}' \rightarrow y$

Outline

1. Introduction
2. Classification of time series: the standard setting
3. Early Classification of Times Series (ECTS)
4. A detour: the LUPI framework
5. Anticipation-based ECTS
6. Experiments and comparisons
7. Conclusions

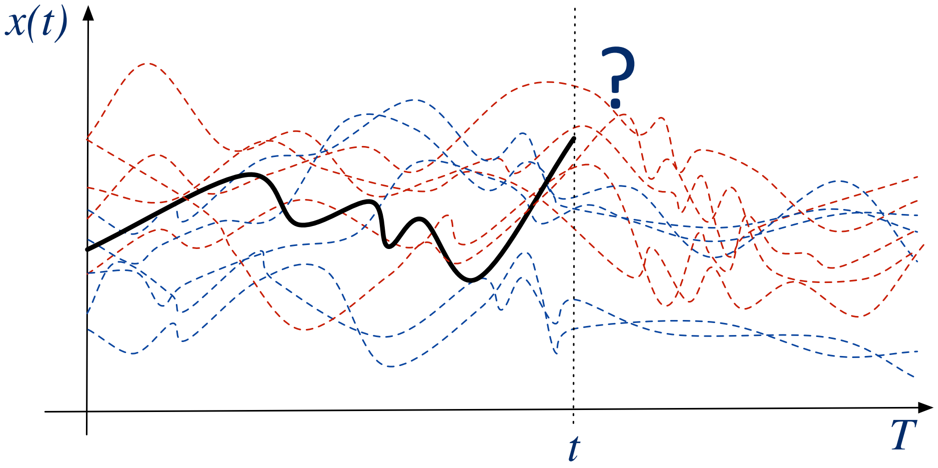
Early classification of time series

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



Early classification of time series

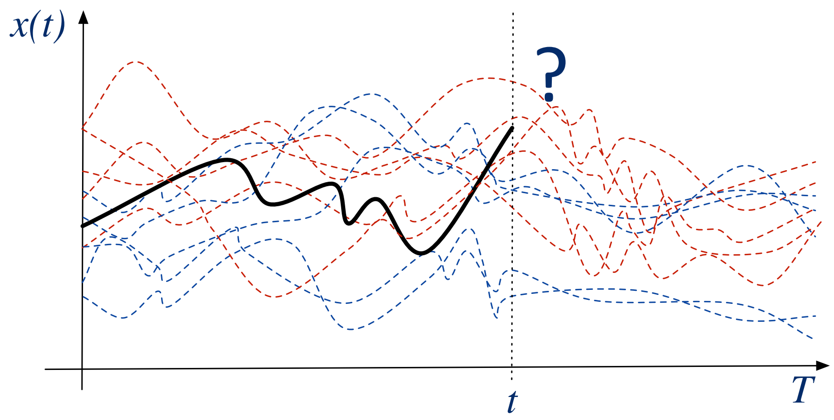
- What is the class of the new **incomplete** time series $\boldsymbol{x_t}$?



- A LUPI framework

Early classification and LUPI

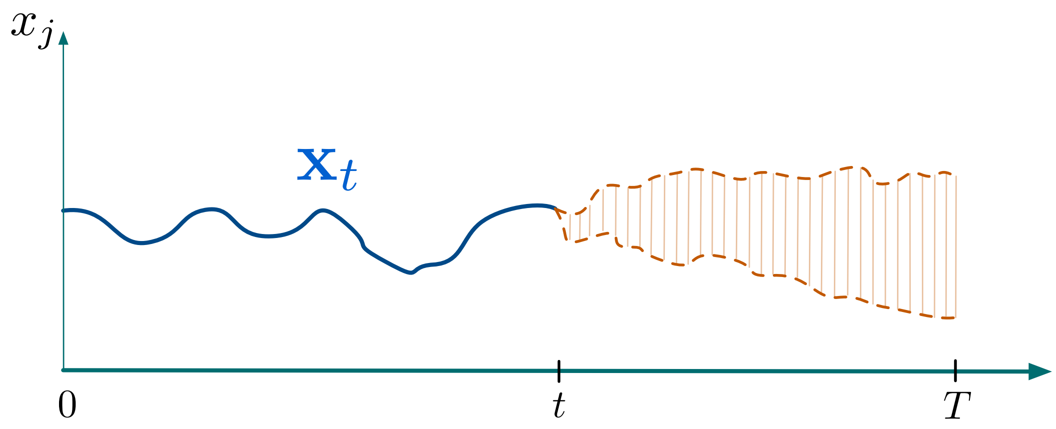
- This is a LUPI setting



How to take advantage of this?

