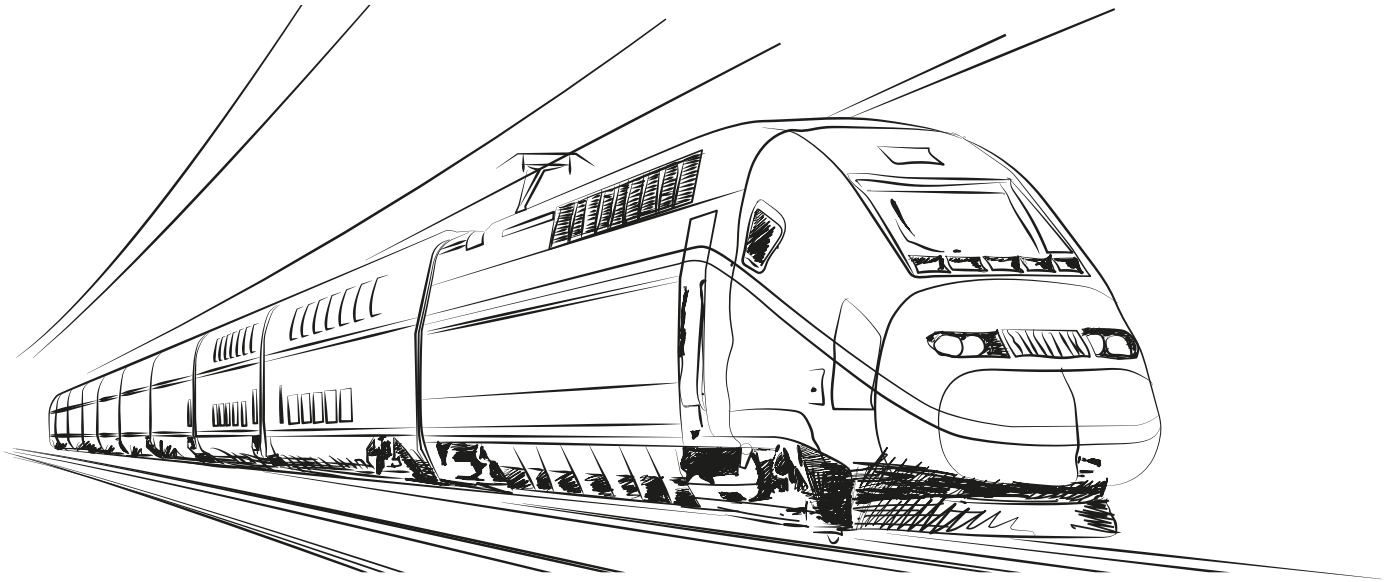
Bayesian feature discovery for predictive maintenance

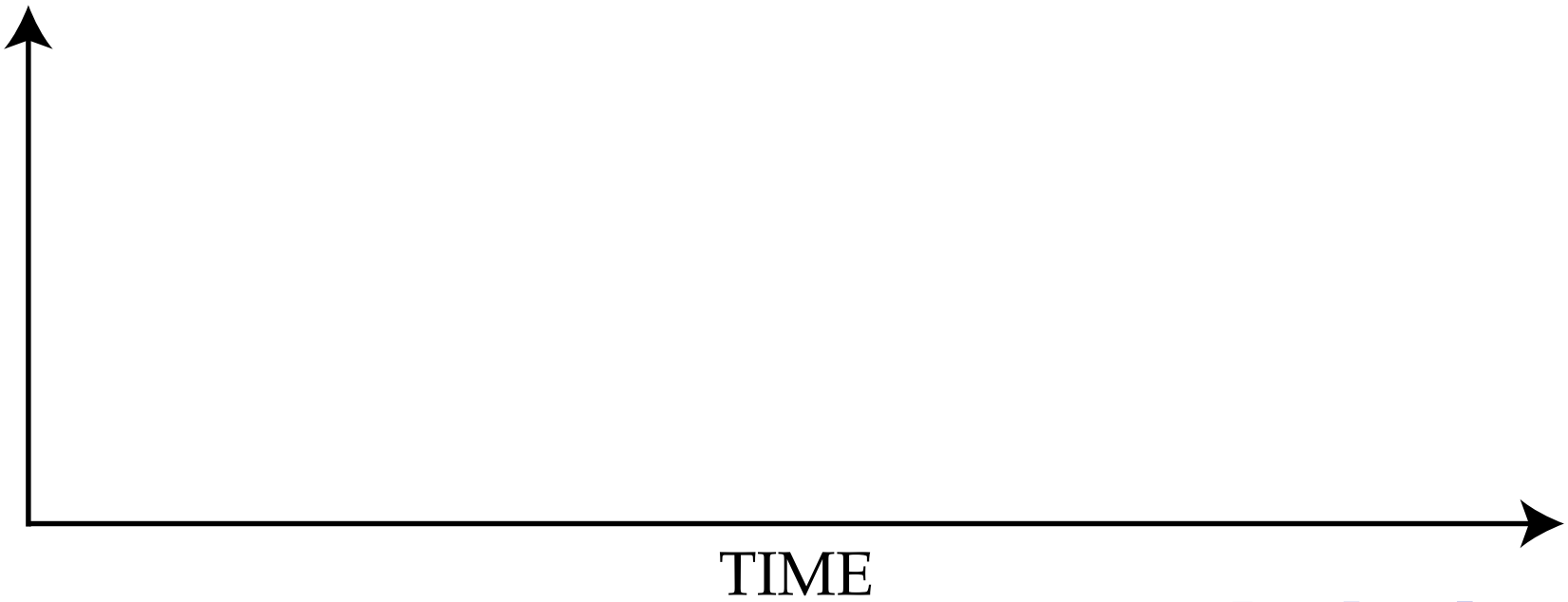
Amir Dib[†],

[†]Université Paris-Saclay, ENS Paris-Saclay, CNRS, Centre Borelli, 91190, Gif-sur-Yvette, France
[‡]ITNOVEM, SNCF, 93120, Saint-Denis, France

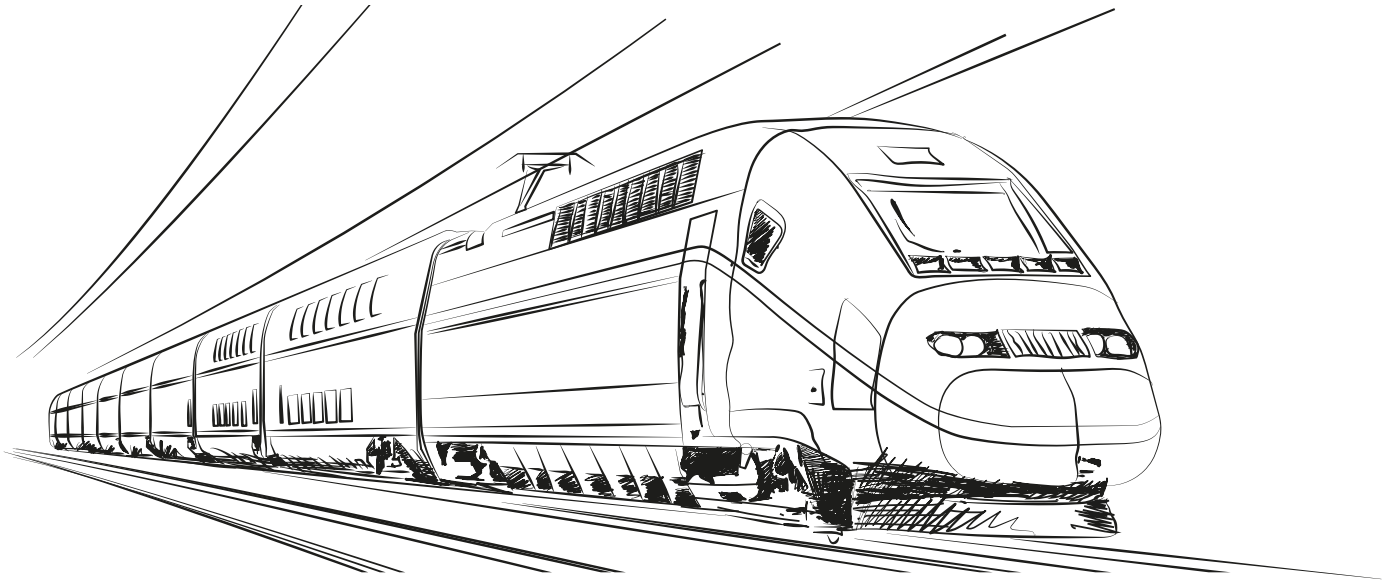
Advisors: Nicolas Vayatis, Mathilde Mougeot
Industrial advisor: Héloise Nonne