Principle

Compute an "envelope" of the **likely continuations** of the time series



- At time t
  - Compute the **expected cost** for each **future time step** $t+\tau$ (until T)
  - **If** at any future time $t+\tau$, the expected cost is lower than the current one, **defer** decision

Cost-based methods

- For each time step *t*, compute the expected cost

Incoming time series at *t*

$$f(\mathbf{x}_t) = \mathbb{E}^{\,t}_{(\hat{y},y)\in\mathcal{Y}^2}\left[\mathrm{C}_m(\hat{y}|y)|\mathbf{x}_t\right] + \mathrm{C}_d(t)$$

Expectancy of the **misclassification cost** making the prediction $\hat{y}$ at *t*

Delay cost at *t*

$$= \sum_{y\in\mathcal{Y}} P_t(y|\mathbf{x}_t) \sum_{\hat{y}\in\mathcal{Y}} P_t(\hat{y}|y,\mathbf{x}_t)\,\mathrm{C}_m(\hat{y}|y) + \mathrm{C}_d(t)$$

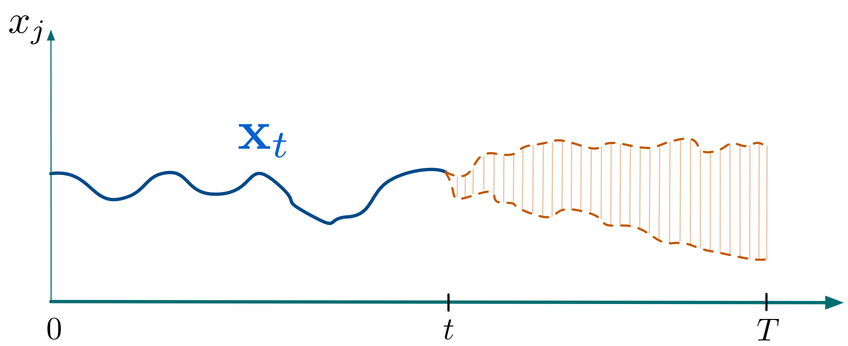
Formalization

$$
\begin{aligned}
f_{\tau}(\mathbf{x}_t) &= \mathbb{E}^{\,t+\tau}_{(\hat{y},y)\in\mathcal{Y}^2}\left[\mathrm{C}_m(\hat{y}|y)\right] \;+\; \mathrm{C}_d(t+\tau) \\
&= \sum_{y\in\mathcal{Y}} P_t(y|\mathbf{x}_t) \underbrace{\int_{\mathbf{x}_{t+\tau}\in\mathcal{X}} P(\mathbf{x}_{t+\tau}|\mathbf{x}_t)}_{LUPI} \sum_{\hat{y}_{t+\tau}\in\mathcal{Y}} P_{t+\tau}(\hat{y}_{t+\tau}|y,\mathbf{x}_{t+\tau})\, \mathrm{C}_m(\hat{y}_{t+\tau}|y)\, \mathrm{d}\mathbf{x}_{t+\tau} \;+\; \mathrm{C}_d(t+\tau)
\end{aligned}
$$

Probability of class *y* given $\mathbf{x}_t$

Over all possible continuations of $\mathbf{x}_t$

Characteristics of the classifier at time t+τ



Formalization

$$
\begin{aligned}
f_{\tau}(\mathbf{x}_t) &= \mathbb{E}^{\,t+\tau}_{(\hat{y},y)\in\mathcal{Y}^2}\left[\mathrm{C}_m(\hat{y}|y)\right] \;+\; \mathrm{C}_d(t+\tau) \\
&= \sum_{y\in\mathcal{Y}} P_t(y|\mathbf{x}_t) \underbrace{\int_{\mathbf{x}_{t+\tau}\in\mathcal{X}} P(\mathbf{x}_{t+\tau}|\mathbf{x}_t)}_{LUPI} \sum_{\hat{y}_{t+\tau}\in\mathcal{Y}} P_{t+\tau}(\hat{y}_{t+\tau}|y,\mathbf{x}_{t+\tau})\, \mathrm{C}_m(\hat{y}_{t+\tau}|y)\, \mathrm{d}\mathbf{x}_{t+\tau} \;+\; \mathrm{C}_d(t+\tau)
\end{aligned}
$$

Probability of class *y* given $\mathbf{x}_t$

Over all possible continuations of $\mathbf{x}_t$

Characteristics of the classifier at time t+τ

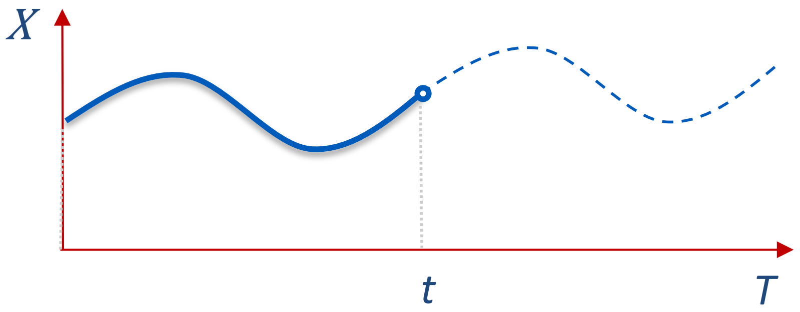
- A rather daunting equation
  - But there are ways to **simplify** it
    - Depending on how to **estimate the likely continuations**
      - Economy_K
      - Economy_γ

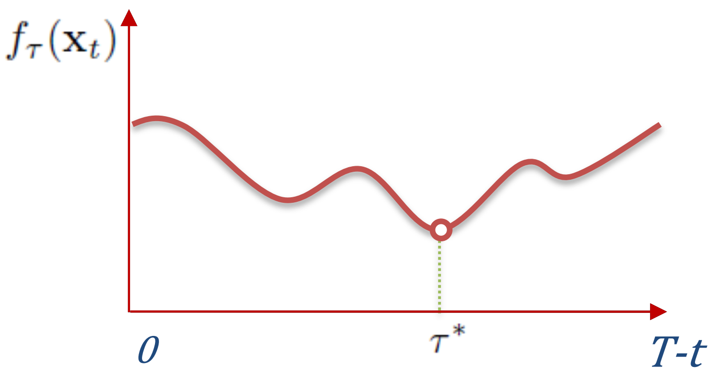
How it works

Achenchabe, Y., Bondu, A., Cornuéjols, A., & Dachraoui, A. (2021). **Early classification of time series: Cost-based optimization criterion and algorithms**. *Machine Learning*, 110(6), 1481-1504.