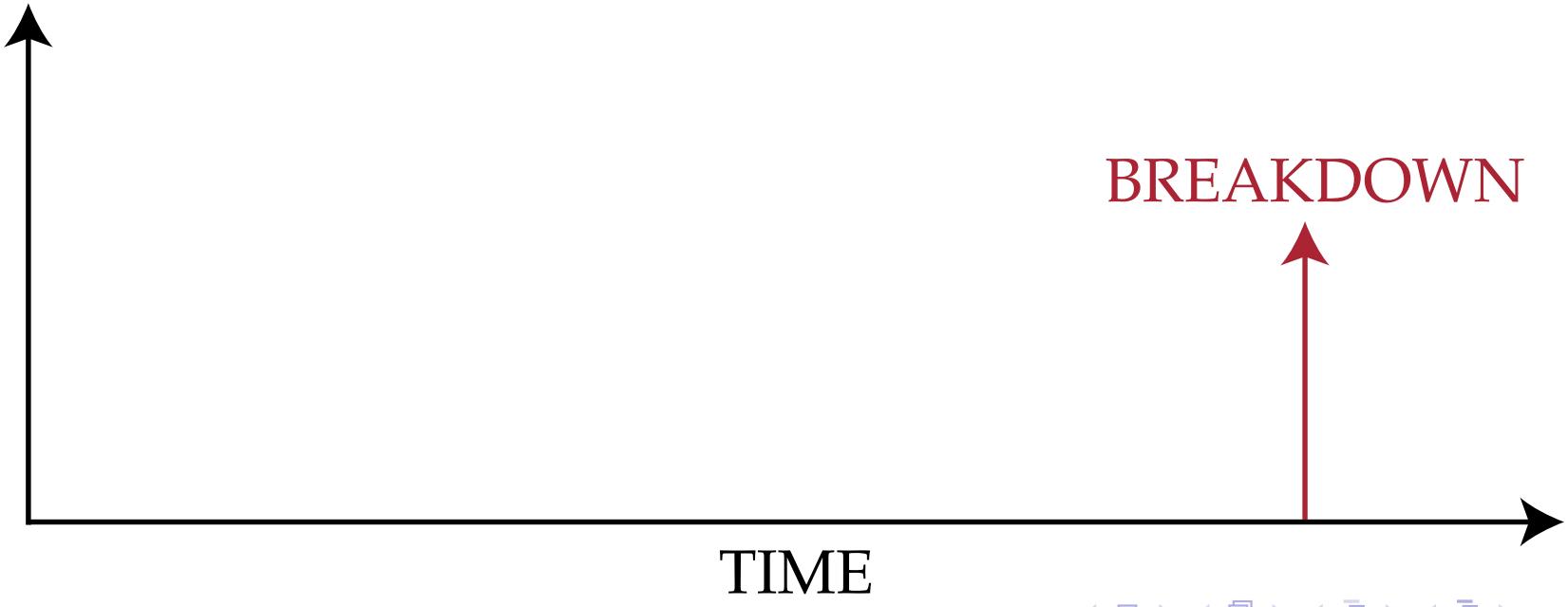
The life of a train



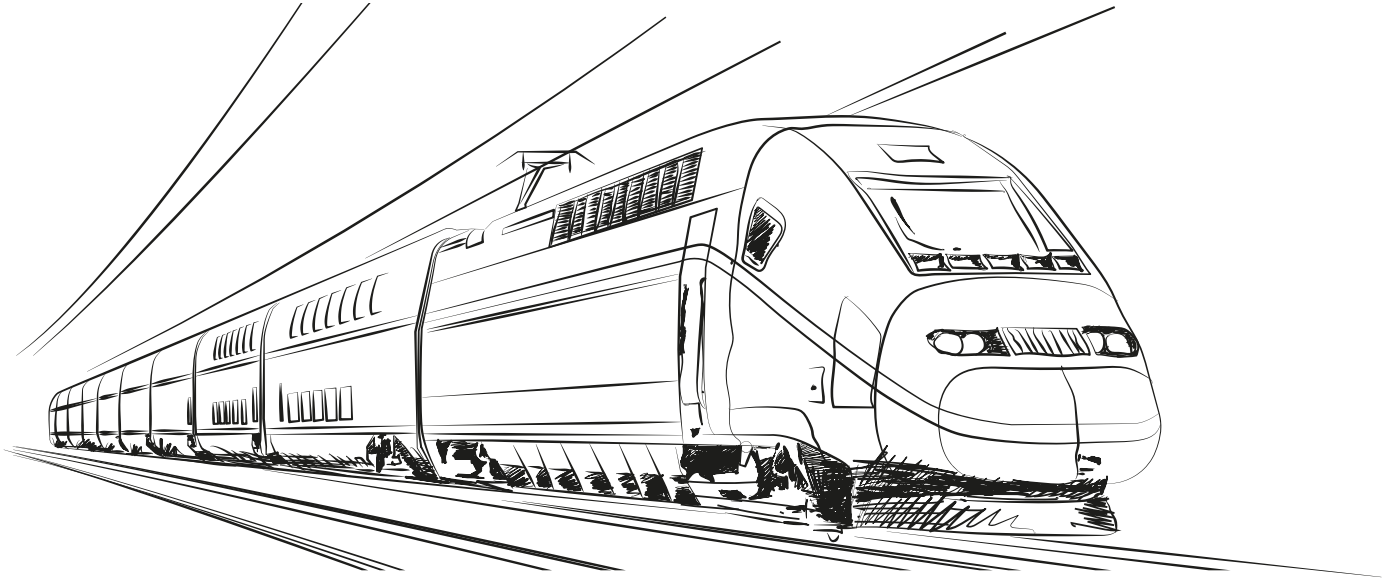


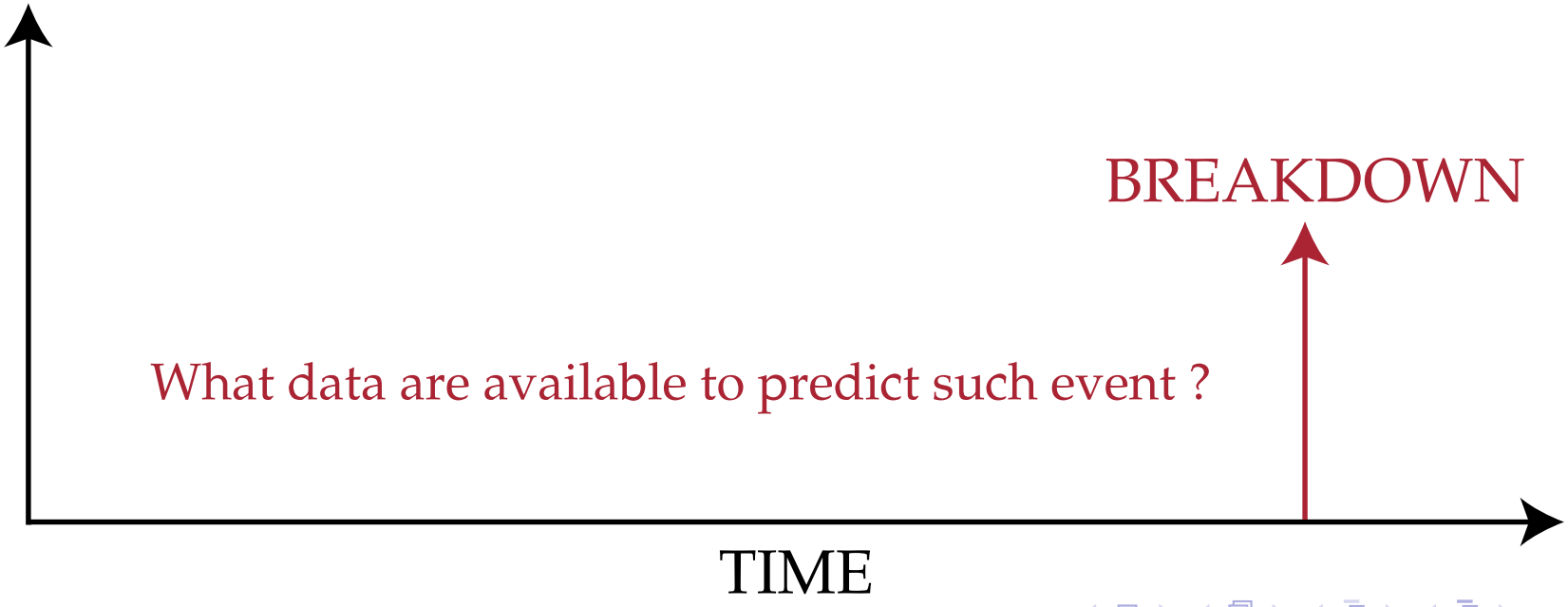
The life of a train



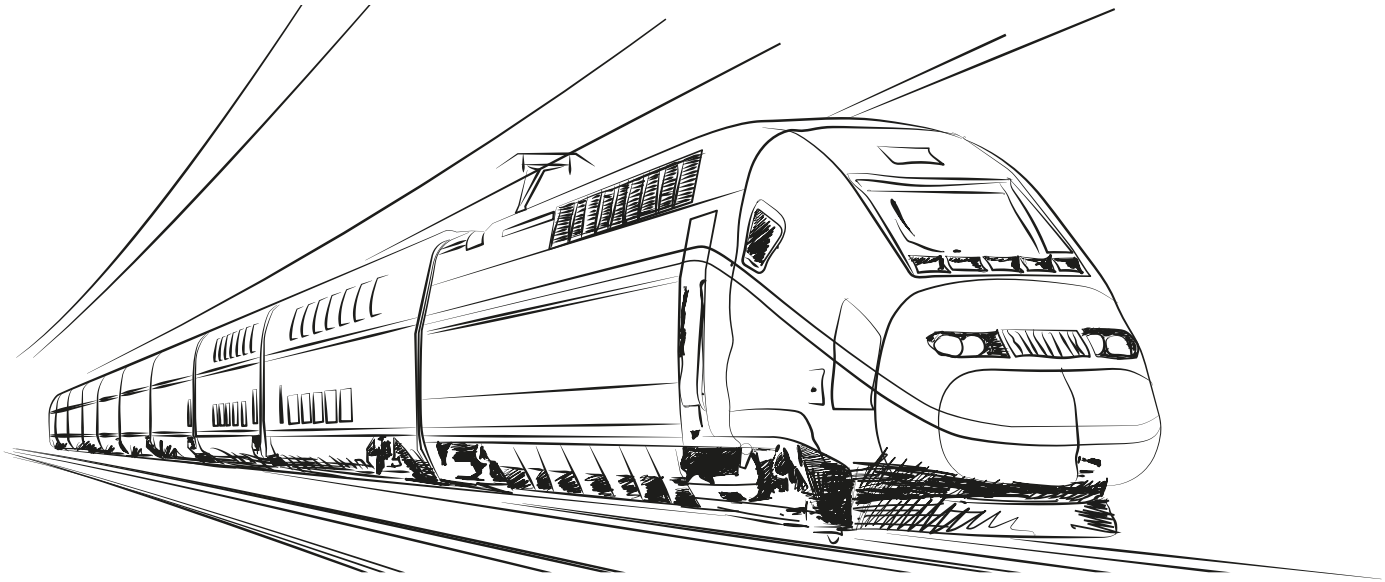


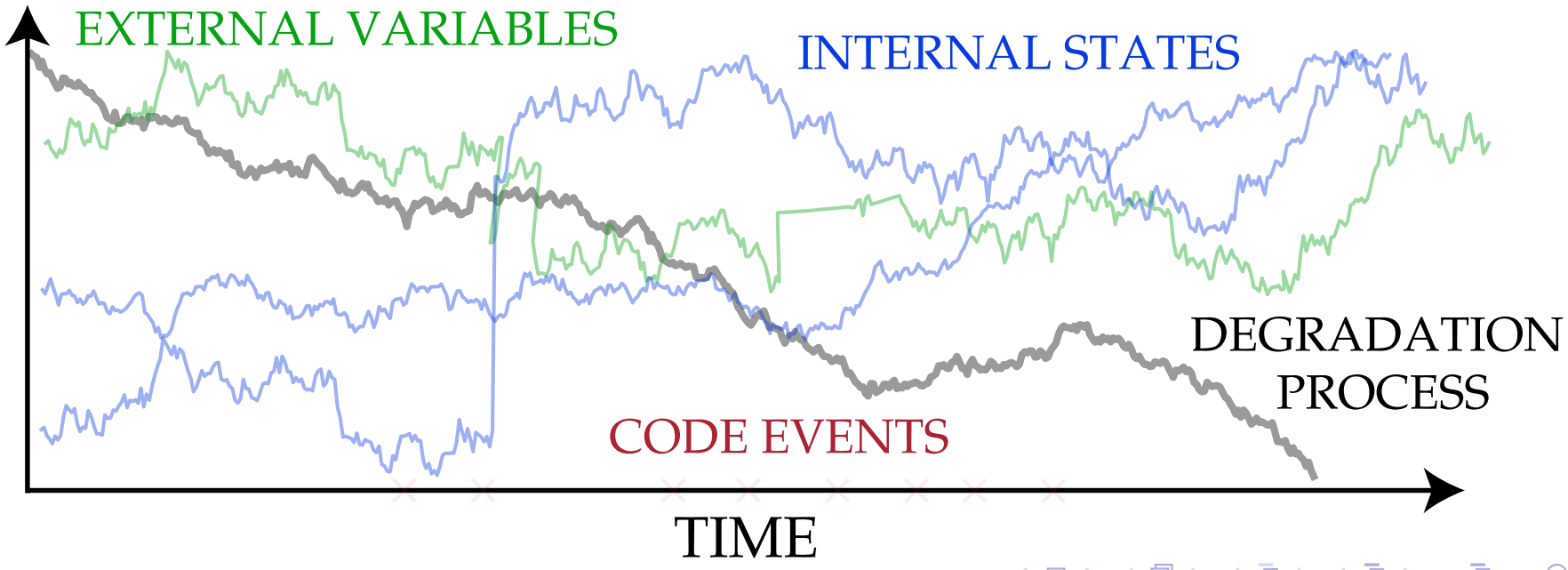
The life of a train



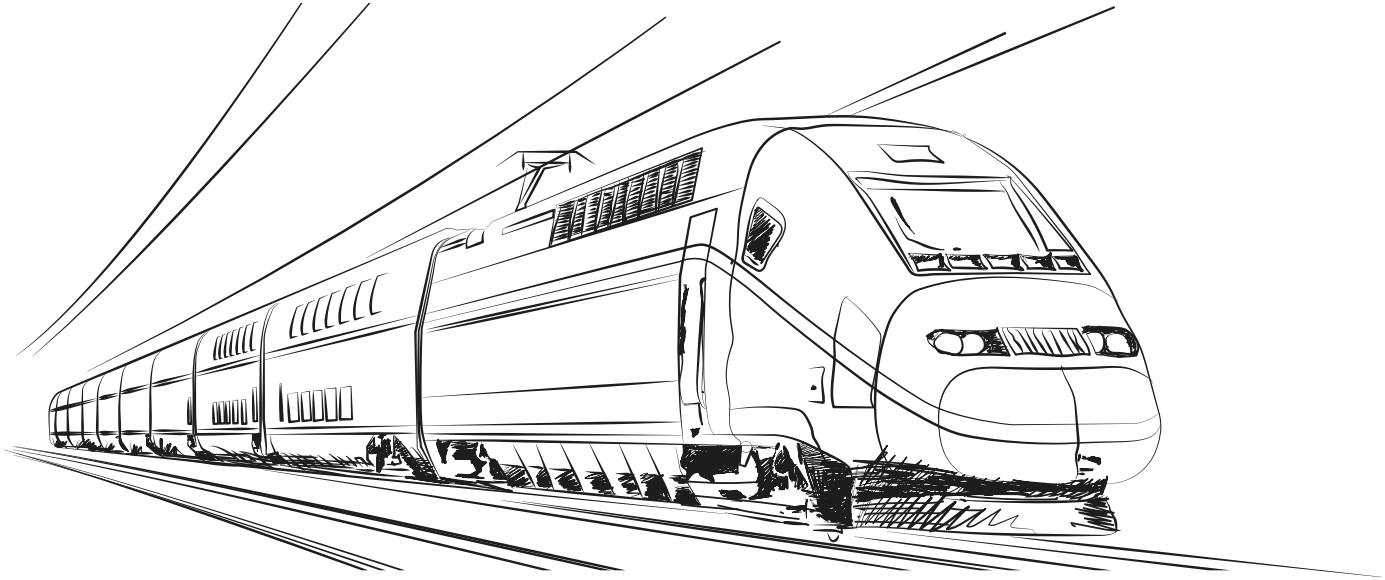


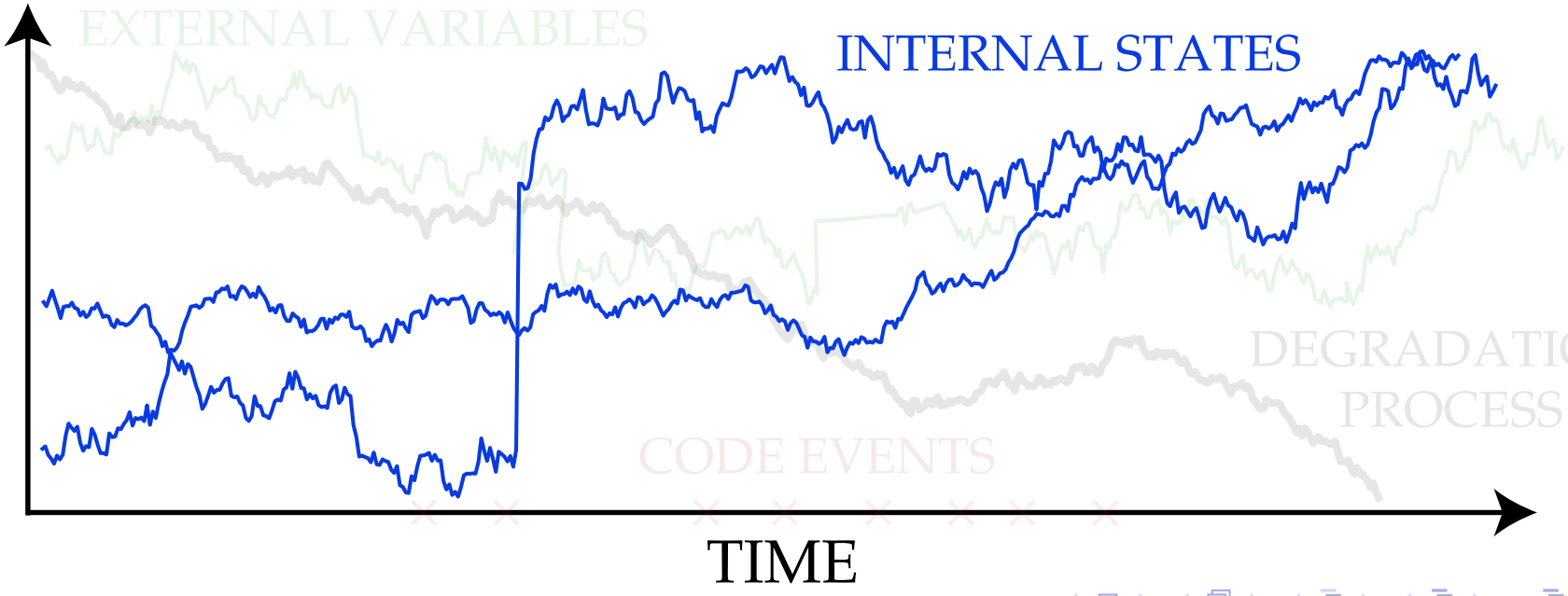
The life of a train



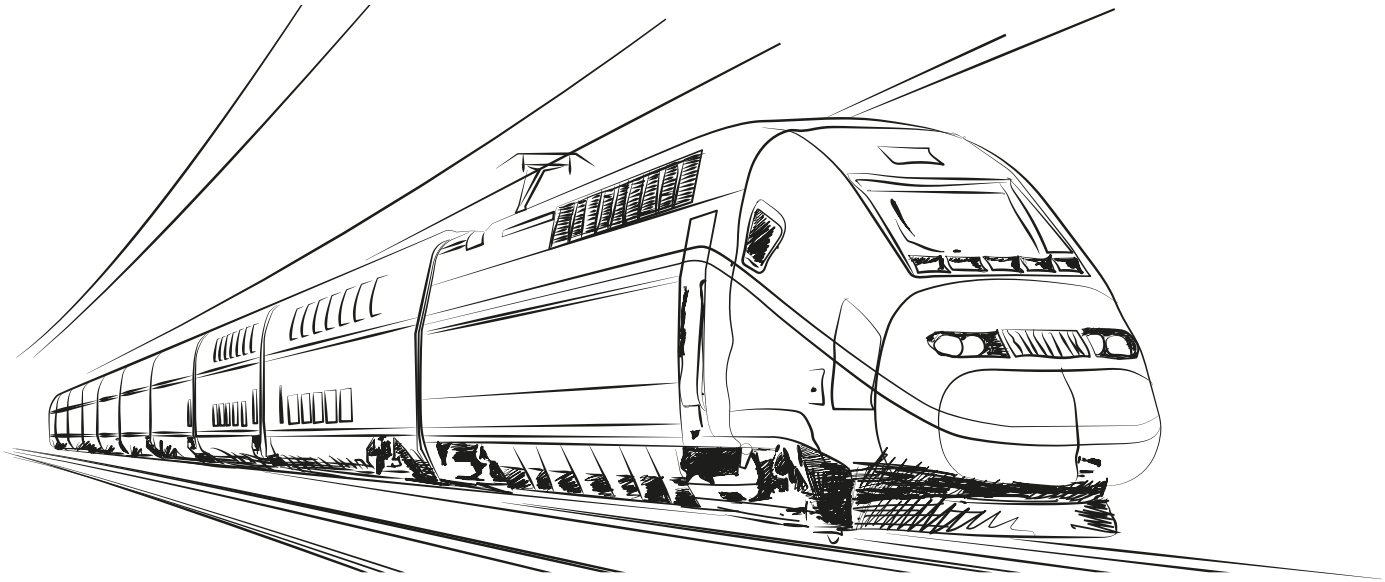


The life of a train

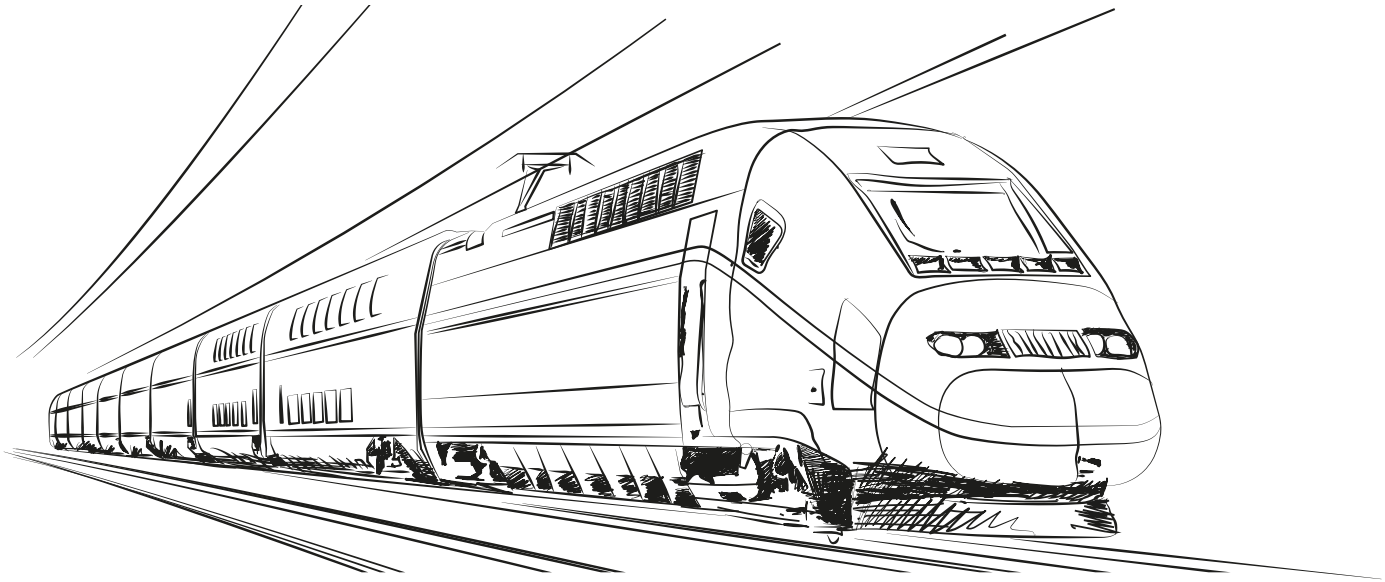


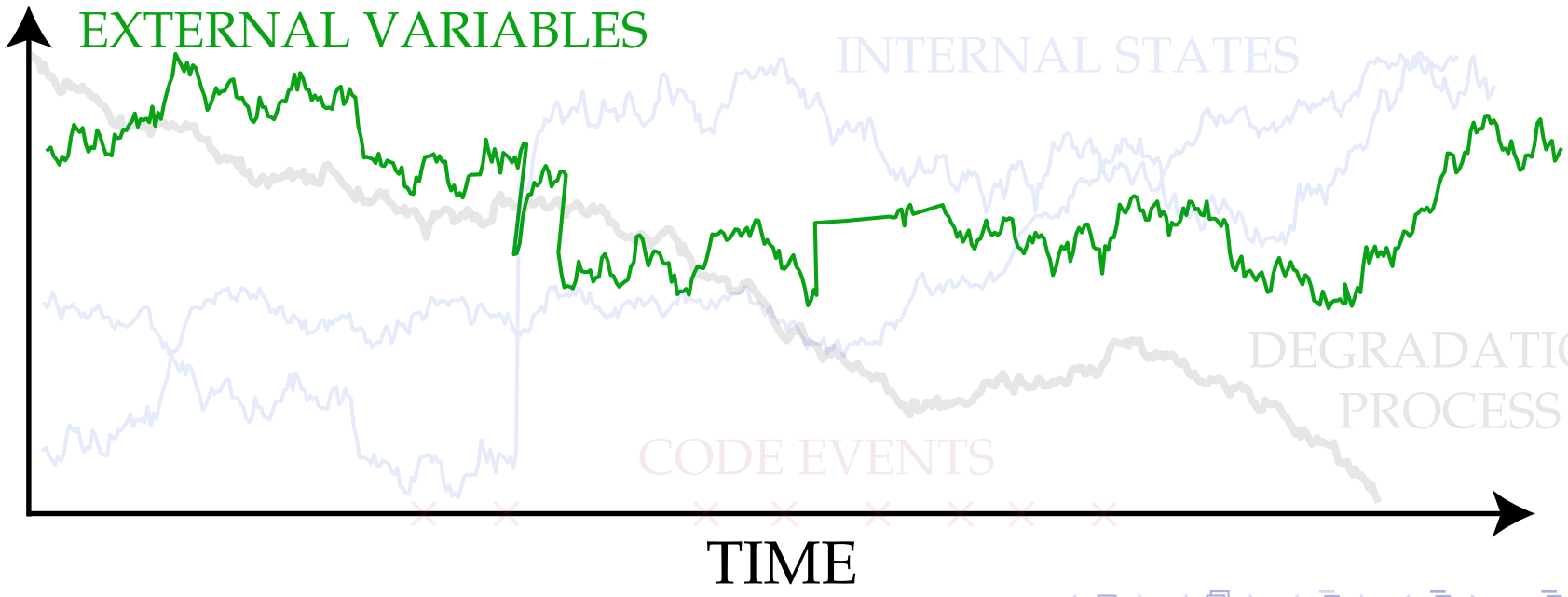


The life of a train

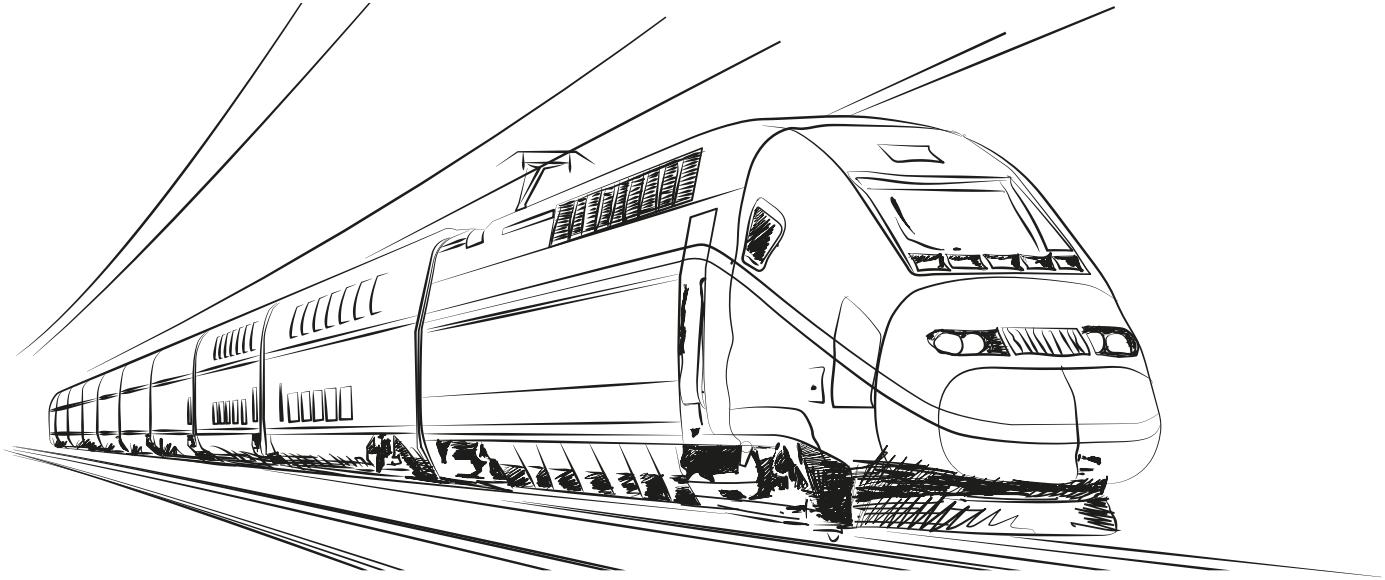


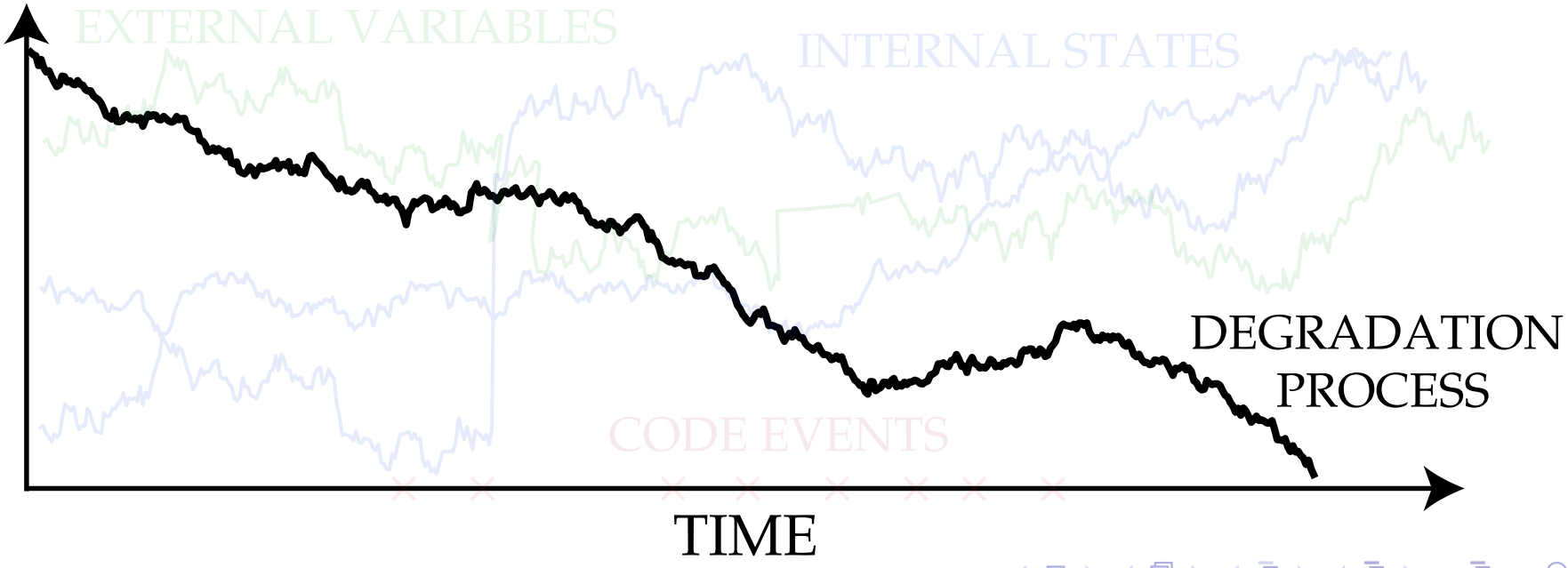
The life of a train





The life of a train





Formalism: the feature space

Let $(\Omega, \mathcal{A}, \mathbb{P})$ be a probability space. We introduce the following variables

Event codes. The set of *error codes* $\Sigma = \{e_i, 1 \leq i \leq d\}$ that can be emitted;

Formalism: the feature space

Let $(\Omega, \mathcal{A}, \mathbb{P})$ be a probability space. We introduce the following variables

Event codes. The set of *error codes* $\Sigma = \{e_i, 1 \leq i \leq d\}$ that can be emitted;

Code	Libellé
8025	PD : Def. clos
8425	PG : Def. clos
16111	Def. camera 2
20052	LT Autorisation RD
20053	LT Autorisation em RD

Formalism: the feature space

Let $(\Omega, \mathcal{A}, \mathbb{P})$ be a probability space. We introduce the following variables

Event codes. The set of *error codes* $\Sigma = \{e_i, 1 \leq i \leq d\}$ that can be emitted;

Feature space. The feature space of event as the set of random covariates $\boldsymbol{X_t} : \Omega \longrightarrow \Sigma \times \mathbb{R}^K$ with K internal and external real valued time series;

Code	Libellé
8025	PD : Def. clos
8425	PG : Def. clos
16111	Def. camera 2
20052	LT Autorisation RD
20053	LT Autorisation em RD

Formalism: the feature space

Let $(\Omega, \mathcal{A}, \mathbb{P})$ be a probability space. We introduce the following variables

Event codes. The set of *error codes* $\Sigma = \{e_i, 1 \leq i \leq d\}$ that can be emitted;

Feature space. The feature space of event as the set of random covariates $\boldsymbol{X_t} : \Omega \longrightarrow \Sigma \times \mathbb{R}^K$ with K internal and external real valued time series;

Code	Libellé
8025	PD : Def. clos
8425	PG : Def. clos
16111	Def. camera 2
20052	LT Autorisation RD
20053	LT Autorisation em RD

Time	Code	X^1	...	X^K
t_1	e_1	$x^1_{t_1}$	...	$x^K_{t_1}$
t_2	e_4	$x^1_{t_2}$	...	$x^K_{t_2}$
t_3	e_1	$x^1_{t_3}$	...	$x^K_{t_3}$
t_4	e_3	$x^1_{t_4}$	...	$x^K_{t_4}$
t_5	e_1	$x^1_{t_5}$	...	$x^K_{t_5}$

Formalism: the target space

Degradation process. A real valued r.v. $Z_t : \Omega \longrightarrow \mathbb{R}$ representing the degradation process at eatch time $t \in \mathbb{R}_+$ and z_f threshold indicating if the system is considered malfunctioning.

Formalism: the target space

Degradation process. A real valued r.v. $Z_t : \Omega \longrightarrow \mathbb{R}$ representing the degradation process at eatch time $t \in \mathbb{R}_+$ and z_f threshold indicating if the system is considered malfunctioning.

Target. The binary health status $Y_t = \mathbb{1}_{Z_t \leq z_f}$ at each time $t \in \mathbb{R}$.

Formalism

This framework spans a very large class of problem that are very evolving system with feature variable valued in an *unordered set* such as

- Graph;
- Sentences;
- DNA sequences.

Challenges

- Construct a relevant feature space with all the available data is a hard task;
- The is no straightforward way to process a symbolic time serie data into a stat model pipeline:
- The output of the prediction pipeline must be interpretable by the experts;
- The overall computational pipeline must run with reasonable computational requirements.

How to process symbolic data ?

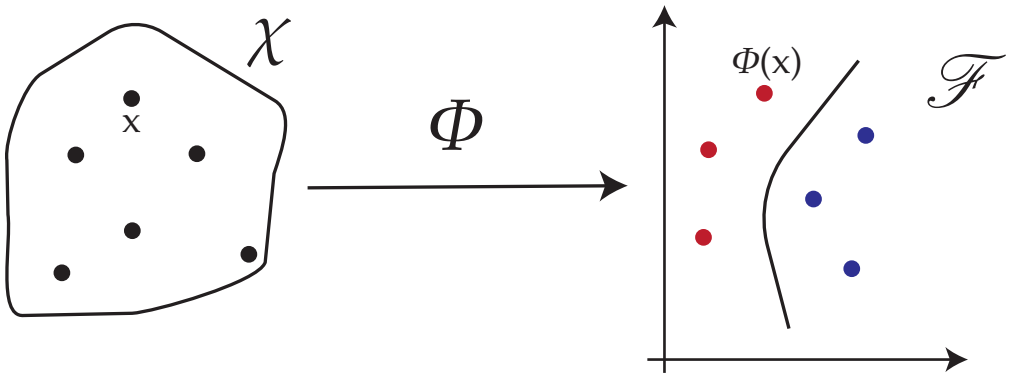
There is three common way to treat symbolic data:

Kernel embeddings. (Muandet et al., 2017) Kernel methods rely on a positive definite kernel function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ that induce a mapping $\phi : \mathcal{X} \to \mathcal{H}$ in a hilbert pace $\mathcal{H}$.

How to process symbolic data ?

There is three common way to treat symbolic data:

Kernel embeddings. (Muandet et al., 2017) Kernel methods rely on a positive definite kernel function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ that induce a mapping $\phi : \mathcal{X} \to \mathcal{H}$ in a hilbert pace $\mathcal{H}$.



How to process symbolic data ?

There is three common way to treat symbolic data:

Kernel embeddings. (Muandet et al., 2017) Kernel methods rely on a positive definite kernel function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ that induce a mapping $\phi : \mathcal{X} \to \mathcal{H}$ in a hilbert pace $\mathcal{H}$.

State machine model (Kamlu and Laxmi, 2019) . Modelizes directly the degradation process through the computation of transition matrix between hidden states against observed random states (see appendix of the thesis manuscript for more details).

How to process symbolic data ?

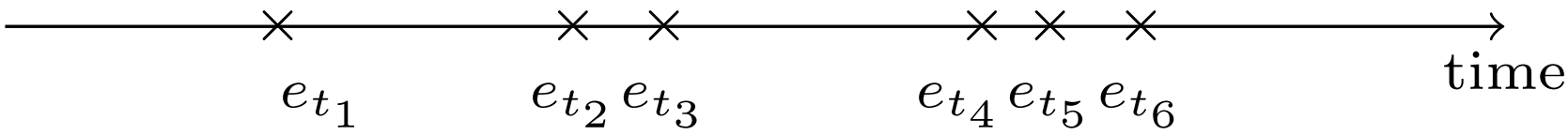
There is three common way to treat symbolic data:

Kernel embeddings. (Muandet et al., 2017) Kernel methods rely on a positive definite kernel function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ that induce a mapping $\phi : \mathcal{X} \to \mathcal{H}$ in a hilbert pace $\mathcal{H}$.

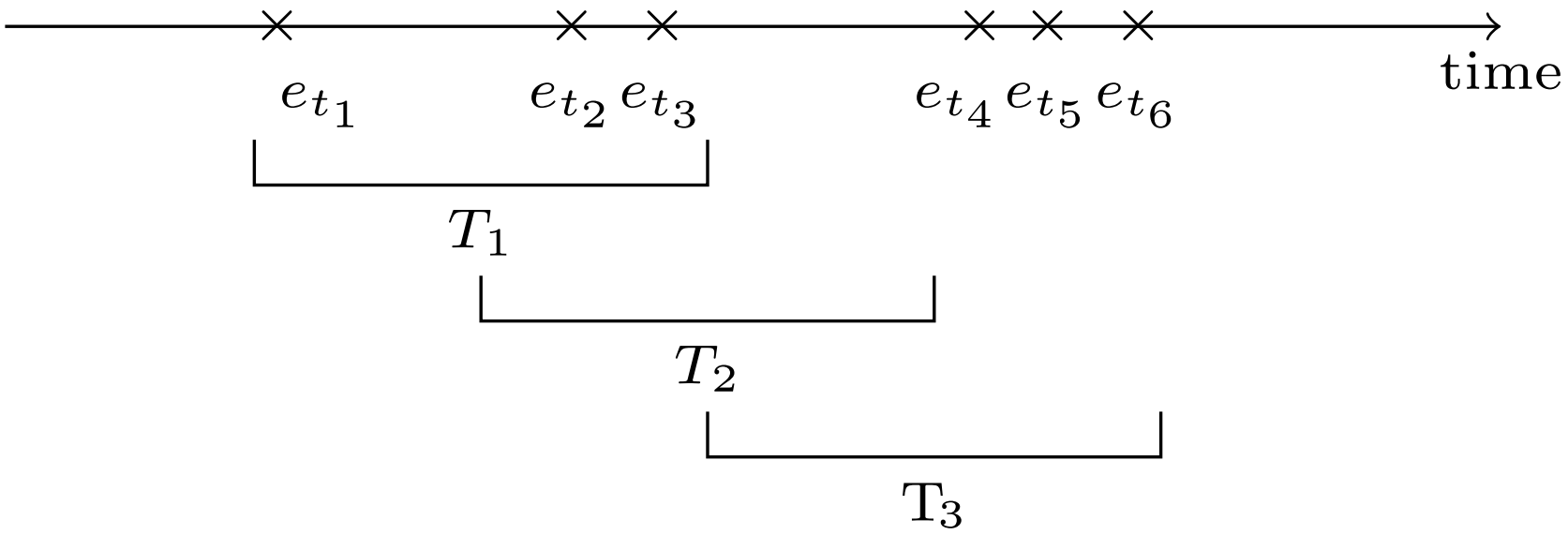
State machine model (Kamlu and Laxmi, 2019) . Modelizes directly the degradation process through the computation of transition matrix between hidden states against observed random states (see appendix of the thesis manuscript for more details).

Windowing approach. Aggregates signal over parametrized time windows.

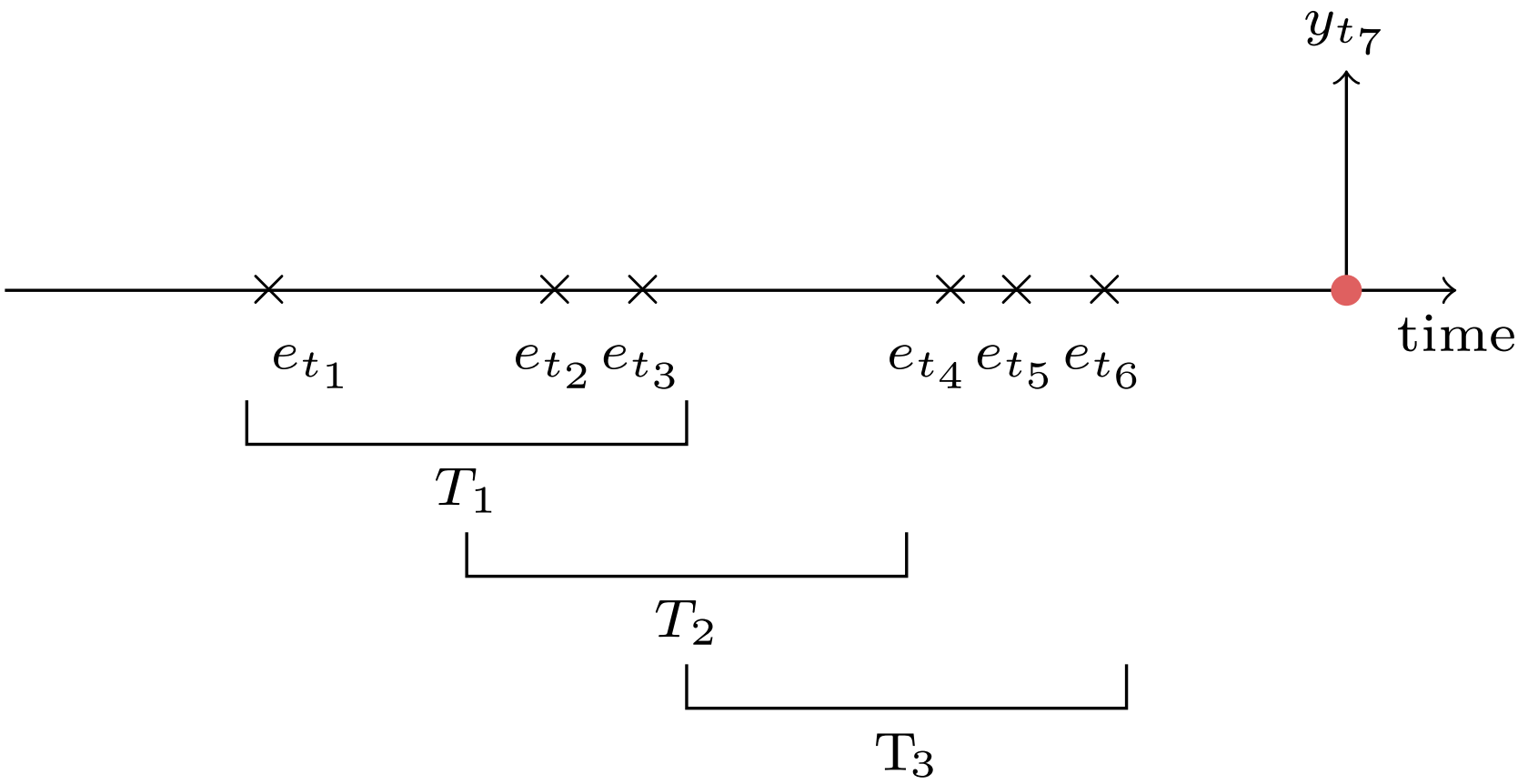
How to process symbolic data ?



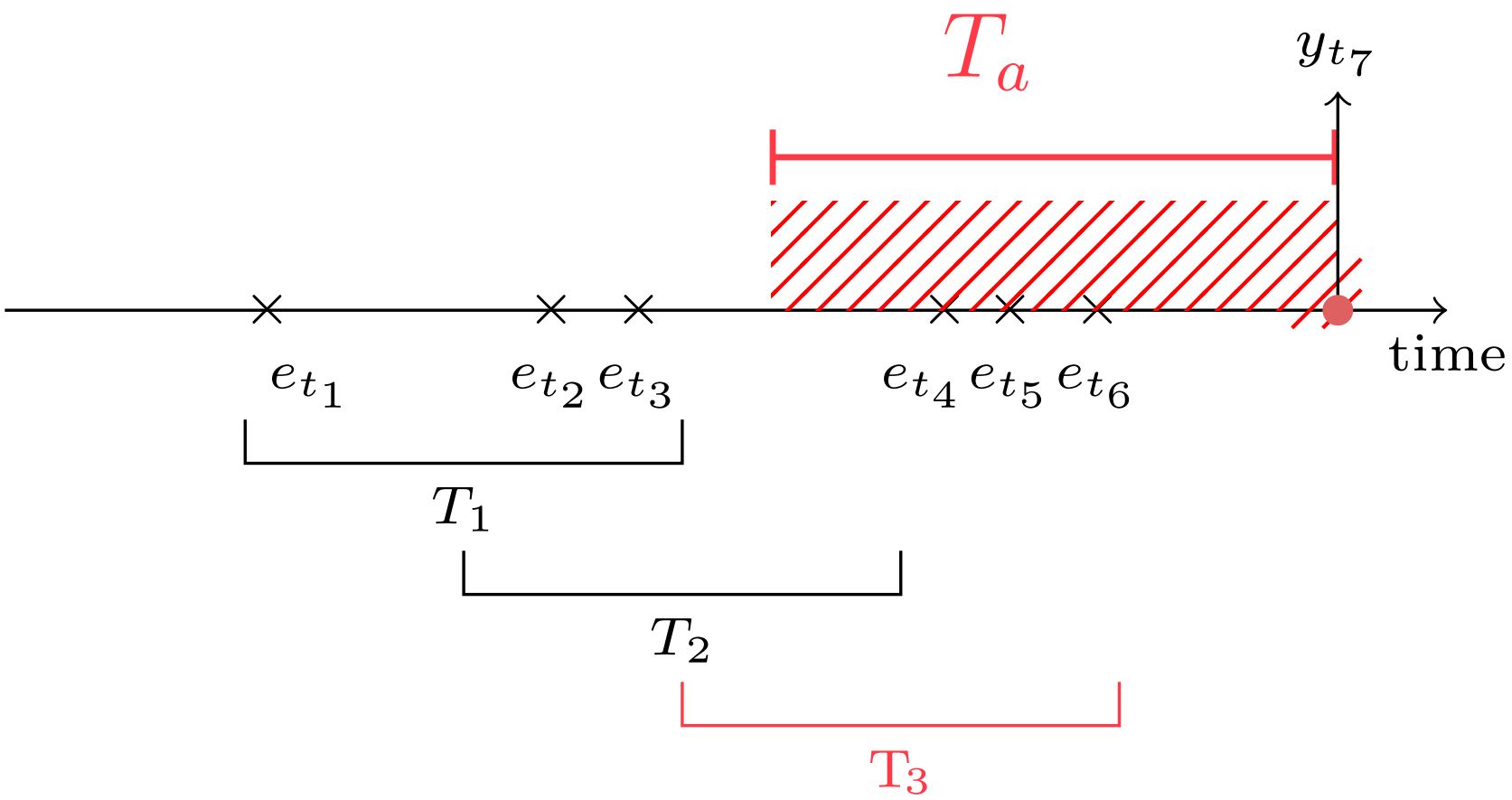
How to process symbolic data ?



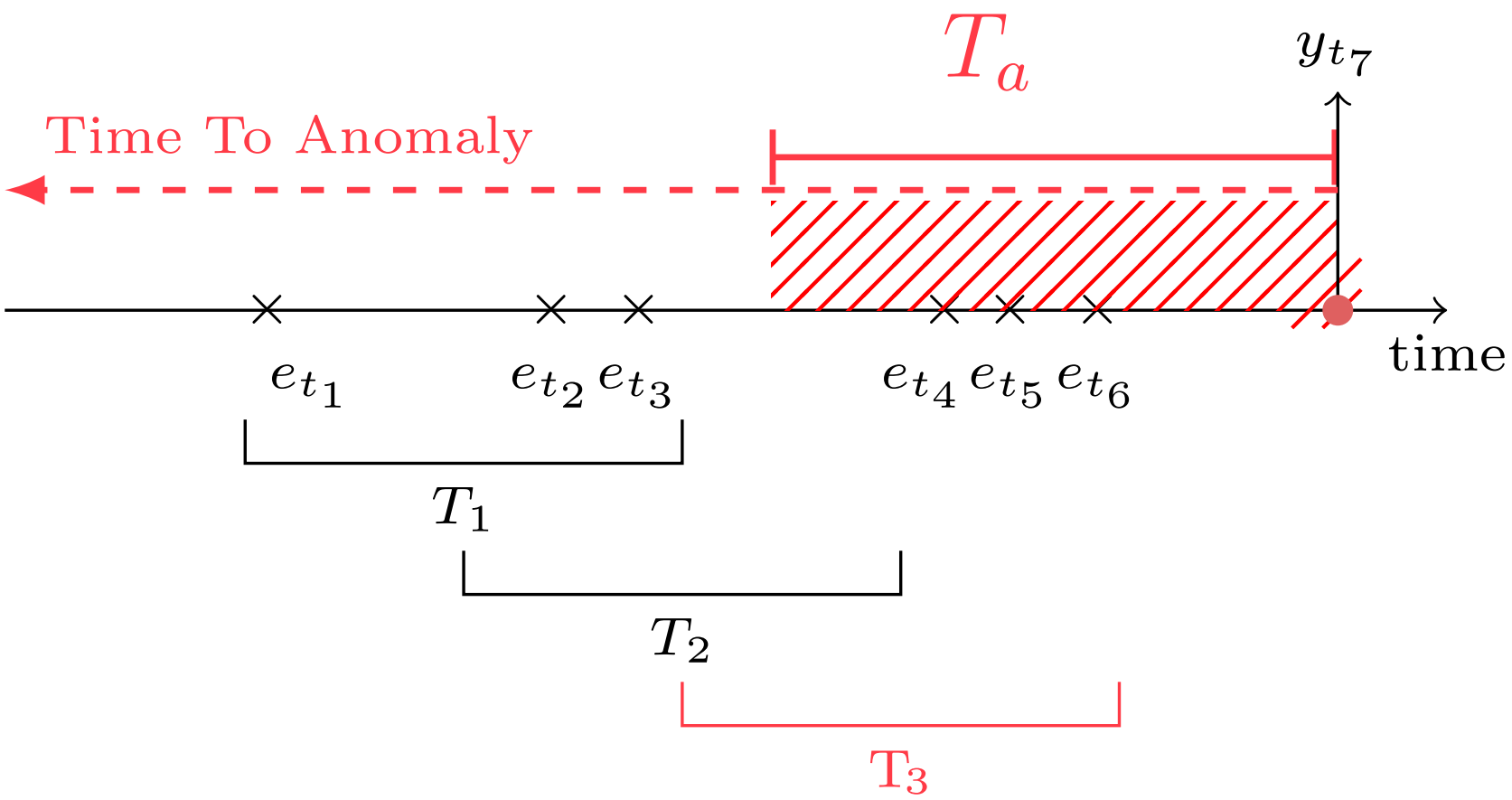
How to process symbolic data ?



How to process symbolic data ?



How to process symbolic data ?



How to process symbolic data ?

At the end, we obtain a classical ML dataset with numerical quantities

Window	C_1	...	C_d	...	TTA (days)
$\mathcal{T}_1$	2	...	10	...	10
$\mathcal{T}_2$	0	...	2	...	5
$\mathcal{T}_3$	1	...	0	...	3

Results: traditional ML approaches for the french train fleet

	X Gradient Boosting			Random Forest			Light Gradient-Boosting Machine			Categorical Boosting			Linear Regression			k-Nearest Neighbors		
	[1]	[1,7]	[1,7,14]	[1]	[1,7]	[1,7,14]	[1]	[1,7]	[1,7,14]	[1]	[1,7]	[1,7,14]	[1]	[1,7]	[1,7,14]	[1]	[1,7]	[1,7,14]
TGV Doors																		
AUC	0.728	0.73	**0.758**	0.72	0.725	**0.749**	0.733	0.73	**0.756**	0.634	0.632	**0.659**	0.699	0.707	**0.725**	**0.582**	0.578	0.562
Accuracy	0.659	0.668	**0.692**	0.659	0.674	**0.683**	0.671	0.669	**0.692**	0.597	0.594	**0.608**	0.645	0.653	**0.667**	**0.567**	0.556	0.55
Recall	0.591	0.59	**0.609**	0.608	0.616	**0.625**	0.575	0.564	**0.597**	0.611	**0.645**	0.628	0.547	0.531	**0.561**	0.542	**0.552**	0.541
F1	0.634	0.64	**0.664**	0.641	0.654	**0.663**	0.636	0.63	**0.659**	0.602	0.613	**0.616**	0.606	0.605	**0.627**	**0.556**	0.554	0.546

Table: Test Accuracy, Recall and AUC 5× cross-validated on datasets reported in the thesis manuscript.

Interpretability

What about interpretability ?

Interpretability

What about interpretability ?

We want an output in term of patterns of codes of the form

$$(e_1, e_5) \rightarrow \textit{Failure}.$$

Interpretability

What about interpretability ?

We want an output in term of patterns of codes of the form

$$(e_1, e_5) \rightarrow \textit{Failure}.$$

This is the domain of Pattern Mining (Agrawal, Imielinski, and Swami, 1993).

Results: mining to interpret

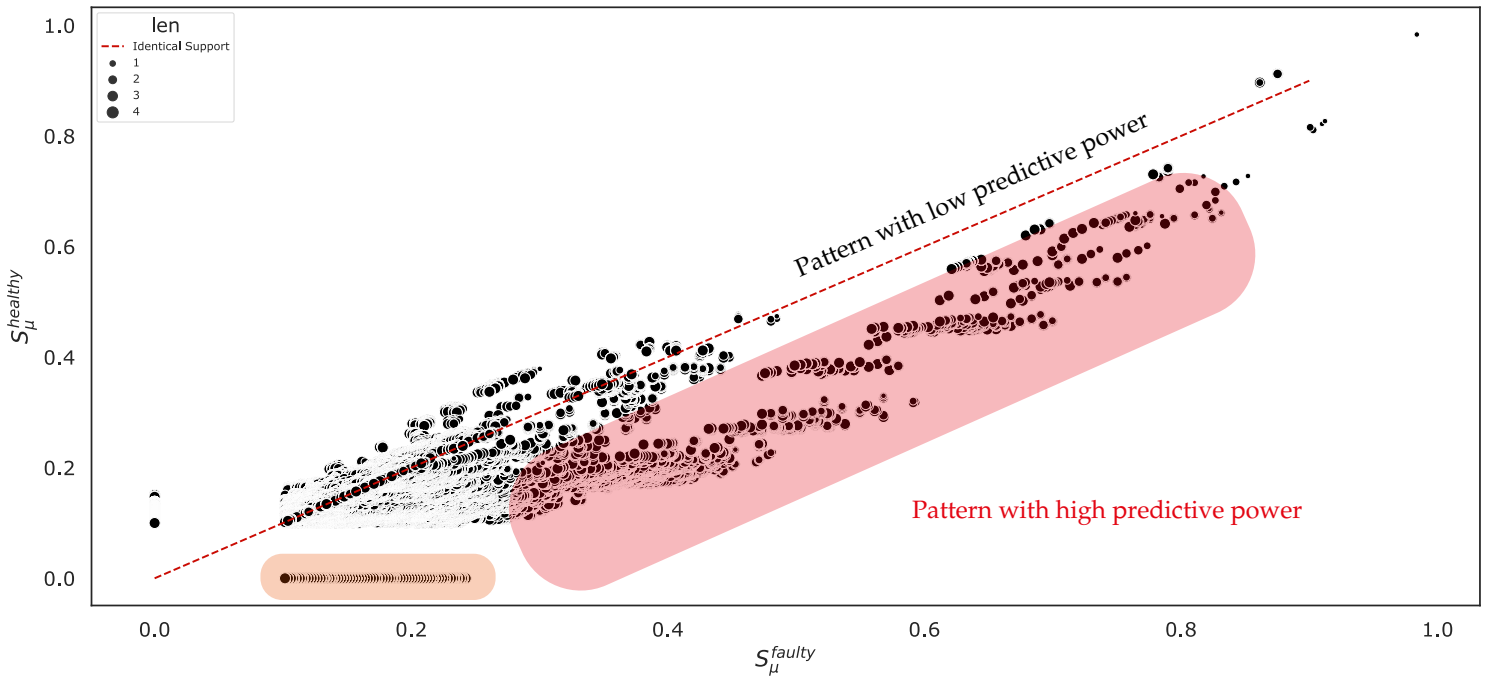


Figure: Support on each class of patterns extracted by algorithm *a priori* (Agrawal and Srikant, 1994) for $\mu = 1\%$ and $\mu = 4\%$ and patterns of different sizes for the Doors dataset.Each black point is a pattern of codes with size representing the length of the pattern. Patterns that are in the upper half of the figure are the patterns that appears mostly near breakdowns events and pattern that are in the bottom half of the bisector (red dotted line) are the one appear in period without breakdowns.