A non myopic decision process

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



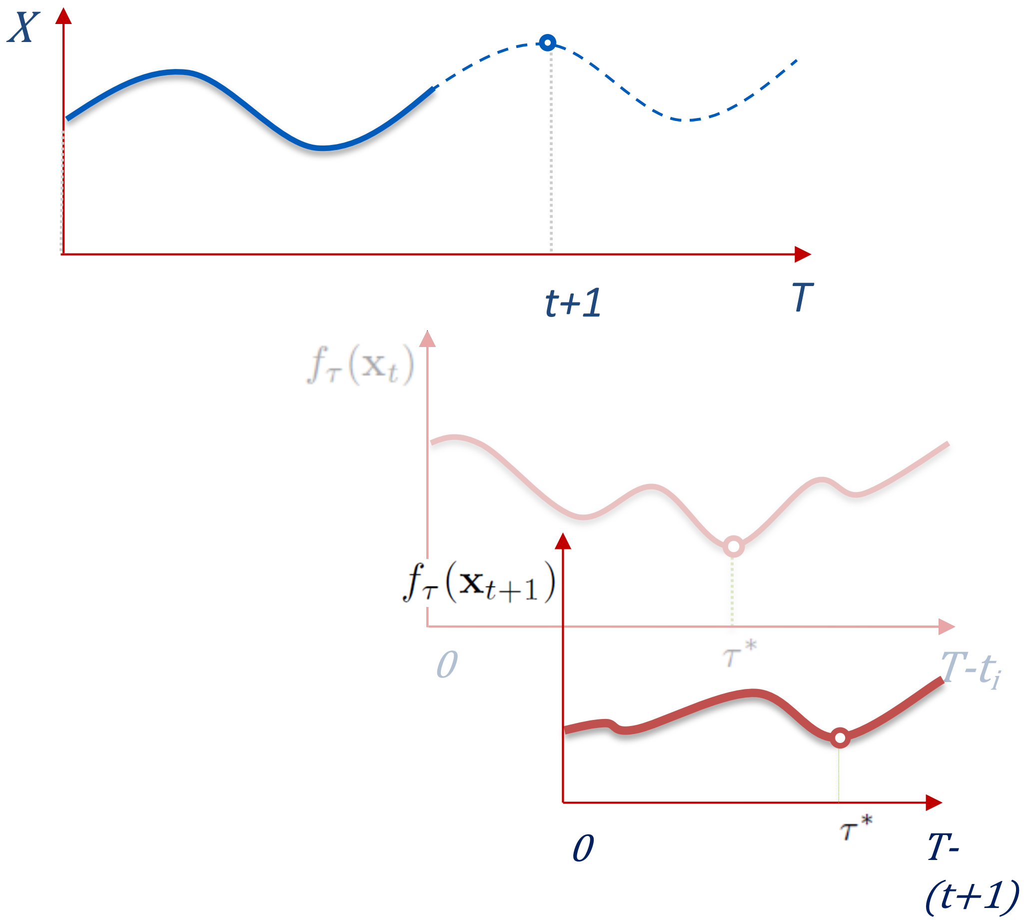


→ Continue monitoring

A non myopic decision process

- Optimal estimated time relative to current time *t*

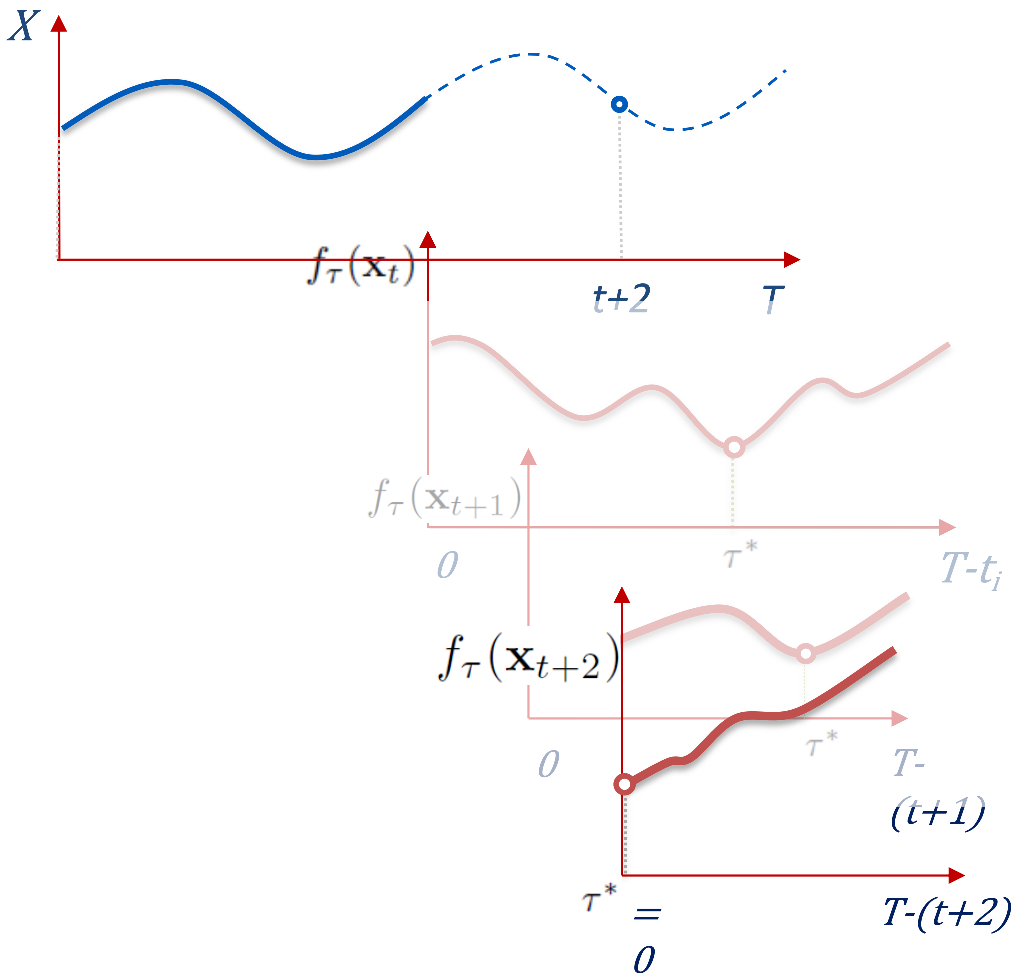
$$\tau^* = \underset{\tau \in \{0,\ldots,T-t\}}{\text{ArgMin}} \; f_\tau(\mathbf{x}_t)$$



→ Continue monitoring

A non myopic decision process

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



→ Make prediction

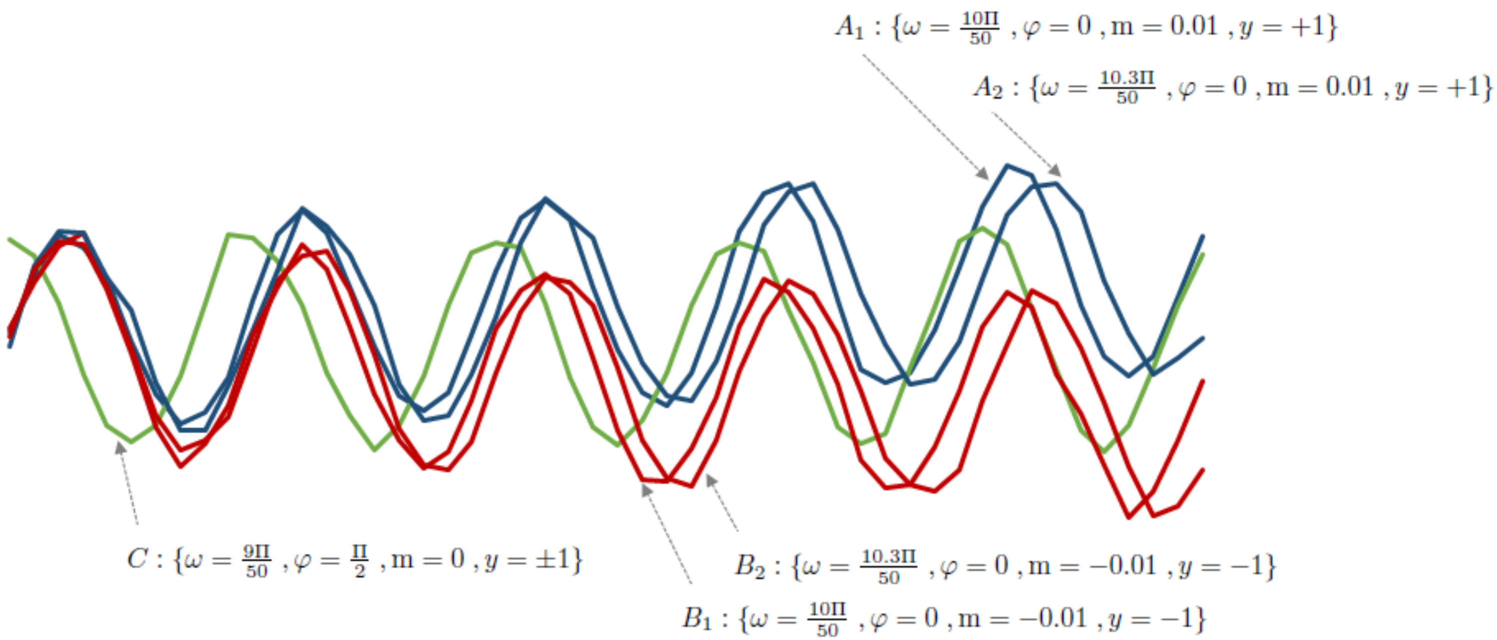
Outline

1. Introduction
2. Classification of time series: the standard setting
3. Early Classification of Times Series (ECTS)
4. A detour: the LUPI framework
5. Anticipation-based ECTS
6. Experiments and comparisons
7. Conclusions

Experiments: Controlled data

- Control of
  - The time-dependent **information** provided: the **slopes** of the **classes**
  - The **shapes** of time series within each class
  - The **noise level**

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



Results: effect of the noise level

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

$C(t)$	$\pm b$ / $\varepsilon(t)$	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 waiting cost

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

$C(t)$	$\pm b$ / $\varepsilon(t)$	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
	0.2	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.