Motivations

Pattern mining of events is crucial towards interpretable anomaly detection.

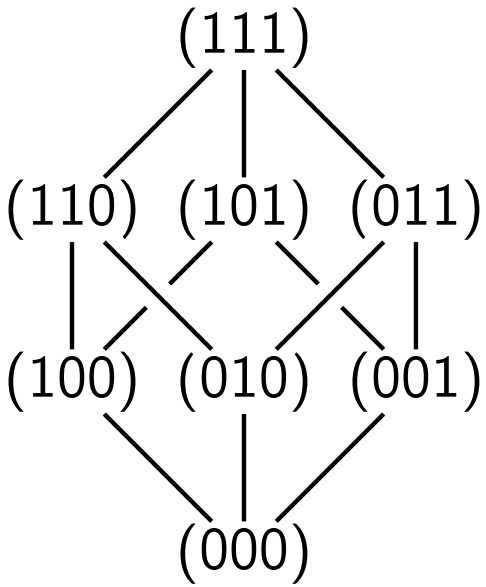
Motivations

Pattern mining of events is crucial towards interpretable anomaly detection.

How do we extract them ?

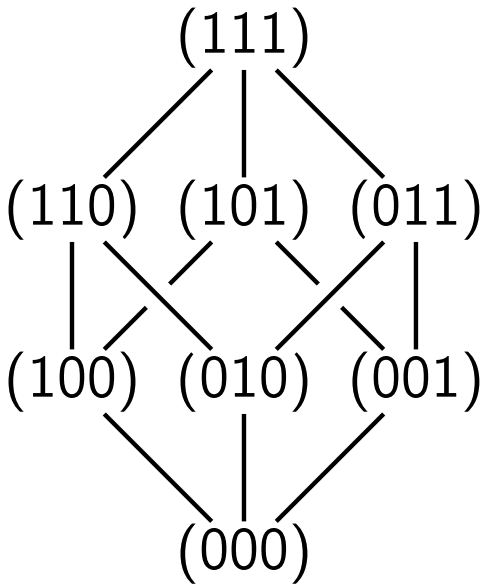
Background on pattern mining

- Let $E = (e_1, \ldots, e_d)$ be any set and consider $\mathcal{X} = \mathcal{P}(E)$ the collection of all 2^d possible patterns on E.



Background on pattern mining

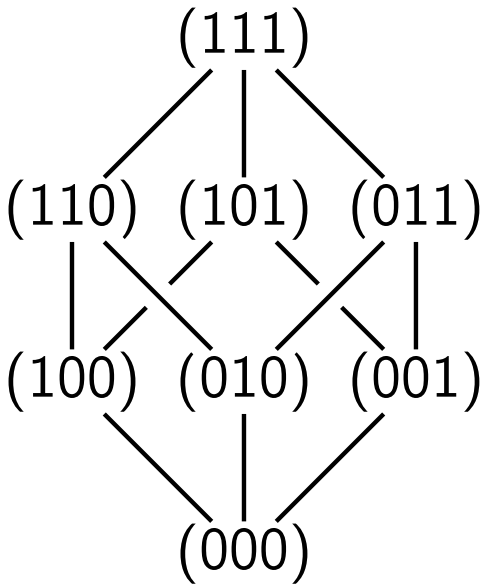
- Let $E = (e_1, \ldots, e_d)$ be any set and consider $\mathcal{X} = \mathcal{P}(E)$ the collection of all 2^d possible patterns on E.
- Consider a r.v. $X : \Omega \to \mathcal{X}$ distributed according to P and a dataset $(X_1, \ldots, X_n) \sim X$.



Sequence	Events
T_1	$\{e_1, e_2\}$
T_2	$\{e_2\}$
T_3	$\{e_1, e_2, e_3\}$
T_4	$\{e_1, e_3\}$
T_5	$\{e_2, e_3\}$

Background on pattern mining

- Let $E = (e_1, \ldots, e_d)$ be any set and consider $\mathcal{X} = \mathcal{P}(E)$ the collection of all 2^d possible patterns on E.
- Consider a r.v. $X : \Omega \to \mathcal{X}$ distributed according to P and a dataset $(X_1, \ldots, X_n) \sim X$.
- Let's compute the support

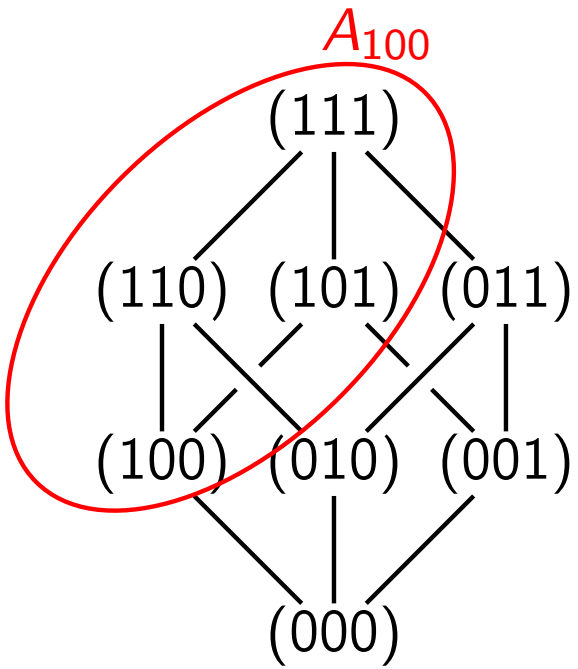


Sequence	Events
T_1	$\{e_1, e_2\}$
T_2	$\{e_2\}$
T_3	$\{e_1, e_2, e_3\}$
T_4	$\{e_1, e_3\}$
T_5	$\{e_2, e_3\}$

$\mathcal{X}$	Supp
$\{e_1\}$	
$\{e_2\}$	
$\vdots$	$\vdots$
$\{e_1, e_2\}$	
$\{e_1, e_2, e_3\}$	

Background on pattern mining

- Let $E = (e_1, \ldots, e_d)$ be any set and consider $\mathcal{X} = \mathcal{P}(E)$ the collection of all 2^d possible patterns on E.
- Consider a r.v. $X : \Omega \to \mathcal{X}$ distributed according to P and a dataset $(X_1, \ldots, X_n) \sim X$.
- Let's compute the support

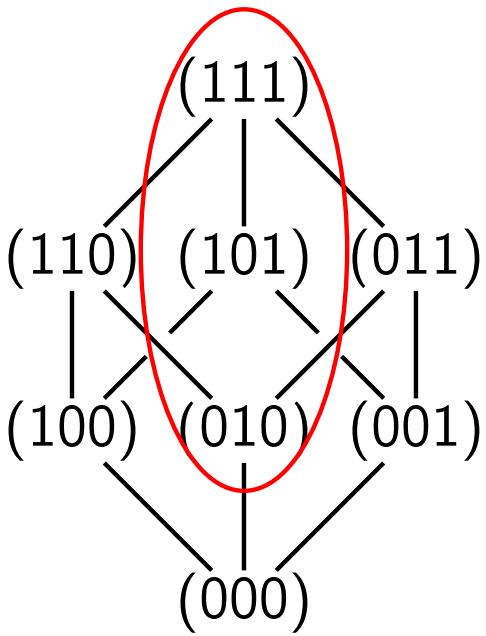


Sequence	Events
T_1	$\{e_1, e_2\}$
T_2	$\{e_2\}$
T_3	$\{e_1, e_2, e_3\}$
T_4	$\{e_1, e_3\}$
T_5	$\{e_2, e_3\}$

$\mathcal{X}$	Supp
$\{e_1\}$	3
$\{e_2\}$	
$\vdots$	$\vdots$
$\{e_1, e_2\}$	
$\{e_1, e_2, e_3\}$	

Background on pattern mining

- Let $E = (e_1, \ldots, e_d)$ be any set and consider $\mathcal{X} = \mathcal{P}(E)$ the collection of all 2^d possible patterns on E.
- Consider a r.v. $X : \Omega \to \mathcal{X}$ distributed according to P and a dataset $(X_1, \ldots, X_n) \sim X$.
- Let's compute the support

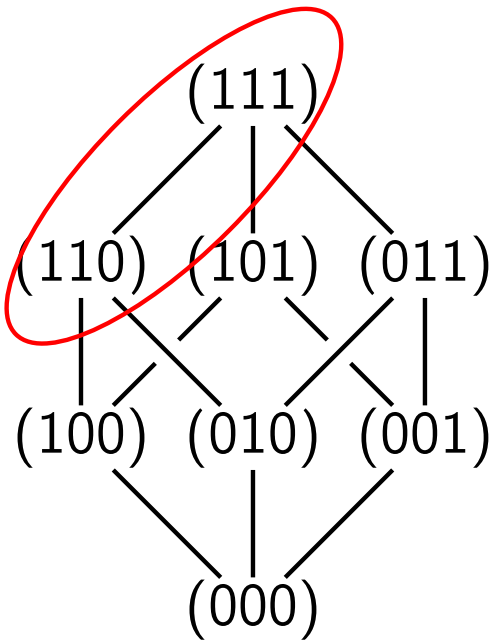


Sequence	Events
T_1	$\{e_1, e_2\}$
T_2	$\{e_2\}$
T_3	$\{e_1, e_2, e_3\}$
T_4	$\{e_1, e_3\}$
T_5	$\{e_2, e_3\}$

$\mathcal{X}$	Supp
$\{e_1\}$	3
$\{e_2\}$	4
$\vdots$	$\vdots$
$\{e_1, e_2\}$	
$\{e_1, e_2, e_3\}$	

Background on pattern mining

- Let $E = (e_1, \ldots, e_d)$ be any set and consider $\mathcal{X} = \mathcal{P}(E)$ the collection of all 2^d possible patterns on E.
- Consider a r.v. $X : \Omega \to \mathcal{X}$ distributed according to P and a dataset $(X_1, \ldots, X_n) \sim X$.
- Let's compute the support

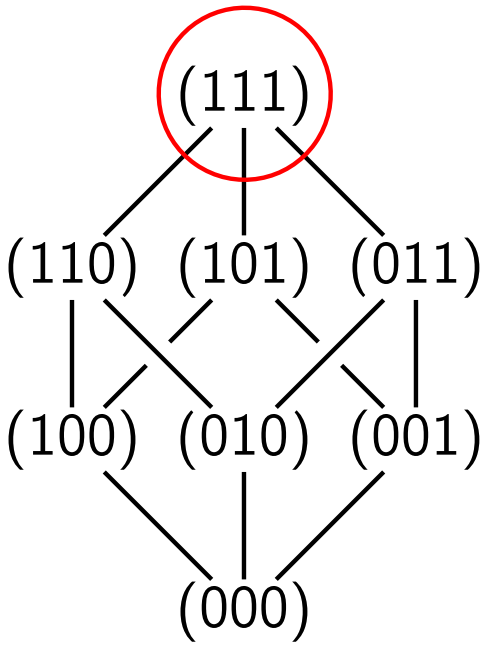


Sequence	Events
T_1	$\{e_1, e_2\}$
T_2	$\{e_2\}$
T_3	$\{e_1, e_2, e_3\}$
T_4	$\{e_1, e_3\}$
T_5	$\{e_2, e_3\}$

$\mathcal{X}$	Supp
$\{e_1\}$	3
$\{e_2\}$	4
$\vdots$	$\vdots$
$\{e_1, e_2\}$	2
$\{e_1, e_2, e_3\}$	

Background on pattern mining

- Let $E = (e_1, \ldots, e_d)$ be any set and consider $\mathcal{X} = \mathcal{P}(E)$ the collection of all 2^d possible patterns on E.
- Consider a r.v. $X : \Omega \to \mathcal{X}$ distributed according to P and a dataset $(X_1, \ldots, X_n) \sim X$.
- Let's compute the support

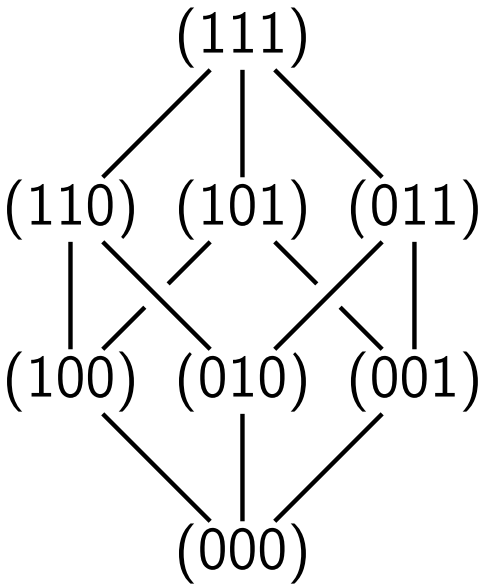


Sequence	Events
T_1	$\{e_1, e_2\}$
T_2	$\{e_2\}$
T_3	$\{e_1, e_2, e_3\}$
T_4	$\{e_1, e_3\}$
T_5	$\{e_2, e_3\}$

$\mathcal{X}$	Supp
$\{e_1\}$	3
$\{e_2\}$	4
$\vdots$	$\vdots$
$\{e_1, e_2\}$	2
$\{e_1, e_2, e_3\}$	1

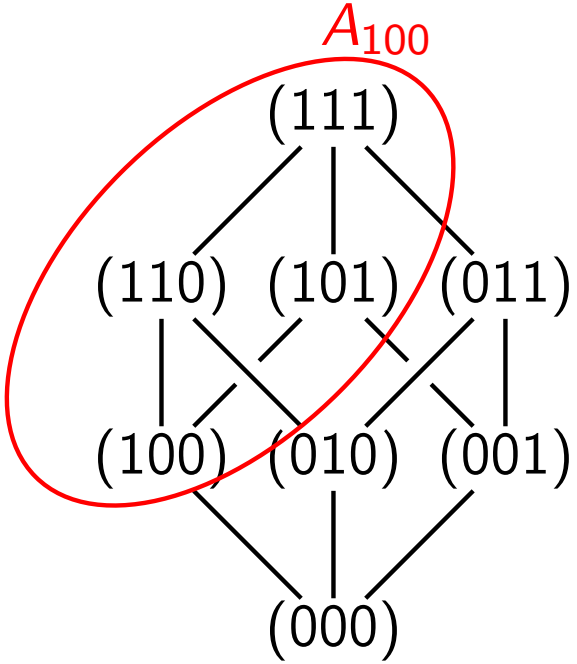
Background on pattern mining

- Let $E = (e_1, \ldots, e_d)$ be any set and consider $\mathcal{X} = \mathcal{P}(E)$ the collection of all 2^d possible patterns on E.
- Consider a r.v. $X : \Omega \to \mathcal{X}$ distributed according to P and a dataset $(X_1, \ldots, X_n) \sim X$.
- Let $A_t = \{z \in \mathcal{X} : z \supseteq t\}$ the set of all itemsets greater than $t \in \mathcal{X}$, $f_t(.) = \mathbb{1}_{.\in A_t}$ and the associated func family $\mathcal{F} = \{f_t : t \in \mathcal{X}\}$. The support of any pattern t is given by $s(t) = \mathbb{E}[\mathbb{1}_{X \in A_t}] = Pf_t$.



Background on pattern mining

- Let $E = (e_1, \ldots, e_d)$ be any set and consider $\mathcal{X} = \mathcal{P}(E)$ the collection of all 2^d possible patterns on E.
- Consider a r.v. $X : \Omega \to \mathcal{X}$ distributed according to P and a dataset $(X_1, \ldots, X_n) \sim X$.
- Let $A_t = \{z \in \mathcal{X} : z \supseteq t\}$ the set of all itemsets greater than $t \in \mathcal{X}$, $f_t(.) = \mathbb{1}_{.\in A_t}$ and the associated func family $\mathcal{F} = \{f_t : t \in \mathcal{X}\}$. The support of any pattern t is given by $s(t) = \mathbb{E}[\mathbb{1}_{X \in A_t}] = Pf_t$.



Example

$$s(e_1) = \mathbb{E}[A_{100}] = Pf_{100}$$

Motivations

The problem can now be stated as follow

Problem statement

Let $E = (e_1, \ldots, e_d)$ be any set and $\mathcal{X} = \mathcal{P}(E)$ be the set of patterns on E. Consider data generated by a r.v. $X : \Omega \to \mathcal{X}$ distributed according to P. For any $x \in \mathcal{X}$, compute

$$r = \frac{P(x|Y = 0)}{P(x|Y = 1)}. \tag{1}$$

Motivations

The problem can now be stated as follow

Problem statement

Let $E = (e_1, \ldots, e_d)$ be any set and $\mathcal{X} = \mathcal{P}(E)$ be the set of patterns on E. Consider data generated by a r.v. $X : \Omega \to \mathcal{X}$ distributed according to P. For any $x \in \mathcal{X}$, compute

$$r = \frac{P(x|Y=0)}{P(x|Y=1)}. \tag{1}$$

To compute it, we can

- Discriminative pattern mining using a generative model for each subclass (Dib et al., 2021)
- Take a subsample of the dataset and bound the expect support with classical machine learning tools (Cousins* and Dib*, 2021).

Motivations

Recall that the support of a pattern is given by Pf_t and the empirical support is denoted $P_n f_t$. The problem can be reformulated as bounding the supremum deviation of an empirical process.

Problématique

Let $\mathcal{X}$ be a set and $\mathcal{F} = \{f_t : t \in \mathcal{X}\}$ a functional class indexed on $\mathcal{X}$ and an $\epsilon \in [0, 1]$. With probability $1 - \delta$, we require that

$$\mathcal{S}_n \mathcal{F} = \sup_{t \in \mathcal{X}} \left| \hat{P}_n f_t - P f_t \right| \leq \epsilon. \tag{2}$$

Motivations

Recall that the support of a pattern is given by Pf_t and the empirical support is denoted $P_n f_t$. The problem can be reformulated as bounding the supremum deviation of an empirical process.

Problématique

Let $\mathcal{X}$ be a set and $\mathcal{F} = \{f_t : t \in \mathcal{X}\}$ a functional class indexed on $\mathcal{X}$ and an $\epsilon \in [0, 1]$. With probability $1 - \delta$, we require that

$$\mathcal{S}_n\mathcal{F} = \sup_{t \in \mathcal{X}} \left| \hat{P}_n f_t - Pf_t \right| \leq \epsilon. \tag{2}$$

Several contributions have been made recently to the topic of probabilistic bound for pattern mining using various methods of the toolbox of statistical learning theory such as using Massart's lemma (Riondato and Upfal, 2015a), VC dimension (Riondato and Upfal, 2015b) or Monte Carlo (Global) Rademacher averages (Pellegrina et al., 2020).

Background on rademacher complexity based bounds: the global rademacher complexity

Left $\mathcal{F}$ be any real-valued functional space, $(X_1, \ldots, X_n)$ an sample of size n drawn from the underlying and unknown distribution P and $(\sigma_1, \ldots, \sigma_n)$ a set of rademacher variables.

$$R_n(\mathcal{F}, x) = \mathop{\mathbb{E}}_{x}\left[\mathop{\mathbb{E}}_{\boldsymbol{\sigma}}\left[\sup_{f\in\mathcal{F}}\left|\frac{1}{n}\sum_{i=1}^{n}\boldsymbol{\sigma}_i f(x_i)\right|\right]\right] \qquad \text{(Empirical Rad. average)}$$

$$R_n(\mathcal{F}) = \mathop{\mathbb{E}}_{x}\left[\mathop{\mathbb{E}}_{\boldsymbol{\sigma}}\left[\sup_{f\in\mathcal{F}}\left|\frac{1}{n}\sum_{i=1}^{n}\boldsymbol{\sigma}_i f(x_i)\right|\right]\right] \qquad \text{(Rademacher average)}$$

Background on rademacher complexity based bounds: the global rademacher complexity

Left $\mathcal{F}$ be any real-valued functional space, $(X_1, \ldots, X_n)$ an sample of size n drawn from the underlying and unknown distribution P and $(\sigma_1, \ldots, \sigma_n)$ a set of rademacher variables. Let $\mathcal{F}_r = \{f \in \mathcal{F}; T(f) \leq r\}$

$$R_n(\mathcal{F}_r, x) = \mathop{\mathbb{E}}_{x}\left[\mathop{\mathbb{E}}_{\boldsymbol{\sigma}}\left[\sup_{f \in \mathcal{F}_r}\left|\frac{1}{n}\sum_{i=1}^{n}\boldsymbol{\sigma}_i f(x_i)\right|\right]\right] \qquad \text{(Local Empirical Rad Av.)}$$

$$R_n(\mathcal{F}_r) = \mathop{\mathbb{E}}_{x}\left[\mathop{\mathbb{E}}_{\boldsymbol{\sigma}}\left[\sup_{f \in \mathcal{F}_r}\left|\frac{1}{n}\sum_{i=1}^{n}\boldsymbol{\sigma}_i f(x_i)\right|\right]\right] \qquad \text{(Local Rademacher Av.)}$$

Background

Using Talgrand's inequality it can be shown the following distribution free uniform bound

Theorem

Let $\mathcal{F}$ be a functional family, $(x_1, \ldots, x_n)$ a i.i.d. sample of size n drawn from P. With probability $1 - \delta$

$$\left(P - \hat{P}_n\right) f \leq 8R_n(\mathcal{F}) + \Sigma(\mathcal{F})\sqrt{\frac{8\log\frac{2}{\delta}}{n}} + \frac{3\log\frac{2}{\delta}}{n}, \tag{3}$$

where $\Sigma^2(\mathcal{F}) := \sup_{f\in\mathcal{F}} \mathbb{E}\left[f^2\right]$ is a bound on the variance of the functions in $\mathcal{F}$.

Background

Using Talgrand's inequality it can be shown the following distribution free uniform bound

Theorem

Let $\mathcal{F}$ be a functional family, $(x_1, \ldots, x_n)$ a i.i.d. sample of size n drawn from P. With probability $1 - \delta$

$$\left(P - \hat{P}_n\right) f \leq 8R_n(\mathcal{F}) + \Sigma(\mathcal{F})\sqrt{\frac{8 \log \frac{2}{\delta}}{n}} + \frac{3 \log \frac{2}{\delta}}{n}, \tag{3}$$

where $\Sigma^2(\mathcal{F}) := \sup_{f \in \mathcal{F}} \mathbb{E}\left[f^2\right]$ is a bound on the variance of the functions in $\mathcal{F}$.

Can we drop the uniform bound for a variance dependend one, allowing for use of localized complexities measures of $\mathcal{F}$ to obtain fast rates ?

Background on rademacher complexity based bounds: the local rademacher complexity

(Bartlett, Bousquet, and Mendelson, 2005) the following non uniform bound based on LRA

Theorem

Assume that ψ is a sub-root function, i.e., $\psi(r;\delta)/\sqrt{r}$ is non-increasing with respect to $r \in \mathbb{R}_+$. Assume the Bernstein condition that $T(f) \leq B_e P f$ for all $f \in \mathcal{F}$. Then with probability at least $1-\delta$, for all $f \in \mathcal{F}$ and $K > 1$,

$$(P - P_n) f \leq \frac{1}{K} P f + \frac{100(K-1) r^*}{B_e}$$

where r^* is the "fixed point" solution of the equation $r = B_e \psi(r;\delta)$.

Main result: a localized bound for the pattern mining problem

In the context of pattern mining, we can establish that (Cousins* and Dib*, 2021)

Proposition (Monte-Carlo Localization Bounds)

Consider the fixed point $r^{\mathrm{U}}(K)$ function of the empricial rademacher average. With probability at least $1-\delta$ and for a function $f \in \mathcal{F}$ we have

$$Pf \geq \sup_{K>0} \min \left\{ \frac{K}{K+1} \hat{P}_n f, \hat{P}_n f - \frac{r^{\mathrm{U}}(K)}{K} \right\},$$

$$Pf \leq \inf_{K>1} \max \left\{ \frac{K}{K-1} \hat{P}_n f, \hat{P}_n f + \frac{r^{\mathrm{U}}(K)}{K} \right\} .$$

Experiment

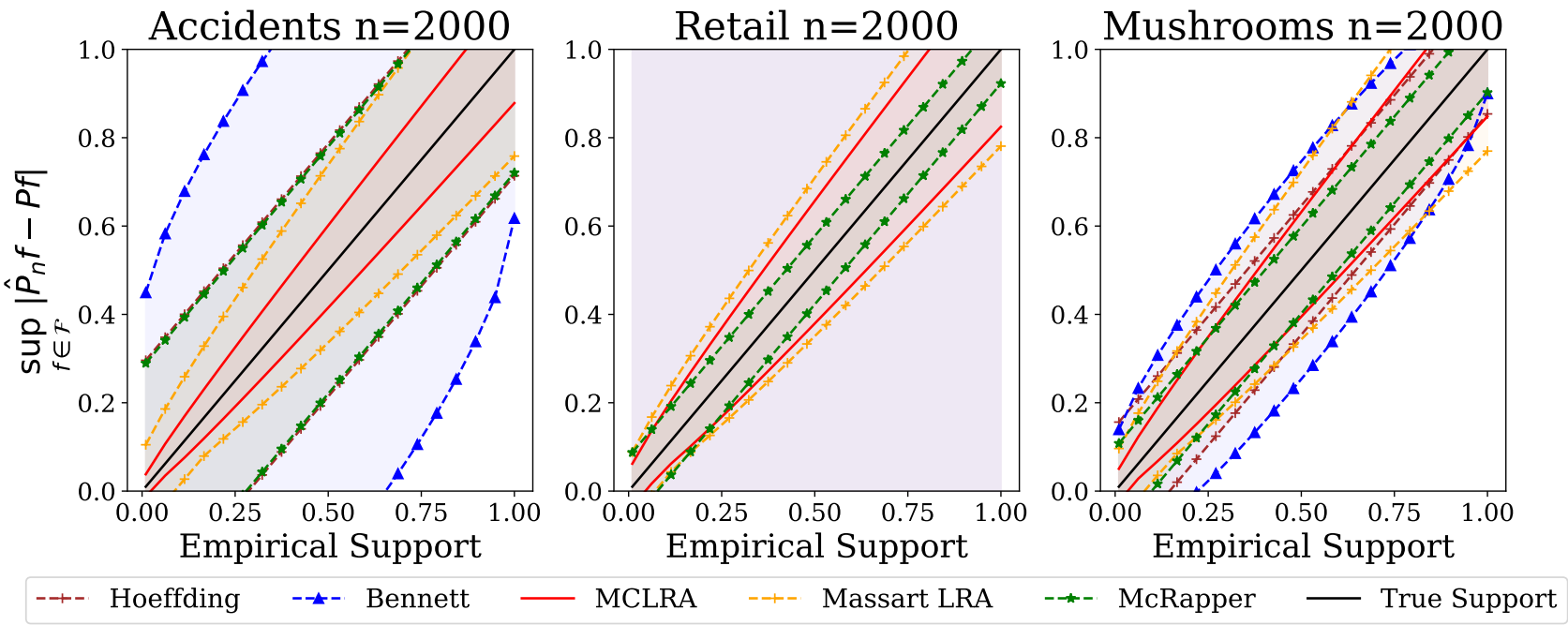


Figure: Experimental comparison of upper and lower bounds (y-axis) given empirical frequencies (x-axis), of our method to existing work on real-world datasets.

Conclusion

- First use of localization in the context of pattern mining;

Conclusion

- First use of localization in the context of pattern mining;
- We showed that using localized complexity allows to bound small variance itemset more tightly than previous methods;

Conclusion

- First use of localization in the context of pattern mining;
- We showed that using localized complexity allows to bound small variance itemset more tightly than previous methods;
- We designed a geometrical approach allowing to compute the fixed point the localized rademacher average in the context of pattern mining;

Conclusion

- First use of localization in the context of pattern mining;
- We showed that using localized complexity allows to bound small variance itemset more tightly than previous methods;
- We designed a geometrical approach allowing to compute the fixed point the localized rademacher average in the context of pattern mining;
- The approach is tested empiracally and shows better convergence behavior for small patterns than state of the art methods.

Descriminative pattern mining problem as a stochastic optimization problem

Recall that for a set $\mathcal{X}$ be the set of patterns on E and a r.v. $X : \Omega \to \mathcal{X}$ distributed according to P, our goal is to compute for following quantity for any pattern $x \in \mathcal{X}$

$$r = \frac{P(x|Y=0)}{P(x|Y=1)}.$$

Descriminative pattern mining problem as a stochastic optimization problem

Recall that for a set $\mathcal{X}$ be the set of patterns on E and a r.v. $X : \Omega \to \mathcal{X}$ distributed according to P, our goal is to compute for following quantity for any pattern $x \in \mathcal{X}$

$$r = \frac{P(x|Y=0)}{P(x|Y=1)}.$$

It can be shown (Dib et al., 2021) that this problem can be reformulated as an optimization when you minimize an objective function of the type

$$\mathcal{L}(\boldsymbol{\lambda}) = \mathbb{E}\Big[F(X^{\boldsymbol{\lambda}})\Big], \tag{4}$$

with $\boldsymbol{\lambda} \in \mathbb{R}^K$ and possibily very large K.

Descriminative pattern mining problem as a stochastic optimization problem

Recall that for a set $\mathcal{X}$ be the set of patterns on E and a r.v. $X : \Omega \to \mathcal{X}$ distributed according to P, our goal is to compute for following quantity for any pattern $x \in \mathcal{X}$

$$r = \frac{P(x|Y=0)}{P(x|Y=1)}.$$

It can be shown (Dib et al., 2021) that this problem can be reformulated as an optimization when you minimize an objective function of the type

$$\mathcal{L}(\boldsymbol{\lambda}) = \mathbb{E}\Big[F(X^{\boldsymbol{\lambda}})\Big], \tag{4}$$

with $\boldsymbol{\lambda} \in \mathbb{R}^K$ and possibily very large K.

How can we speed up such inference ?

Stochastic optimization procedure

Given a sample $(X_1, \ldots, X_N)$ of size N, typical MCVI consists of a gradient descent at each step k

$$\lambda_{k+1} = \lambda_k - \alpha_k \underbrace{\frac{1}{N} \sum_{i=1}^{N} \nabla_\lambda F\left(X_i^{\lambda_k}\right)}_{\widehat{g}_{\mathrm{MC}}^N}.$$

Stochastic optimization procedure

Given a sample $(X_1, \ldots, X_N)$ of size N, typical MCVI consists of a gradient descent at each step k

$$\lambda_{k+1} = \lambda_k - \alpha_k \underbrace{\frac{1}{N} \sum_{i=1}^{N} \nabla_\lambda F\left(X_i^{\lambda_k}\right)}_{\widehat{g}_{\mathrm{MC}}^N}.$$

Gradient descent descent convergence speed crucially depends on the following (Bottou, Curtis, and Nocedal, 2018) quantity

$$\mathbb{E}|g|_{\ell_2}^2 = \operatorname{tr} \mathbb{V} g + |\mathbb{E} g|_{\ell_2}^2.$$

Stochastic optimization procedure

Is there ways to reduce the gradient variance ?

- Modify the gradient formula to reduce the variance (Miller et al., 2017; Roeder, Wu, and Duvenaud, 2017);
- Control variate (Geffner and Domke, 2018);
- Alternative sampling (Pagès, 2015; Buchholz, Wenzel, and Mandt, 2018; Tran, Nott, and Kohn, 2017; Ruiz, Titsias, and Blei, 2016).

Stochastic optimization procedure

Is there ways to reduce the gradient variance ?

- Modify the gradient formula to reduce the variance (Miller et al., 2017; Roeder, Wu, and Duvenaud, 2017);
- Control variate (Geffner and Domke, 2018);
- Alternative sampling (Pagès, 2015; Buchholz, Wenzel, and Mandt, 2018; Tran, Nott, and Kohn, 2017; Ruiz, Titsias, and Blei, 2016).

What if we want variance-free gradient ?

The optimal quantizer

- Let $X : (\Omega, \mathcal{F}) \longrightarrow (\mathbb{R}^d, \mathcal{B}(\mathbb{R}^d))$ be a r.v. with finite p moments

The optimal quantizer

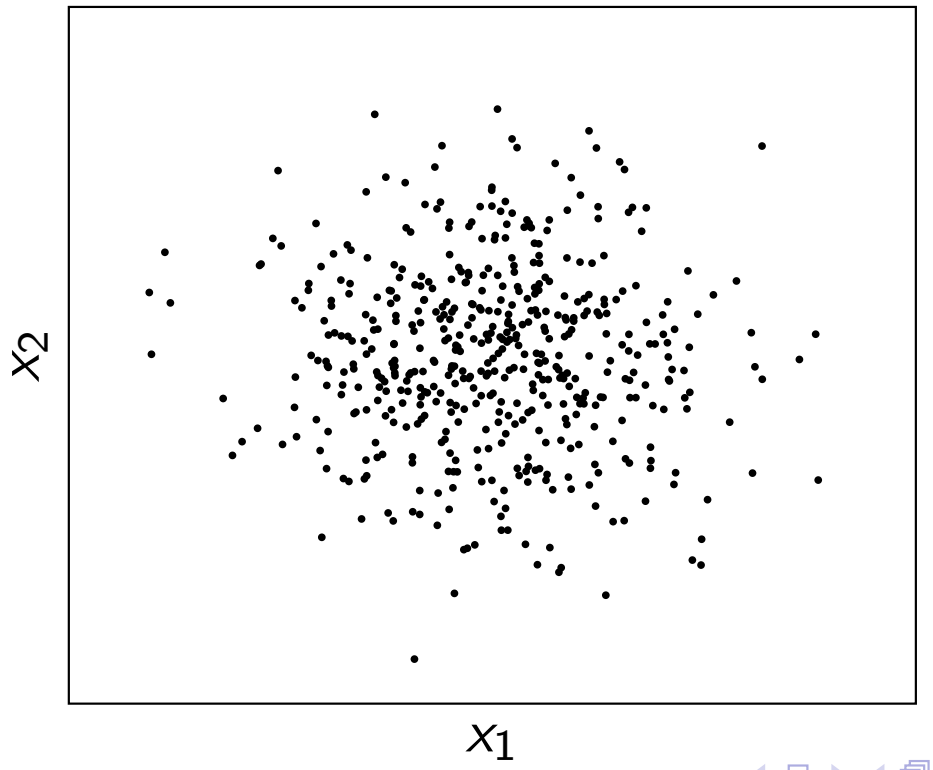
- Let $X : (\Omega, \mathcal{F}) \longrightarrow (\mathbb{R}^d, \mathcal{B}(\mathbb{R}^d))$ be a r.v. with finite p moments
- **Goal**: we want to find the best r.v. $\hat{X}$ with finite support $\Gamma \subset \mathbb{R}^d$ to replace X. Let $q : \mathbb{R}^d \to \Gamma$ be the function s.t. $q(X) = \hat{X}$.

The optimal quantizer

- Let $X : (\Omega, \mathcal{F}) \longrightarrow (\mathbb{R}^d, \mathcal{B}(\mathbb{R}^d))$ be a r.v. with finite p moments
- **Goal**: we want to find the best r.v. $\hat{X}$ with finite support $\Gamma \subset \mathbb{R}^d$ to replace X. Let $q : \mathbb{R}^d \to \Gamma$ be the function s.t. $q(X) = \hat{X}$.
- Example: take the following samples. $|\Gamma| = 1$?

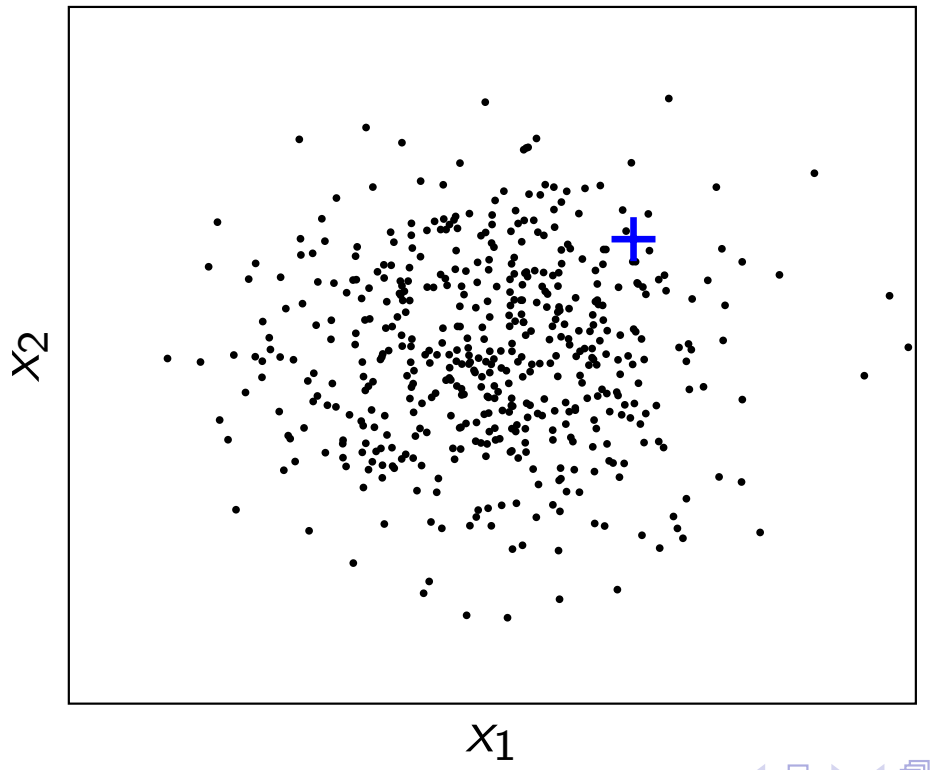
The optimal quantizer

- Let $X : (\Omega, \mathcal{F}) \longrightarrow (\mathbb{R}^d, \mathcal{B}(\mathbb{R}^d))$ be a r.v. with finite p moments
- **Goal**: we want to find the best r.v. $\hat{X}$ with finite support $\Gamma \subset \mathbb{R}^d$ to replace X. Let $q : \mathbb{R}^d \to \Gamma$ be the function s.t. $q(X) = \hat{X}$.
- Example: take the following samples. What would be the optimal choice for $|\Gamma| = 1$?



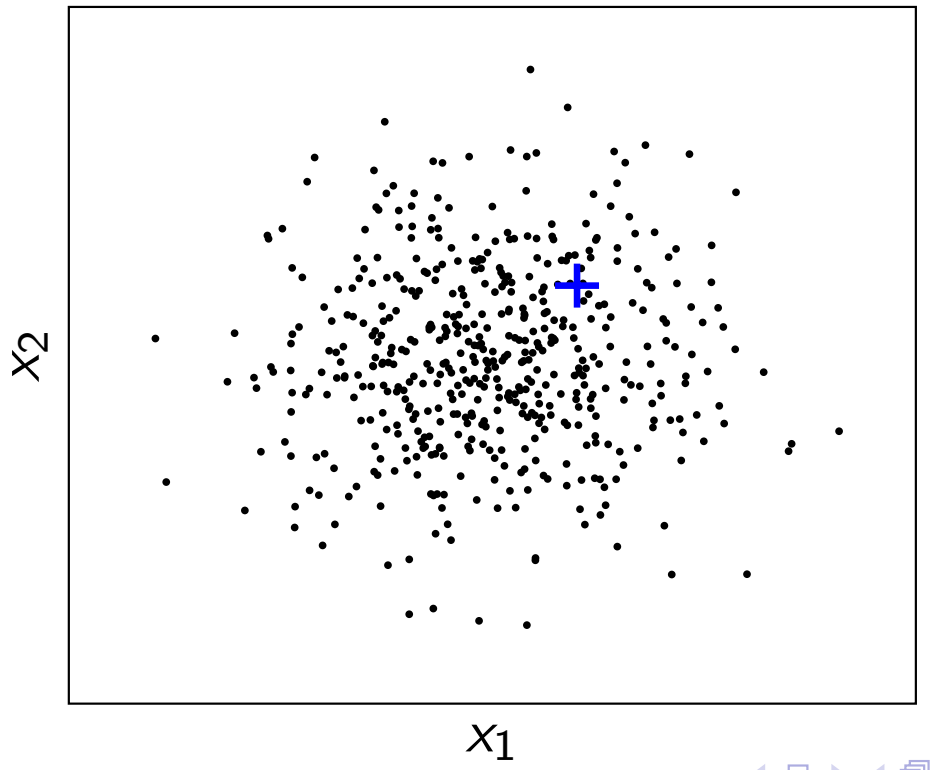
The optimal quantizer

- Let $X : (\Omega, \mathcal{F}) \longrightarrow (\mathbb{R}^d, \mathcal{B}(\mathbb{R}^d))$ be a r.v. with finite p moments
- **Goal**: we want to find the best r.v. $\hat{X}$ with finite support $\Gamma \subset \mathbb{R}^d$ to replace X. Let $q : \mathbb{R}^d \to \Gamma$ be the function s.t. $q(X) = \hat{X}$.
- Example: take the following samples. What would be the optimal choice for $|\Gamma| = 1$?



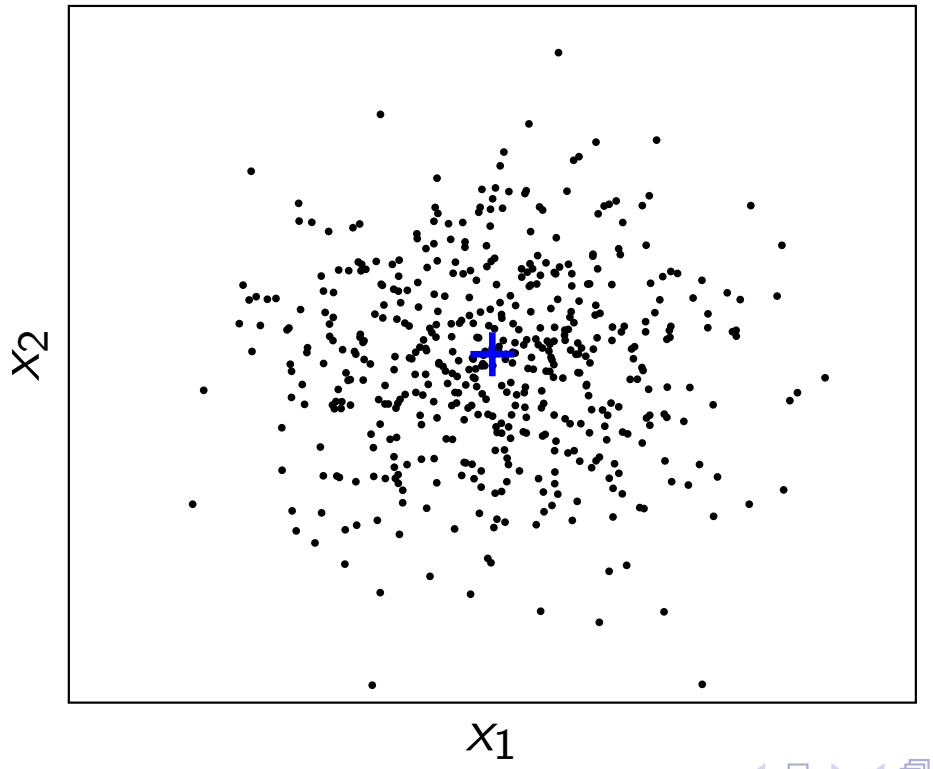
The optimal quantizer

- Let $X : (\Omega, \mathcal{F}) \longrightarrow (\mathbb{R}^d, \mathcal{B}(\mathbb{R}^d))$ be a r.v. with finite p moments
- **Goal**: we want to find the best r.v. $\hat{X}$ with finite support $\Gamma \subset \mathbb{R}^d$ to replace X. Let $q : \mathbb{R}^d \to \Gamma$ be the function s.t. $q(X) = \hat{X}$.
- Example: take the following samples. What would be the optimal choice for $|\Gamma| = 1$?