AgroParisTech

Results: effect of the difference between classes

slope →

Increase of the difference between classes

The **performance** increases (AUC)

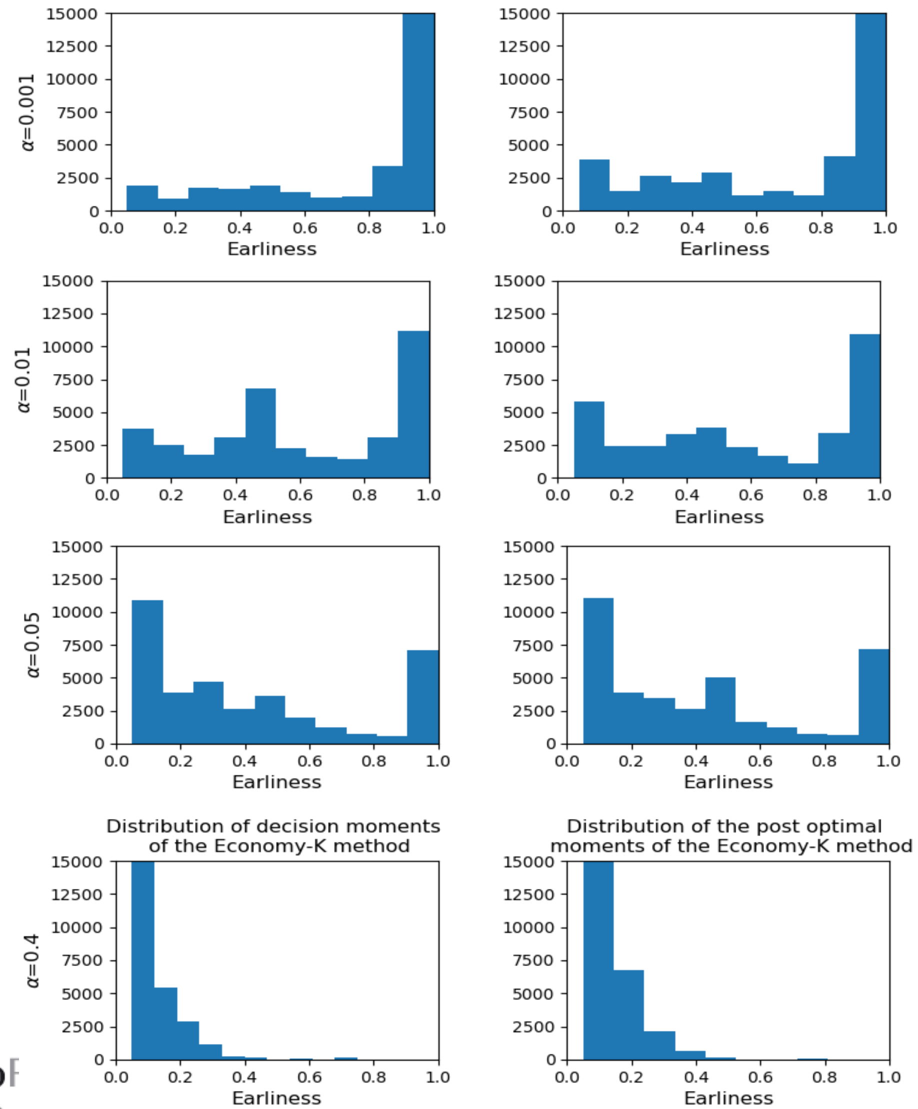
The *waiting time* is not much changed in these experiments

$C(t)$	$\pm b$ $\varepsilon(t)$	**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.