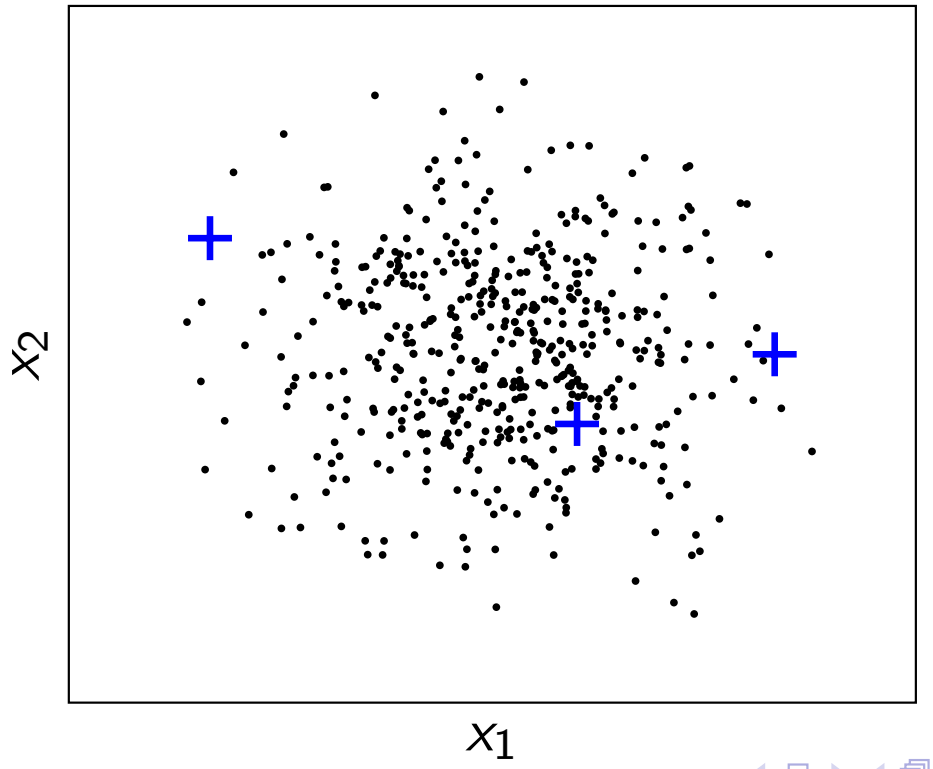
The optimal quantizer

- Let $X : (\Omega, \mathcal{F}) \longrightarrow (\mathbb{R}^d, \mathcal{B}(\mathbb{R}^d))$ be a r.v. with finite p moments
- **Goal**: we want to find the best r.v. $\hat{X}$ with finite support $\Gamma \subset \mathbb{R}^d$ to replace X. Let $q : \mathbb{R}^d \to \Gamma$ be the function s.t. $q(X) = \hat{X}$.
- Example: take the following samples. What would be the optimal choice for $|\Gamma| = 1$?



The optimal quantizer

- Let $X : (\Omega, \mathcal{F}) \longrightarrow (\mathbb{R}^d, \mathcal{B}(\mathbb{R}^d))$ be a r.v. with finite p moments
- **Goal**: we want to find the best r.v. $\hat{X}$ with finite support $\Gamma \subset \mathbb{R}^d$ to replace X. Let $q : \mathbb{R}^d \to \Gamma$ be the function s.t. $q(X) = \hat{X}$.
- Example: take the following samples. What would be the optimal choice for $|\Gamma| = 3$?



The optimal quantizer

What is the best q function to minize the pointwise distance ?

The optimal quantizer

What is the best q function to minize the pointwise distance ?

- Consider the voronoi cells associated with $\Gamma = (x_1, \ldots, x_N)$ such that

$$V(x_i, \Gamma) = \left\{ z \in \mathbb{R}^d : |z - x_i| = \min_{x \in \Gamma} |z - x| \right\}.$$

The optimal quantizer

What is the best q function to minize the pointwise distance ?

- Consider the voronoi cells associated with $\Gamma = (x_1, \ldots, x_N)$ such that

$$V(x_i, \Gamma) = \left\{ z \in \mathbb{R}^d : |z - x_i| = \min_{x \in \Gamma} |z - x| \right\}.$$

- Additionaly, take the closest projection onto the Voronoi cells defined by

$$\widehat{X}_\Gamma = \Pi_\Gamma X \tag{5}$$

$$= \sum_{i=1}^{N} x_i \mathbb{1}_{X \in V(\Gamma, x_i)}. \tag{6}$$

The optimal quantizer

What is the best q function to minize the pointwise distance ?

- Consider the voronoi cells associated with $\Gamma = (x_1, \ldots, x_N)$ such that

$$V(x_i, \Gamma) = \left\{ z \in \mathbb{R}^d : |z - x_i| = \min_{x \in \Gamma} |z - x| \right\}.$$

- Additionaly, take the closest projection onto the Voronoi cells defined by

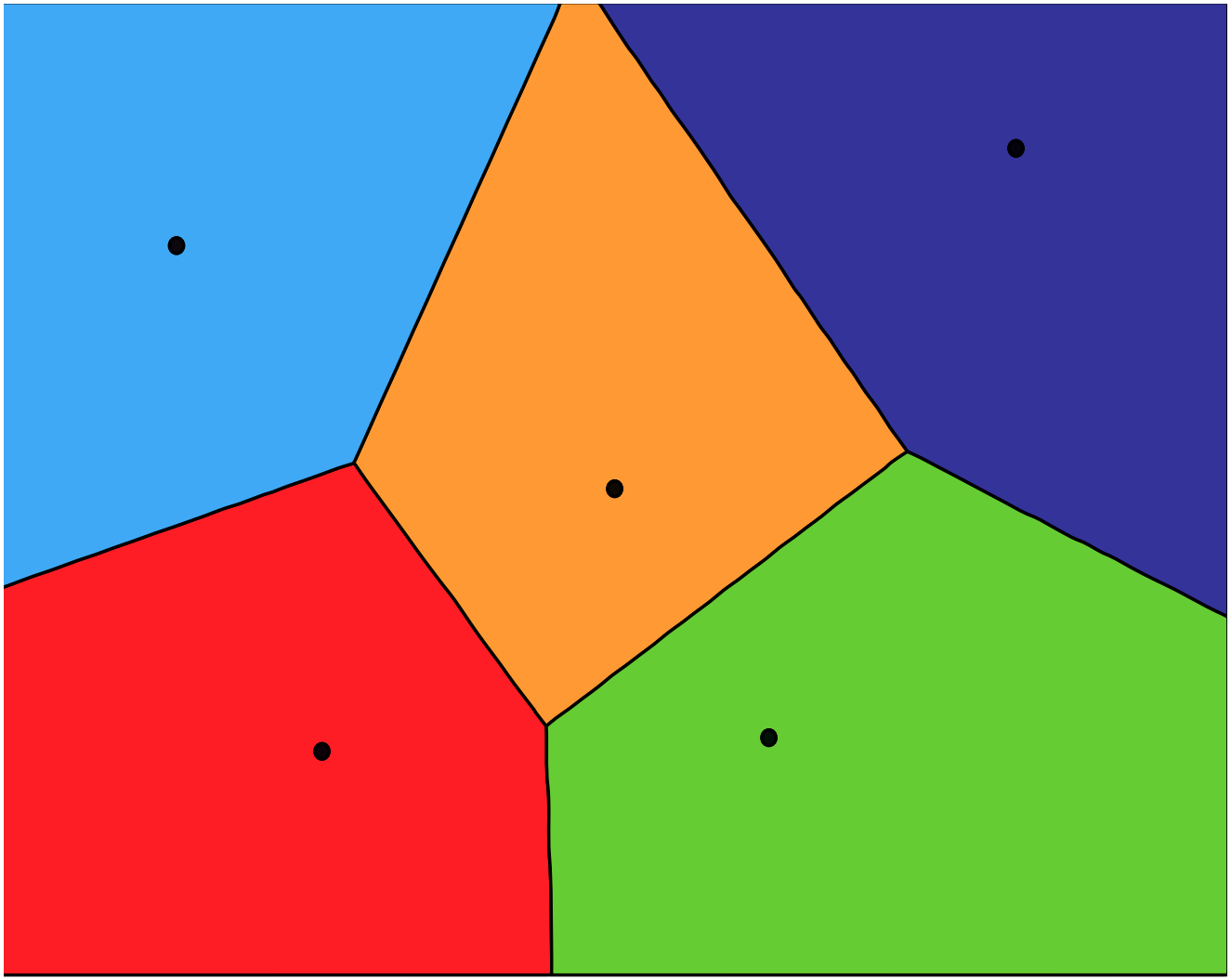
$$\widehat{X}_\Gamma = \Pi_\Gamma X \tag{5}$$

$$= \sum_{i=1}^{N} x_i \mathbb{1}_{X \in V(\Gamma, x_i)}. \tag{6}$$

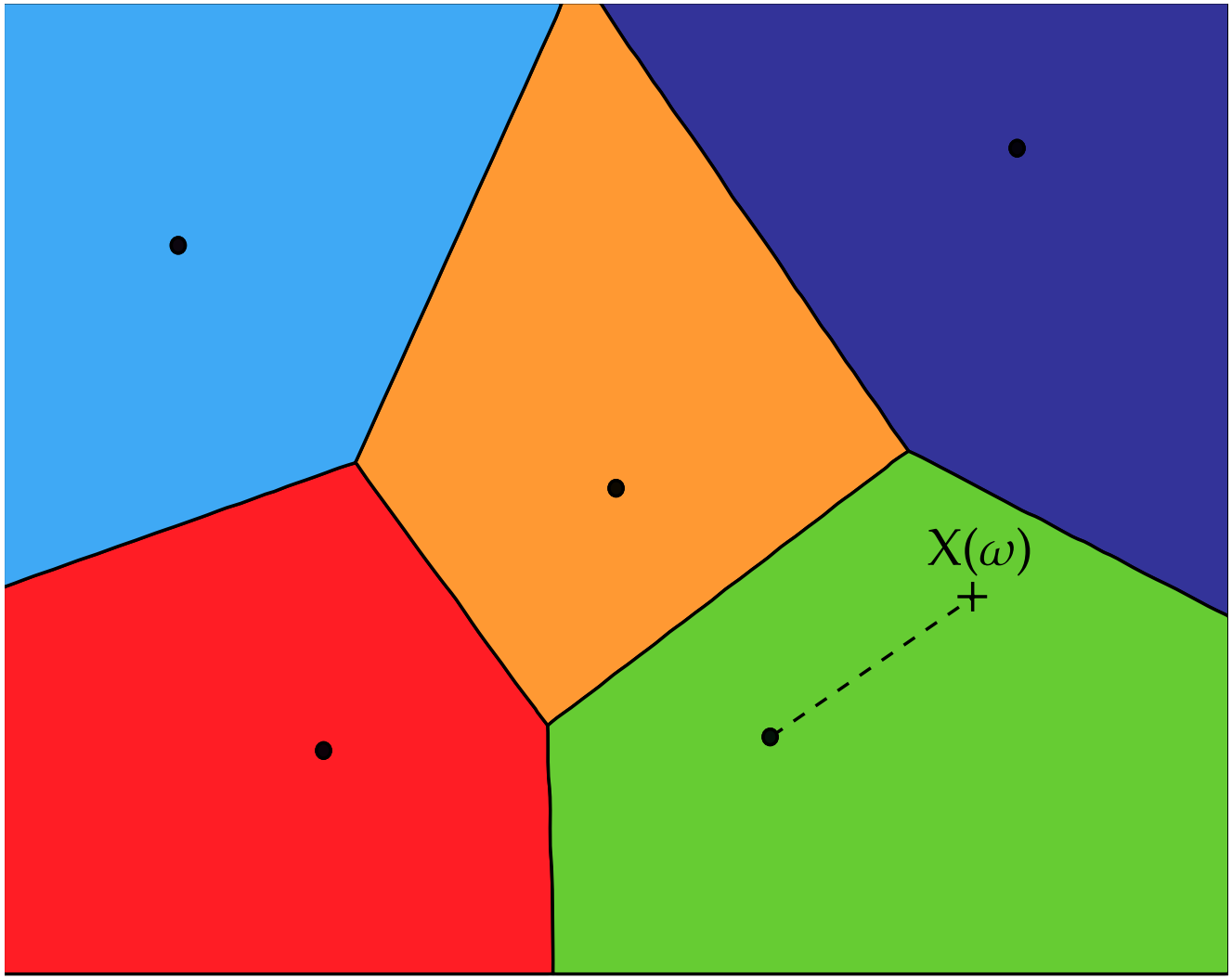
- Then we have that

$$|X - \widehat{X}_\Gamma| = \min_{x \in \Gamma} |X - x|.$$

The optimal quantizer: illustration



The optimal quantizer: illustration



The distortion function

Definition (Optimal Quantizer)

Let $X : ((\Omega, \mathcal{F})) \longrightarrow (E, \mathcal{B}(E))$ be a random variable in $L_p(\Omega, \mathcal{A}, \mathbb{P})$ with distribution μ and consider a finite subset $\Gamma \subset E$ of size n. The $L_p(\Omega, \mathcal{A}, \mathbb{P})$ *distortion function* $\mathcal{D}_{p,\mu}$ of μ at level n is defined by

$$\begin{array}{rccc} \mathcal{D}_{p,\mu} : & E^n & \longrightarrow & \mathbb{R}_+ \\ & \Gamma & \longmapsto & \mathbb{E}\left[\min_{x_i \in \Gamma} \|X - x_i\|^p\right] \end{array}, \tag{7}$$

and the *quantization error* function by

$$e_{p,\mu} = \mathcal{D}_{p,\mu}^{\frac{1}{p}}. \tag{8}$$

The minimizer of $e_{n,\mu}(\Gamma)$ is called a $L_p(\Omega, \mathcal{A}, \mathbb{P})$ *optimal quantizer of* μ *at level* n.

The optimal quantizer: normal standard distribution

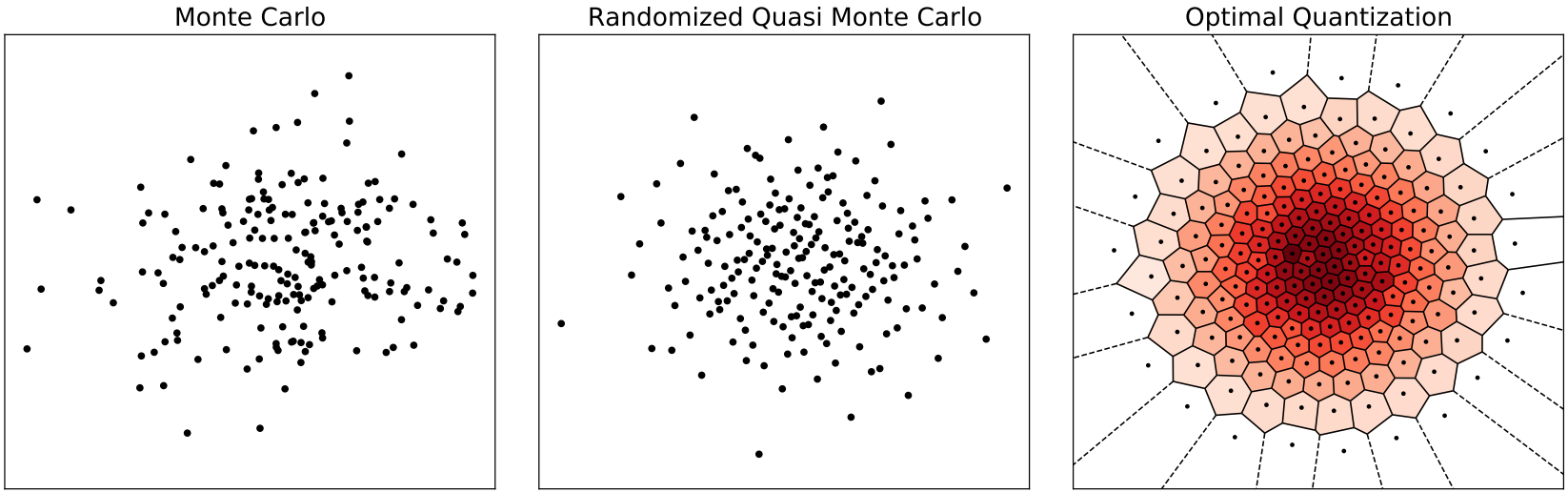


Figure: Monte Carlo (left), Randomized Monte Carlo (center) and Optimal Quantization with the associated Voronoi Cells (right), for a sampling size $N = 200$ of the bivariate normal distribution $\mathcal{N}(0, I_2)$

The optimal quantizer: the cubature formula

The key property of the optimal quantizer lays in the simplicity of his cubature formula (Pagès, 2018).

Proposition

Let $\hat{X}^N$ be a quantizer over $\Gamma_N = (x_1, \ldots, x_N)$ the optimal quantizer.For every measurable function $F(X) \in L^2_{\mathbb{R}^d}(\Omega, \mathcal{A}, \mathbb{P})$

$$\mathbb{E}\left[F\left(\hat{X}^N\right)\right] = \sum_{i=1}^{N} \omega_i F\left(x_i^N\right), \tag{9}$$

with $\omega_i = \mathbb{P}\left(\widehat{X}_\Gamma = x_i\right)$

The optimal quantizer: the cubature formula

The key property of the optimal quantizer lays in the simplicity of his cubature formula (Pagès, 2018).

Proposition

Let $\hat{X}^N$ *be a quantizer over* $\Gamma_N = (x_1, \ldots, x_N)$ *the optimal quantizer.For every measurable function* $F(X) \in L^2_{\mathbb{R}^d}(\Omega, \mathcal{A}, \mathbb{P})$

$$\mathbb{E}\big[F(\hat{X}^N)\big] = \sum_{i=1}^{N} \omega_i F\big(x_i^N\big), \tag{9}$$

with $\omega_i = \mathbb{P}\big(\widehat{X}_\Gamma = x_i\big)$

This formula allows the use of $\mathbb{E}\big[F(\hat{X}^N)\big]$ **in place of** $\mathbb{E}\big[F(X^\lambda)\big]$**.**

Quantized optimization procedure

The quantized optimization procedure considers the optimal quantization instead of the traditional MC. Precisely, Taking the optimal quantizer at level N, $X^{\Gamma_N,\lambda}$, results in the following gradient descent scheme

$$\lambda_{k+1} = \lambda_k - \alpha_k \nabla_\lambda \sum_{i=1}^{N} \omega_i^k F\left(X_i^{\Gamma_N,\lambda_k}\right), \tag{10}$$

with $\omega_i^k = \mathbb{P}\left(X^{\Gamma_N^k,\lambda_k} = x_i^k\right)$.