Are the decision times optimal

- Comparisons
  - Higher values of α mean higher delay cost

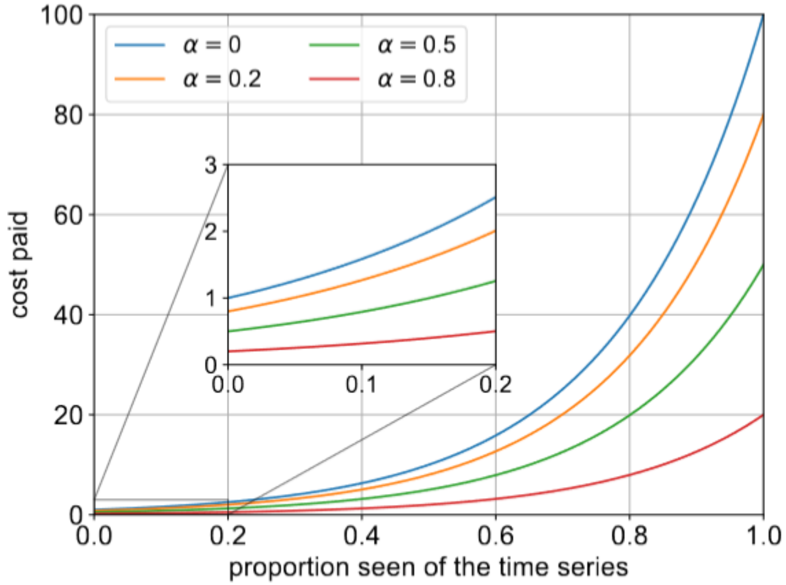


Left: decision times with Economy_K

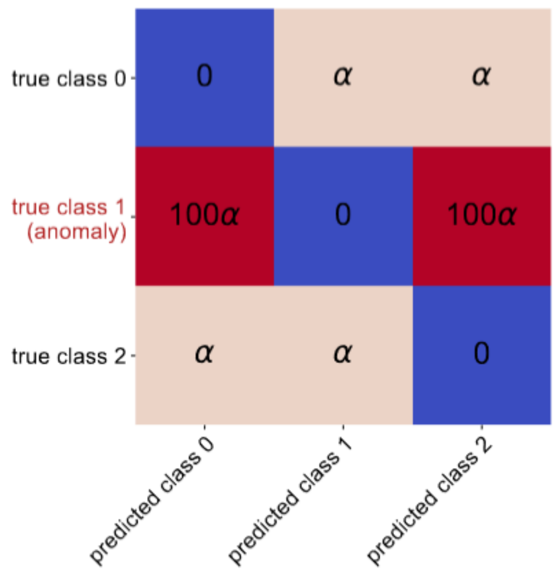
Right: optimal decision times afterwards

Achenchabe, Y. (2022). **From the early classification of time series to machine learning-based early decision-making** (Doctoral dissertation, Université Paris-Saclay).