Quantized variational inference: the bayesian model

Given data y, a model $p(y, z)$ with latent variable z, we want to approximate the posterior distribution $p(z|y)$.
Take a a variational distribution q_λ that approximates $p(.|y)$, the following decomposition can be obtained (Saul, Jaakkola, and Jordan, 1996)

$$\log p(y) = \underbrace{\mathop{\mathbb{E}}_{z\sim q_\lambda}\left[\log \frac{p(z,y)}{q_\lambda(z)}\right]}_{\text{ELBO } \mathcal{L}(\lambda)} + \underbrace{\mathrm{KL}\left(q_\lambda(z)\|p(z|y)\right)}_{\text{KL-divergence}}. \tag{11}$$

Quantized variational inference: the bayesian model

Given data y, a model $p(y, z)$ with latent variable z, we want to approximate the posterior distribution $p(z|y)$.
Take a a variational distribution q_λ that approximates $p(.|y)$, the following decomposition can be obtained (Saul, Jaakkola, and Jordan, 1996)

$$\log p(y) = \underbrace{\mathop{\mathbb{E}}_{z\sim q_\lambda}\left[\log \frac{p(z,y)}{q_\lambda(z)}\right]}_{\text{ELBO } \mathcal{L}(\lambda)} + \underbrace{\mathrm{KL}\left(q_\lambda(z)\|p(z|y)\right)}_{\text{KL-divergence}}. \tag{11}$$

Using the reparametrization trick (Kingma, Salimans, and Welling, 2015) with noise parameter $X \sim q$ and denoting $X^\lambda = h_\lambda(X)$, the inference problem can be rewritten as finding λ^* such as

$$\lambda^* \in \operatorname{argmax} \mathbb{E}_q\left[f(X^\lambda)\right]. \tag{12}$$

Some theoretical guarantee: the Elbo quantization error

For ELBO maximization problem it can be shown that bias on the objective function is controlled by the quantization error (Dib, 2020)

Proposition

Let $\lambda^* = \min_{\lambda \in \mathbb{R}^K} \mathcal{L}(\lambda)$ *and* $\lambda_q^* = \min_{\lambda \in \mathbb{R}^K} \widehat{\mathcal{L}}_{OQ}^N(\lambda)$, *then*

$$\mathcal{L}(\lambda^*) - \widehat{\mathcal{L}}_{OQ}^N(\lambda_q^*) \leq C \left[2\|X^{\lambda^*} - X^{\Gamma,\lambda^*}\|_2 + \|X^{\lambda_q^*} - X^{\Gamma,\lambda_q^*}\|_2 \right].$$

Results: Poisson general model

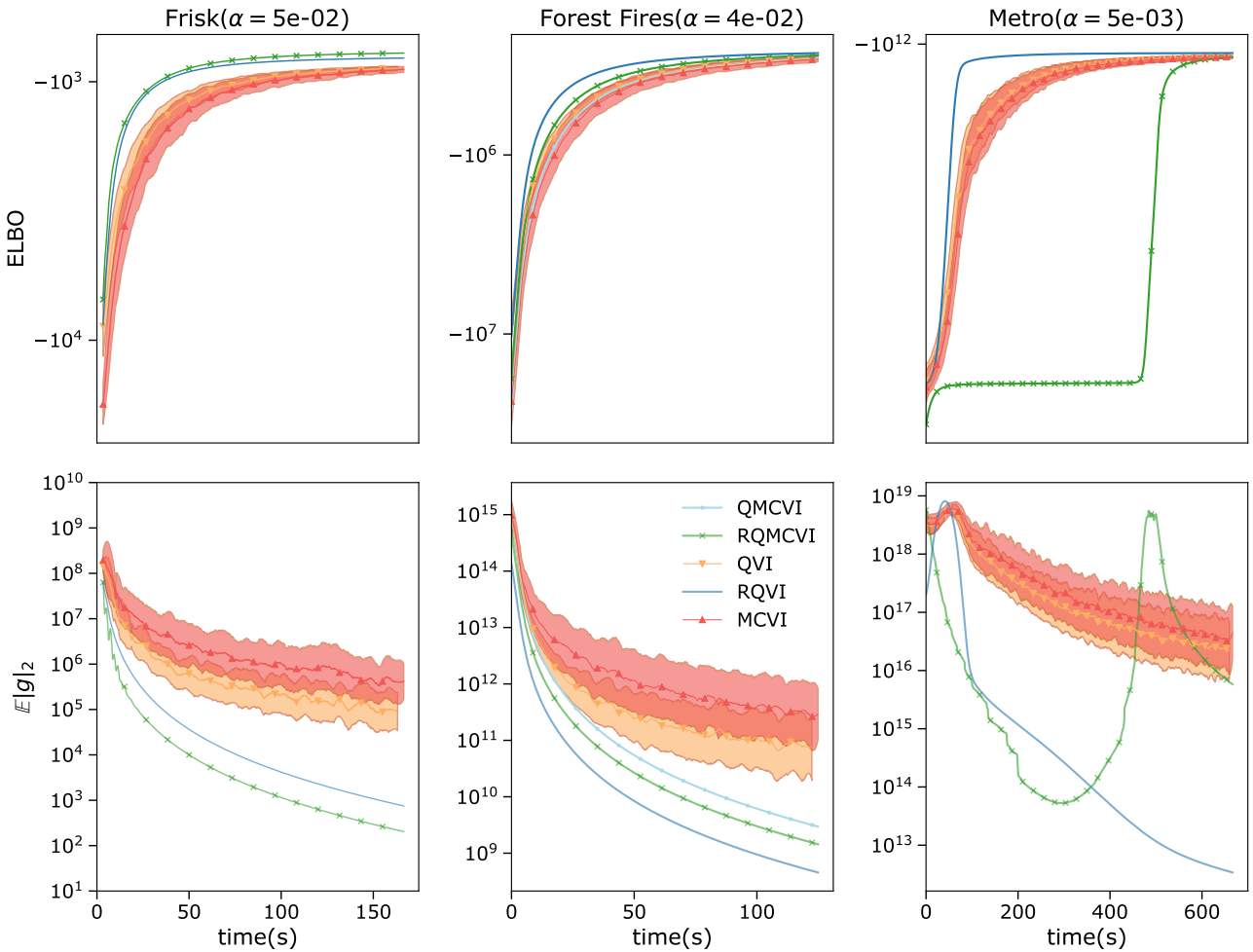


Figure: ELBO (first row, log scale) and expect gradient norm (second row, log scale) during the optimization procedure for various models: Poisson Generalized Linear Model (left), Bayesian Linear Regression (center) and Bayesian Neural

Conclusion

Advantages

- Variance free Gradient estimator allowing for swift slides ;
- Optimal Quantization is preserved through linear transformation (scaling and shifting) for large class of q_λ. Hence, to optimization need not to recompute the OQ at each steps!
- Implementation is (rather) simple with reparametrize gradient in the use case of variational inference.

Conclusion

Advantages

- Variance free Gradient estimator allowing for swift slides ;
- Optimal Quantization is preserved through linear transformation (scaling and shifting) for large class of q_λ. Hence, to optimization need not to recompute the OQ at each steps!
- Implementation is (rather) simple with reparametrize gradient in the use case of variational inference.

Limitations

- Doesn't apply to any type of prob model;
- Reducing bias is challenging (can lead to computationnal instability).

A computational pipeline for the railway french fleet

Summarize of the work

- Construction of an industrial machine learning pipeline on the real-world usecase of predictive maintenance for the french train fleet;
- Designed a two-sample based pipeline pruning to reduce drasticaly the computational requirements needed to optimize on the set of hyperparameters of the pipeline;
- Introduced a model that allow both taking into account expert knowledge and output easily interpretable results based on patterns.

Bayesian method for discriminative pattern mining

Part of this work has been published in *29th IEEE European Signal Processing Conference (EUSIPCO)* (Dib et al., 2021).

Summarize of the contribution

- New parametric approach for the discriminative pattern mining problem that allow for expert knowledge through priors;
- Design of new algorithm to enrich any classifier with discriminative patterns and showed score improvement over traditional methods on real-world use cases.

Future work and perspectives

- Improve the model by using a non parametric approach for the bernoulli mixture model using bread stick approach to replace the choice of K;
- Find new discriminative score that can be better suited.

Reproductibility. The results and figures are be fully reproductible and accessible on public repository.

Localized pattern mining

This work corresponds to the preprint (Cousins* and Dib*, 2021)[1] to be submitted.

Summarize of the contribution

- First use of localized complexity for the pattern mining problem;
- Designed a double optimization scheme to compute the bound based on empirical quantities;

Future work and perspectives

- Requires to use the set of closed itemsets above a certain treshold;
- Apply to more challeging problem such as DNA sequence classification or graph mining.

Reproductibility. The results and figures are be fully reproductible and accessible on public repository.

[1] equal contributions.

Optimal Quantization

Part of this work has been published in *Advances in Neural Information Processing Systems 33 Proceedings (NeurIPS 2020)* (Dib, 2020).

Summarize of the contribution

- A new sampling method for the general stochastic optimization problem that allow for variance free optimization;
- Proposed a new algorithm for the VI problem and showed that it can be used at comparable computational cost than MC based methods;
- Showed on real-world and challenging experiments that qvi outperforms most advanced approaches towards variance reduction;

Future work and perspectives

- Design new ways to reduce the bias;
- Apply to other frameworks such as RL (Mohamed et al., 2020);
- Use the semi-discrete optimal transport approach to contruct the optimal quantizer trough the Sliced Wasserstein distance;

Reproductibility. The results and figures are be fully reproductible and accessible on public repository.

Publications and preprint

Long life to the train !

- Amir Dib. "Quantized Variational Inference". In: *Advances in Neural Information Processing Systems* 33 (2020)
- Amir Dib et al. "Bayesian Feature Discovery for Predictive Maintenance". In: *2021 29th European Signal Processing Conference (EUSIPCO)*. IEEE, Mar. 2021
- Cyrus Cousins* and Amir Dib*. "Fast Convergence Rates for Low-Frequency Pattern Mining with Localization". In: *To Be Submitted.* 2021
- Marie Garin et al. "Epidemic Models for COVID-19 during the First Wave from February to May 2020: A Methodological Review". In: *arXiv:2109.01450 [q-bio, stat]* (Sept. 2021)

Personal page: `https://www.amirdib.com/`

Github: `https://github.com/amirdib`

From reactive to predictive maintenance

What are the strategies towards data-based maintenance ?

From reactive to predictive maintenance

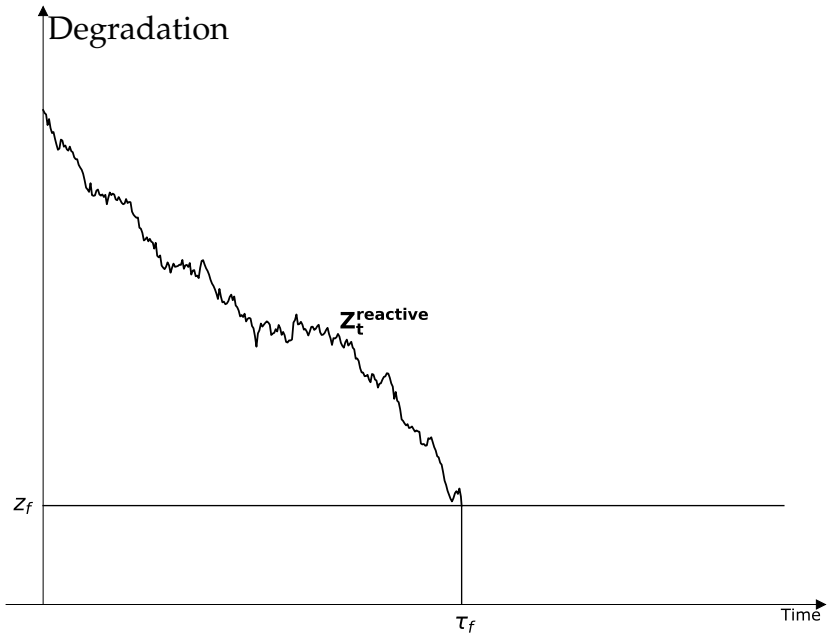
Reactive Maintenance

Maintenance is performed when equipment has failed.

From reactive to predictive maintenance

Reactive Maintenance

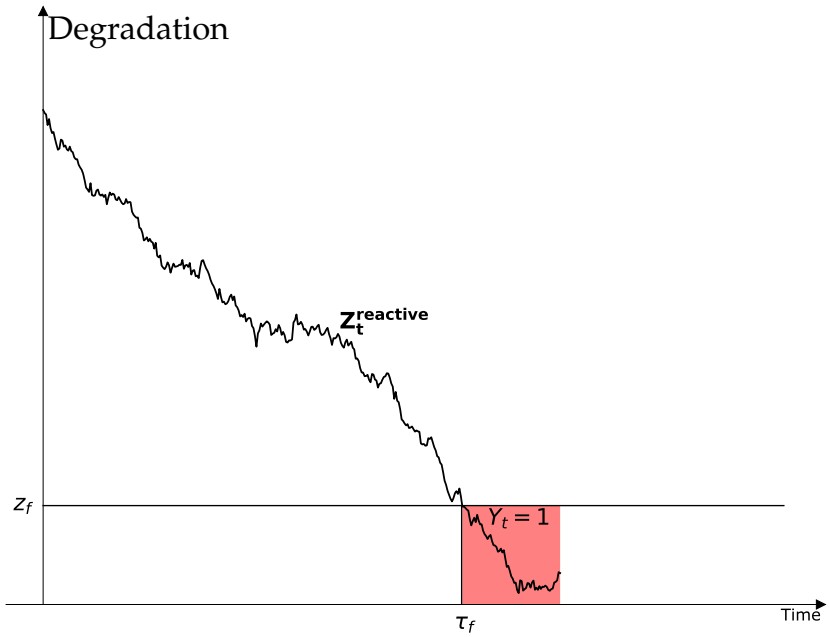
Maintenance is performed when equipment has failed.



From reactive to predictive maintenance

Reactive Maintenance

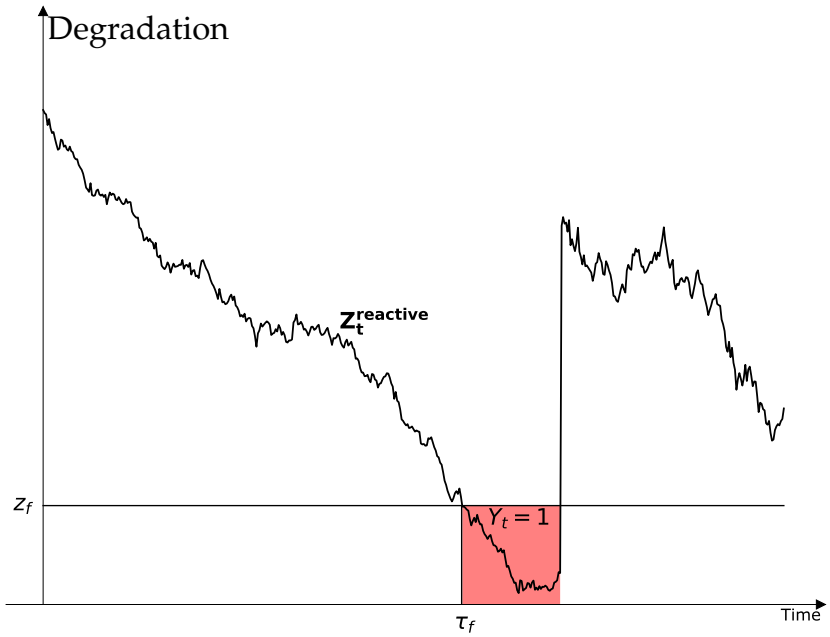
Maintenance is performed when equipment has failed.



From reactive to predictive maintenance

Reactive Maintenance

Maintenance is performed when equipment has failed.



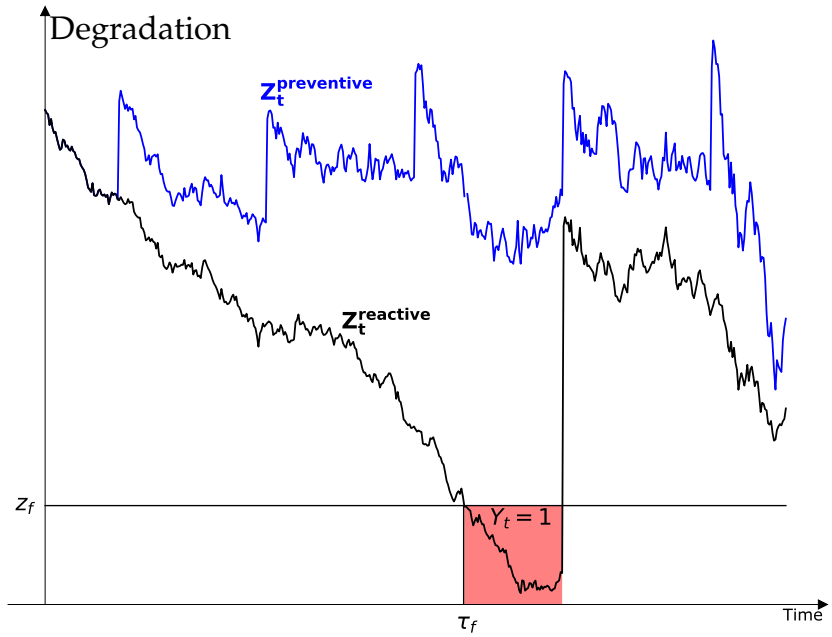
From reactive to predictive maintenance

Reactive Maintenance

Maintenance is performed when equipment has failed.

Preventive Maintenance

Maintenance is performed regularly on equipment to reduce probability of failure



From reactive to predictive maintenance

Reactive Maintenance

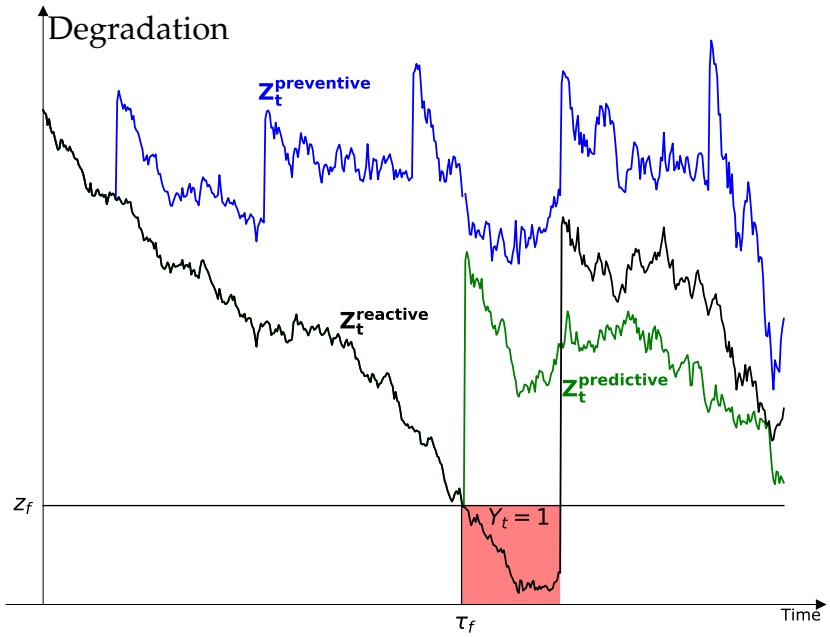
Maintenance is performed when equipment has failed.

Preventive Maintenance

Maintenance is performed regularly on equipment to reduce probability of failure

Predictive Maintenance

Maintenance is performed before equipment failure using predictive insights.



The stochastic optimisation framework

Let $(\Omega, \mathcal{A}, \mathbb{P})$ be the probability space, $X^\lambda : (\Omega, \mathcal{A}, \mathbb{P}) \to \left(E, |\cdot|_E\right)$ a random variable parameterized by $\lambda \in \mathbb{R}^K$.
For $X^\lambda \in L^2_{\mathbb{R}^d}(\Omega, \mathcal{A}, \mathbb{P})$, we investigate the general Stochastic Optimization problem Find $\boldsymbol{\lambda}^*$ such that

$$\begin{aligned} f(\boldsymbol{\lambda}) &= \mathbb{E}\left[F(X, \boldsymbol{\lambda})\right] \\ &= \int_E F(x, \boldsymbol{\lambda})\mu(dx), \end{aligned}$$

is minimized.

The stochastic optimisation framework

steps; simulation (yellow) and optimization (green). The first step produces The simulation phase produces a simulation of the stochastic system or interaction withthe environment, as well as unbiased estimators of the gradient (adapted from Mohamed et al., 2020).

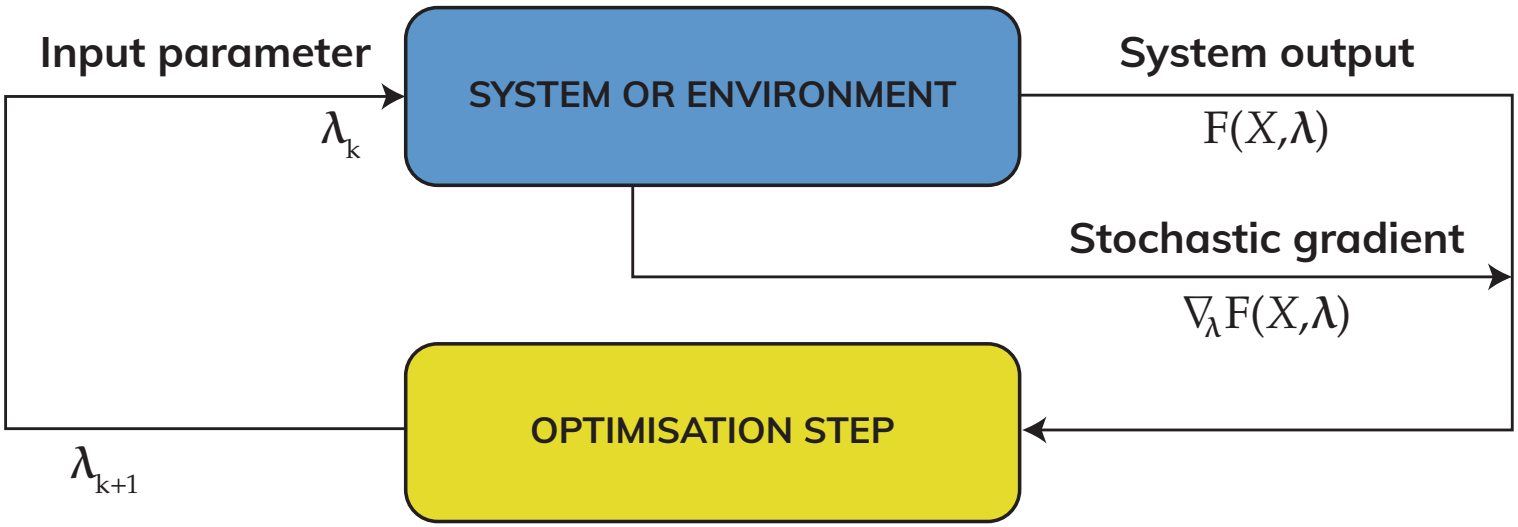


Figure: A typical stochastic optimization process composed of two

Quantized variational inference: Bayes model definition

- Take data y and latent variables z,
- Choose a model $p(y, z)$ represents our view of the studied phenomenon through the choice of $p(y|z)$ and $p(z)$.

The goal of the Bayesian statistician is to find the best latent variable that fits the data, hence the likelihood $p(z|y)$. These quantities are linked by the bayes formula which gives that

$$p(z|y) = \frac{p(z)p(y|z)}{p(y)}, \tag{13}$$

where $p(y)$ is the marginal distribution or normalizing factor, which is a constant.