Unbalanced misclassification cost and exponential delay costs

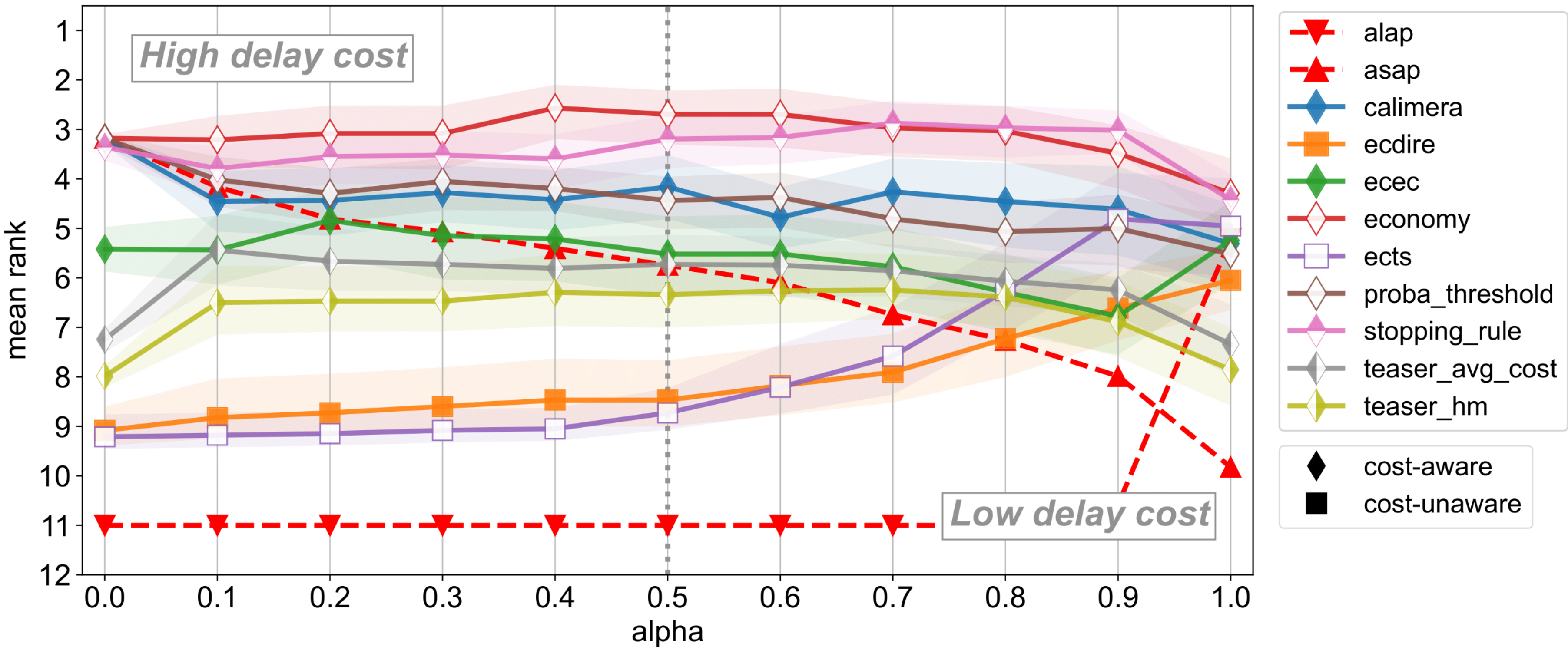


(a) Exponential delay cost



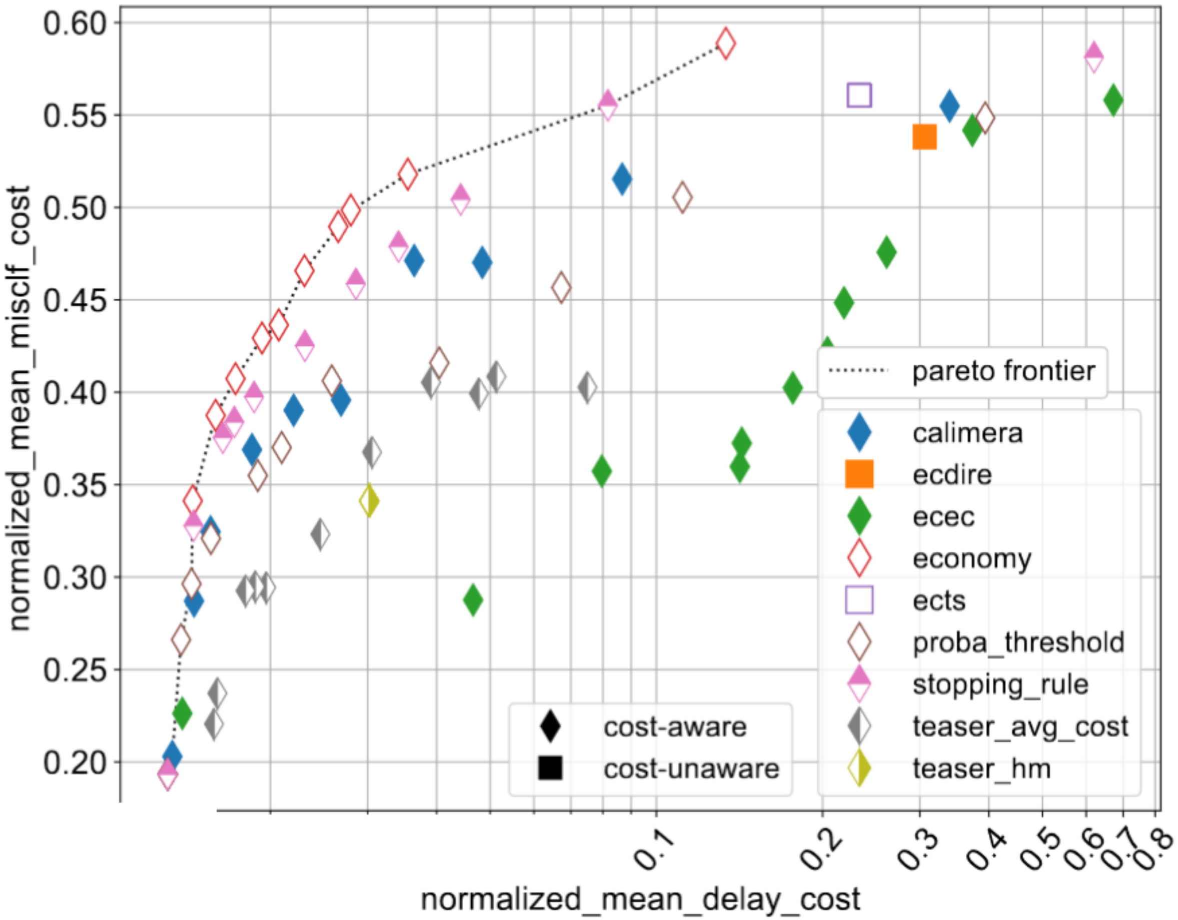
(b) Misclassification cost matrix

Unbalanced misclassification cost and exponential delay costs



Economy is on average, for all values of alpha, the **top method**

Unbalanced misclassification cost and exponential delay costs



Economy is on the Pareto front and tends to decide a little bit **earlier** than **SR**

Outline

AgroParisTech

- Lots of **applications**
  - Predictive **maintenance**
  - Early prediction of looming **disaster** (e.g. volcanic eruption)
  - Monitoring **patients**
  - Early prediction of late frost in **agriculture**
  - …

- **Extensions**

  Early classification on **data streams** (no end time *T*)

  ECTS when decisions are **revokable**

  - Autonomous car
    - **Believe** there is an obstacle → brake
    - Then **revoke** the former belief → increase speed

- When the decision **changes the future**
  - E.g. Cold chain
    - Predicting that merchandise will arrive spoiled → change the temperature

A counter part in **cognitive science**

and **experimental economy**?

The sunk-cost fallacy

Two avid sports fans plan to travel 40 miles to see a basketball game. **One** of them *paid* for his ticket; **the other** was on his way to purchase a ticket when he *got one free* from a friend.

A **blizzard** is announced for the night of the game with potential dire consequences for the drivers.

Which one of the two ticket holders is more likely to brave the blizzard at its own risk to see the game?

Daniel Kahneman (2017). **Thinking, fast and slow.** (p.343)

- **Misclassification** cost

	$p_+ = 0.5$	$p_- = 0.5$
True class / **Predicted** class	+	-
go	TP **gain = 100 - 40**	FP **gain = -1000-40**
stop	**gain = -40**	

Gain matrix

- **Misclassification** cost

- For the sport fan who paid 40$ for his ticket

	$p_+ = 0.5$	$p_- = 0.5$
True class / **Predicted** class	+	-
go	TP **gain = 100 - 40**	FP **gain = -1000-40**
stop	**gain = -40**	

Gain matrix

$$\mathbb{E}^{\,t}_{(\text{go},\text{y})\in\mathcal{Y}^2}\left[\mathrm{C}_m(\text{go}_t|y)\right]$$

$$= (p_+ \times (100 - 40)) + (p_- \times (-1000 - 40))$$

$$= \;30 - 520 \;=\; -490$$

$$\mathbb{E}^{\,t}_{(\text{stop},\text{y})\in\mathcal{Y}^2}\left[\mathrm{C}_m(\text{stop}_t|y)\right]$$

$$= (p_+ \times (-40)) + (p_- \times (-40)) \;=\; -40$$

$p_+ = 0.5$ $p_- = 0.5$

- **Misclassification** cost
- For the sport fan who got a free ticket

True class / Predicted class	+	-
go	**TP gain = 100**	**FP gain = -1000**
stop	**gain = 0**	

Gain matrix

$$\mathbb{E}^{t}_{(\text{go},\text{y})\in\mathcal{Y}^2}\left[\text{C}_m(\text{go}_t|y)\right]$$
$$= (p_+ \times (100)) + (p_- \times (-1000))$$
$$= 50 - 500 = -450$$

$$\mathbb{E}^{t}_{(\text{stop},\text{y})\in\mathcal{Y}^2}\left[\text{C}_m(\text{stop}_t|y)\right]$$
$$= (p_+ \times (0)) + (p_- \times (0))$$
$$= 0$$

- **Rationally**, the two sport fans **should not** try to drive in the blizzard to see the game

  (same difference between deciding to go and deciding to stop)

The sunk-cost fallacy

“**The sunk-cost fallacy**, (to keep a project alive when the rational decision would be to abandon it and star a new one,) keeps people **for too long** in poor jobs, unhappy marriages, and unpromising research projects. I have often observed young scientists struggling to salvage a doomed project when they would be better advised to drop it and start a new one.”

Daniel Kahneman (2017). **Thinking, fast and slow.** (p.346)

- **Delay** cost

$$C_d(t) = -40$$

$$C_d(t+1) = -50$$

	$p_+ = 0.4$	$p_- = 0.6$
True class / **Predicted** class	+	-
Go (one more step)	TP **gain = 100-50**	FP **gain = -10-50**
stop	FN **gain = -40**	TN **gain = -40**

Gain matrix

$p_+ = 0.4$ $p_- = 0.6$

True class / Predicted class	+	-
Go (one more step)	TP gain = 100-50	FP gain = -10-50
stop	FN gain = -40	TN gain = -40

Gain matrix

- **Delay** cost

$$C_d(t) = -40$$

$$C_d(t+1) = -50$$

True class / Predicted class	+	-
+	TP = 0.6	FP = 0.4
-	FN = 0.4	TN = 0.6

Confusion matrix of the classifier at time *t+1*

	$p_+ = 0.4$	$p_- = 0.6$
True class / Predicted class	+	-
Go (one more step)	TP gain = 100-50	FP gain = -10-50
stop	FN gain = -40	TN gain = -40

Gain matrix

- **Delay** cost

$$C_d(t) = -40$$

$$C_d(t+1) = -50$$

True class / Predicted class	+	-
+	TP = 0.6	FP = 0.4
-	FN = 0.4	TN = 0.6

Confusion matrix of the classifier at time *t+1*

$$\mathbb{E}^{t}_{(\mathrm{go},\mathrm{y})\in\mathcal{Y}^2}\left[C_m(\mathrm{go}_t|y)\right] = \mathbb{E}^{t+1}_{(\hat{\mathrm{y}}_{t+1},\mathrm{y})\in\mathcal{Y}^2}\left[C_m(\hat{y}_{t+1}|y)\right] = -40.4$$

$$\mathbb{E}^{t}_{(\mathrm{stop},\mathrm{y})\in\mathcal{Y}^2}\left[C_m(\mathrm{stop}_t|y)\right] = -40$$

==> STOP

$p_{+} = 0.4$ $p_{-} = 0.6$

- **Delay** cost

~~$= -40$~~

$$C_d(t) = -50$$

~~$= -50$~~

$$C_d(t+1) = -60$$

True class / Predicted class	+	-
Go (one more step)	TP gain = 100-60	FP gain = -10-60
stop	FN gain = -50	TN gain = -50

Gain matrix

True class / Predicted class	+	-
+	TP = 0.6	FP = 0.4
-	FN = 0.4	TN = 0.6

Confusion matrix of the classifier at time *t+1*

$$\mathbb{E}^{\,t}_{(\text{go},\text{y})\in\mathcal{Y}^2}\left[\text{C}_m(\text{go}_t|y)\right] = \mathbb{E}^{\,t+1}_{(\hat{\text{y}}_{t+1},\text{y})\in\mathcal{Y}^2}\left[\text{C}_m(\hat{y}_{t+1}|y)\right] = -52.6$$

$$\mathbb{E}^{\,t}_{(\text{stop},\text{y})\in\mathcal{Y}^2}\left[\text{C}_m(\text{stop}_t|y)\right] = -50$$

==> STOP (even more so)

- But human deciders tend to do the opposite

  - Choosing to go **one step further** and that all the more that the **cost already paid** is **high**

  - As if:
    - The **probability of success** was higher than it is
    - The **increased cost** of sticking to the project was less than it actually is

Underlying impulsion: humans want to **recover** the costs incurred to date

- The ECTS algorithm with anticipation (LUPI) is like a system 2 (rational decision system)