Background: the voronoi diagram

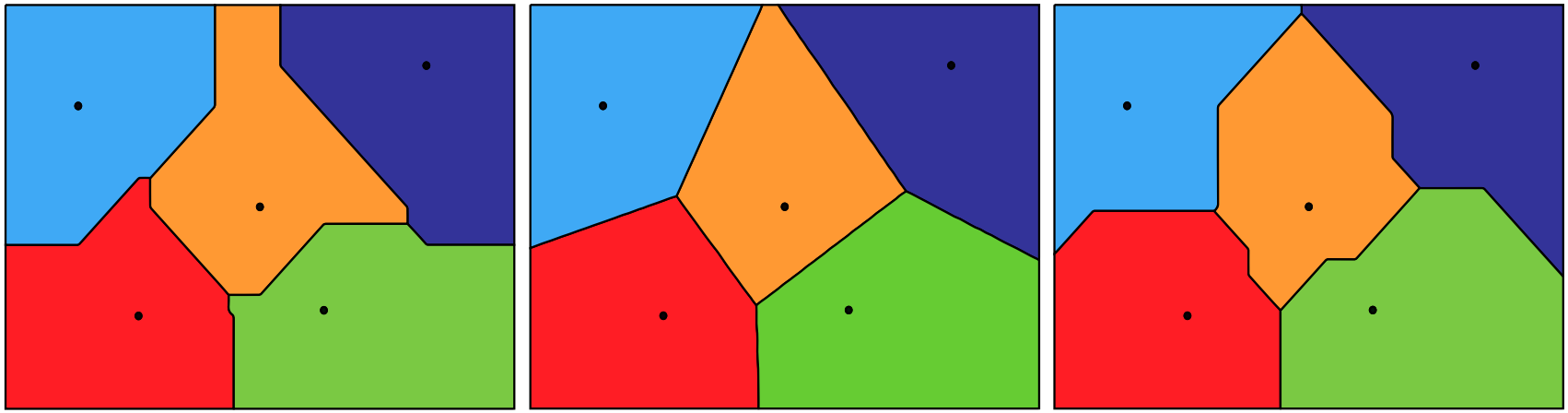


Figure: Voronoi diagram for a finite subset $\Gamma \subset R^2$ with size $n = 5$ for the $\ell_1\left(\mathbb{R}^2\right)$ norm (Manhattan distance, left), $\ell_2\left(\mathbb{R}^2\right)$ norm (Euclidean distance, center) and $\ell_\infty\left(\mathbb{R}^2\right)$ norm (Chebyshev distance, right). Each point x of $\mathbb{R}^2$ is colored by it's associated Voronoi cell. Notably, the Voronoi cells are star-shaped for all considered distances (see Proposition **??**), are convex polytopes in the euclidian case and the separating sets are hyperplanes of $\mathbb{R}^2$.

The optimal transport approach

Definition

Let $(E, \|.\|)$ be a vector space equiped with the norm $\|.\|$, $\mu \in \mathcal{M}(E)$ a probability measure with p-th finite moment and $n \in \mathbb{N}$ the quantization level. Denoting $\mathcal{M}(n)$ the space of probability measure with support at most n, the optimal quantizer $\hat{\nu}_n$ of μ is defined by

$$\hat{\nu}_n = \underset{\nu \in \mathcal{M}(K)}{\operatorname{argmin}} \mathcal{W}_p(\mu, \nu). \tag{14}$$

Main result: a localized bound for the pattern mining problem

Definition

Let $\mathcal{F}$ be the functional family and $\hat{\mathcal{F}}_r$ the empirical star localized class (??). For Rademacher trial count m, sample size n, and any $\delta \in [0,1]$, define the following

$$\hat{\psi}_{n,m}(r) \doteq 2\hat{\mathcal{R}}_n\left(\hat{\mathcal{F}}_r, \boldsymbol{x}, \boldsymbol{\sigma}\right) + 2\hat{r}\breve{\phi}\left(\frac{2\ln\frac{4}{\delta}}{nm\hat{r}}\right) + r\hat{\phi}\left(\frac{2\ln\frac{4}{\delta}}{nr}\right) \quad , \tag{15}$$

with $\hat{r} \doteq 3r + 5r\hat{\phi}\left(\frac{\ln\frac{4}{\delta}}{5nr}\right)$ and consider the *fixed point* $\hat{r}_n^*$ such that $\hat{r}_n^* = \hat{\psi}_{n,m}(\hat{r}_n^*)$. For all $K > 0$, we set $r^{\mathrm{U}}(K)$ to be the fixed point w.r.t. r of the following equation

$$\sqrt{r\hat{r}_n^*} + \left[2\sqrt{r\hat{r}_n^*} + r\right]\breve{\phi}\left(\frac{\frac{1}{n}\ln\frac{4}{\delta}}{2\sqrt{r\hat{r}_n^*} + r}\right) = \frac{r}{K} \quad . \tag{16}$$

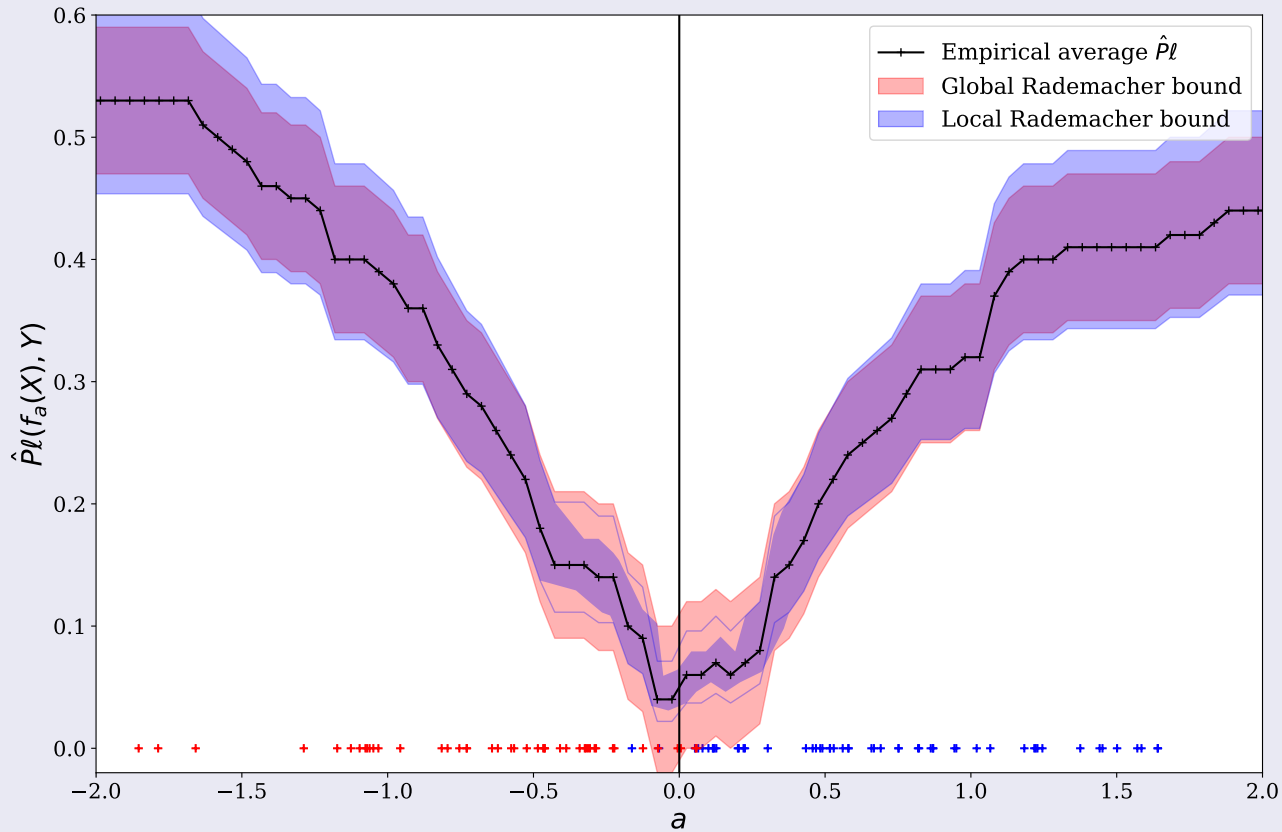
Local rademacher complexity: an intuitive example

- $X \sim \mathcal{N}(0, 1)$
- $Y \sim \text{sign}(\alpha + X + \epsilon)$
- $\ell(x, y) = |y - x|$
- $\mathcal{F} = \{\text{sign}(x + a); a \in \mathbb{R}\}$

Local rademacher complexity: an intuitive example

- $X \sim \mathcal{N}(0, 1)$
- $Y \sim \text{sign}(\alpha + X + \epsilon)$
- $\ell(x, y) = |y - x|$
- $\mathcal{F} = \{\text{sign}(x + a); a \in \mathbb{R}\}$

Image



The discriminative mining problem

- Let $E = (e_1, \ldots, e_d)$ the base dictionary of events and $\mathcal{E} = \mathcal{P}(E)$ the collection of all 2^d possible patterns on E.
- A database of pattern from a random process valued in $\mathcal{E}$ is composed of ordered set of event from E and an associated label, such that $\mathcal{D} = \{(x_i, l_i)_{i=1}^n\}$ of elements of $\mathcal{E} \times \{0, 1\}$

Sequence	Label	Events
T_1	1	$\{e_1, e_2\}$
T_2	0	$\{e_1, e_2, e_4\}$
T_3	1	$\{e_1, e_2, e_3, e_4\}$
T_4	0	$\{e_1, e_3\}$
T_5	0	$\{e_2, e_3, e_4\}$

- Question: For any pattern in $x \in \mathcal{P}(E)$, what is the statistical difference of frequency in each class ?

The discriminative pattern mining problem

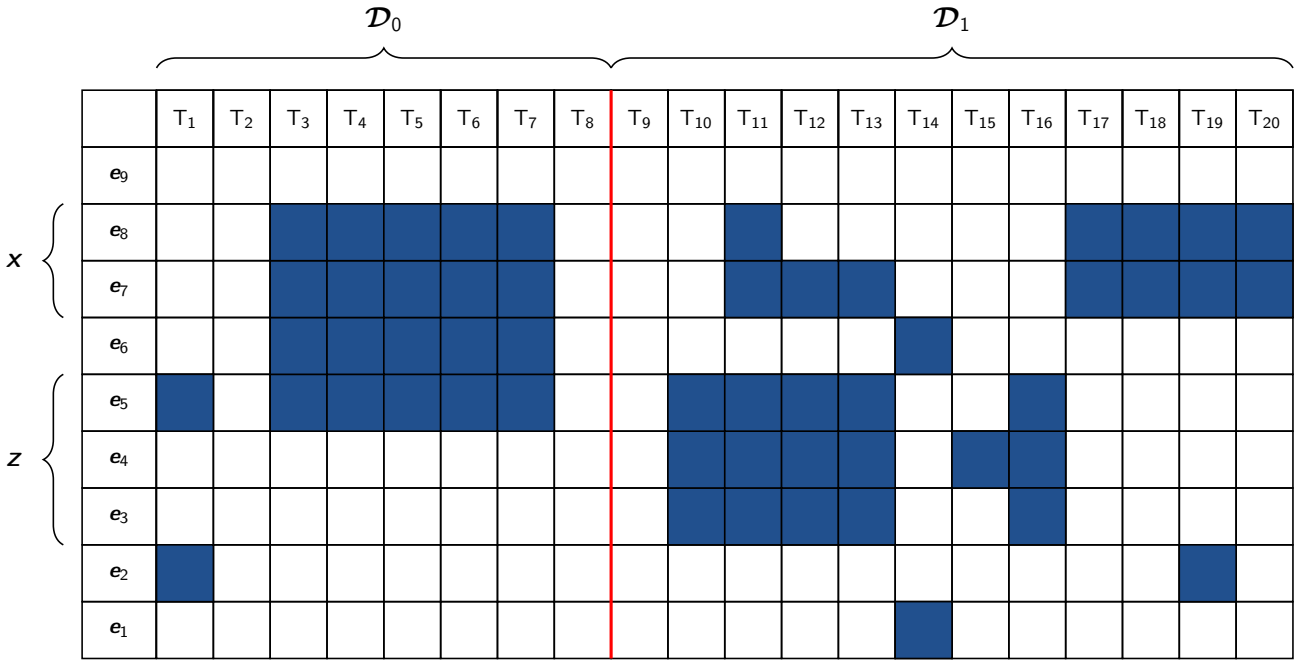


Figure: An example data set of events $\mathcal{D} = \mathcal{D}_0 \cup \mathcal{D}_1$. Row corresponds to items in $E = (e_1, \ldots, e_9)$ and columns to $n = 20$ samples. A blue colored area indicates that the item is present in the sample column considered. In this data set, the pattern $x = \{e_7, e_8\}$ in $\mathcal{E}$ seems to be nondiscriminative since $s_0(x) = s_1(x)$. On the contrary, the pattern $z = \{e_3, e_4, e_5\}$ appears to be specific to the positive class $l = 1$.

The model

Let $\mathrm{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$ be an i.i.d.sample and suppose the underlying model is a bmm with K components. For $k \in \{1, \ldots, K\}$, the k-ith sampling distribution $p_k(\boldsymbol{x_i}|\boldsymbol{\theta_k})$ depends has parameter $\boldsymbol{\theta_k} = (\theta_{kj})_{j=1}^{d}$. Denoting λ_k the probability of sampling from the k-th component with $\sum_{k=1}^{K} \lambda_k = 1$, the global sampling distribution writes

$$p_(\boldsymbol{x_i}|\Theta, \boldsymbol{\lambda}) = \sum_{h=1}^{K} \lambda_k p_k(\boldsymbol{x_i}|\boldsymbol{\theta_k}),$$

where $\Theta = (\boldsymbol{\theta_k})_{k=1}^{K}$ and $\boldsymbol{\lambda} = (\lambda_k)_{k=1}^{K})$.

The model

Knowing the mixture component parameter $\boldsymbol{\lambda}$, the component indicator $\boldsymbol{w_i} = (w_{i1}, \ldots, w_{iK})$ for the sample i is thus distributed as Multin($\boldsymbol{\lambda}$). Finally, the joint distribution is derived as

$$\begin{aligned} p(\mathrm{X}, \mathrm{W}|\Theta, \boldsymbol{\lambda}) &= p(\mathrm{W}|\boldsymbol{\lambda}) p(\mathrm{X}|\mathrm{W}, \Theta) \\ &= \sum_{k=1}^{K} \lambda_k \prod_{i=1}^{n} p_k(\boldsymbol{x_i}|\boldsymbol{\theta_k})^{w_{ik}}. \end{aligned}$$

$$\begin{aligned} \boldsymbol{\lambda}|\boldsymbol{\alpha} &\sim \text{Dirichlet}\,(\boldsymbol{\alpha})\,, \\ \boldsymbol{w_i}|\boldsymbol{\lambda} &\sim \text{Multin}(\boldsymbol{\lambda}), \\ \theta_{kj}|\boldsymbol{\beta}, \boldsymbol{\gamma} &\sim \text{Beta}(\boldsymbol{\beta}, \boldsymbol{\gamma}), \\ x_{ij}|\theta_{kj} &\sim \text{Bernoulli}(\theta_{kj}). \end{aligned}$$

The BFP algorithm

BFP algorithm consists mainly of three steps:

- Given $\mathcal{D} = \mathcal{D}_0 \cup \mathcal{D}_1$, fit the bernoulli mixture model on each subset to find the set of optimal parameter $\Gamma_i = (\Theta_i, \boldsymbol{\lambda_i}, K)$ associated with label i.
- For a pattern $x \in \mathcal{E}$ compute the ratio

$$
\begin{aligned}
r(x) &= \frac{p(\mathcal{M}_1 \mid x)}{p(\mathcal{M}_0 \mid x)} \\
&= \frac{p(\mathcal{M}_1)}{p(\mathcal{M}_0)} \times \frac{p(x \mid \Gamma_1)}{p(x \mid \Gamma_0)}.
\end{aligned}
$$

- The best discriminative pattern are then appended as a variable in the feature space on which any classifier can be trained.

Experiments

Table: Test Accuracy, Recall and AUC 10× cross-validated for bpfd, pf and bc classifiers (with grid-search hyperparameter tuning) for benchmark datasets.

	X Gradient Boosting			Random Forest			Light Gradient-Boosting Machine			Categorical Boosting			Linear Regression			k-Nearest Neighbors		
	BC	PF	bpfd	BC	PF	bpfd	BC	PF	bpfd	BC	PF	bpfd	BC	PF	bpfd	BC	PF	bpfd
ijcnn1																		
AUC	0.728	0.769	**0.927**	0.726	0.767	**0.913**	0.732	0.769	**0.926**	0.727	0.768	**0.927**	0.714	0.732	**0.899**	0.614	0.643	**0.841**
Accuracy	0.906	0.907	**0.929**	0.906	0.907	**0.928**	0.906	0.907	**0.929**	0.906	0.907	**0.93**	0.905	0.905	**0.918**	0.89	0.897	**0.922**
Recall	0.0398	0.0465	**0.403**	0.0411	0.0479	**0.416**	0.0238	0.0372	**0.401**	0.0413	0.0474	**0.407**	0	0.0002	**0.245**	0.106	0.105	**0.419**
F1	0.0742	0.0862	**0.519**	0.0762	0.0885	**0.523**	0.0455	0.0702	**0.516**	0.0765	0.0877	**0.523**	0	0.0003	**0.362**	0.154	0.16	**0.505**
cod-rna																		
AUC	0.776	0.496	**0.815**	0.776	0.496	**0.815**	0.776	0.496	**0.815**	0.776	0.496	**0.815**	0.765	0.495	**0.813**	0.706	0.5	**0.764**
Accuracy	0.718	0.667	**0.775**	0.718	0.667	**0.775**	0.717	0.667	**0.775**	0.718	0.667	**0.775**	0.713	0.667	**0.774**	0.688	0.591	**0.739**
Recall	**0.588**	0	0.383	**0.585**	0	0.386	**0.592**	0	0.384	**0.588**	0	0.384	**0.512**	0	0.364	0.483	0.231	**0.516**
F1	**0.581**	0	0.532	**0.58**	0	0.534	**0.583**	0	0.532	**0.581**	0	0.532	**0.544**	0	0.518	0.503	0.263	**0.568**
a9a																		
AUC	0.89	**0.896**	0.88	0.863	0.869	**0.875**	0.894	0.9	**0.903**	0.894	0.9	**0.904**	0.893	0.902	**0.902**	0.837	0.848	**0.85**
Accuracy	0.841	0.844	**0.846**	0.825	0.826	**0.829**	0.844	0.846	**0.849**	0.844	0.847	**0.848**	0.841	**0.849**	0.847	0.817	**0.826**	0.824
Recall	0.597	0.604	**0.615**	0.564	**0.582**	0.578	0.606	0.613	**0.626**	0.595	0.606	**0.611**	0.581	**0.611**	0.604	0.566	0.584	**0.589**
F1	0.643	0.649	**0.658**	0.607	0.616	**0.619**	0.651	0.656	**0.666**	0.646	0.654	**0.66**	0.637	**0.659**	0.655	0.597	0.616	**0.617**
Doors																		
AUC	0.707	0.691	**0.736**	0.713	0.707	**0.753**	0.706	0.697	**0.739**	0.722	0.715	**0.749**	0.635	0.629	**0.637**	0.557	**0.574**	0.574
Accuracy	0.643	0.629	**0.679**	0.655	0.645	**0.686**	0.647	0.637	**0.681**	0.663	0.657	**0.684**	**0.6**	0.592	0.597	0.546	**0.551**	0.551
Recall	0.614	0.608	**0.642**	0.594	0.585	**0.608**	0.595	0.577	**0.619**	0.569	0.56	**0.592**	0.652	**0.674**	0.648	**0.545**	0.526	0.526
F1	0.632	0.62	**0.667**	0.632	0.622	**0.659**	0.627	0.613	**0.66**	0.627	0.619	**0.652**	0.62	**0.623**	0.617	**0.545**	0.539	0.539


Rakesh Agrawal, Tomasz Imielinski, and Arun Swami. "Mining Association Rules between Sets of Items in Large Databases". In: *In: Proceedings of the 1993 Acm Sigmod International Conference on Management of Data, Washington Dc (Usa*. 1993, pp. 207–216.


Rakesh Agrawal and Ramakrishnan Srikant. "Fast Algorithms for Mining Association Rules". In: *Proc. 20th Int. Conf. Very Large Data Bases, VLDB*. Vol. 1215. Citeseer, 1994, pp. 487–499.


Peter L. Bartlett, Olivier Bousquet, and Shahar Mendelson. "Local Rademacher Complexities". In: *The Annals of Statistics* 33.4 (Aug. 2005), pp. 1497–1537.


Léon Bottou, Frank E. Curtis, and Jorge Nocedal. "Optimization Methods for Large-Scale Machine Learning". In: *SIAM Review* 60.2 (Jan. 2018), pp. 223–311.


Alexander Buchholz, Florian Wenzel, and Stephan Mandt. "Quasi-Monte Carlo Variational Inference". In: *International Conference on Machine Learning*. July 2018. Chap. Machine Learning, pp. 668–677.

Cyrus Cousins* and Amir Dib*. "Fast Convergence Rates for Low-Frequency Pattern Mining with Localization". In: *To Be Submitted.* 2021.

Amir Dib. "Quantized Variational Inference". In: *Advances in Neural Information Processing Systems* 33 (2020).

Amir Dib et al. "Bayesian Feature Discovery for Predictive Maintenance". In: *2021 29th European Signal Processing Conference (EUSIPCO)*. IEEE, Mar. 2021.

Marie Garin et al. "Epidemic Models for COVID-19 during the First Wave from February to May 2020: A Methodological Review". In: *arXiv:2109.01450 [q-bio, stat]* (Sept. 2021).

Tomas Geffner and Justin Domke. "Using Large Ensembles of Control Variates for Variational Inference". In: *Advances in Neural Information Processing Systems 31*. Ed. by S. Bengio et al. Curran Associates, Inc., 2018, pp. 9960–9970.

Sushma Kamlu and V Laxmi. "Condition-Based Maintenance Strategy for Vehicles Using Hidden Markov Models". In: *Advances in Mechanical Engineering* 11.1 (Jan. 2019), p. 168781401880638.

- Durk P Kingma, Tim Salimans, and Max Welling. "Variational Dropout and the Local Reparameterization Trick". In: *Advances in Neural Information Processing Systems 28*. Ed. by C. Cortes et al. Curran Associates, Inc., 2015, pp. 2575–2583.
- Andrew Miller et al. "Reducing Reparameterization Gradient Variance". In: *Advances in Neural Information Processing Systems 30*. Ed. by I. Guyon et al. Curran Associates, Inc., 2017, pp. 3708–3718.
- Shakir Mohamed et al. "Monte Carlo Gradient Estimation in Machine Learning.". In: *J. Mach. Learn. Res.* 21.132 (2020), pp. 1–62.
- Krikamol Muandet et al. "Kernel Mean Embedding of Distributions: A Review and Beyond". In: *Foundations and Trends® in Machine Learning* 10.1-2 (2017), pp. 1–141.
- Gilles Pagès. "Introduction to Vector Quantization and Its Applications for Numerics". In: *ESAIM: Proceedings and Surveys* 48 (Jan. 2015), pp. 29–79.
- Gilles Pagès. *Numerical Probability: An Introduction with Applications to Finance*. Universitext. Springer International Publishing, 2018.

Leonardo Pellegrina et al. "MCRapper: Monte-Carlo Rademacher Averages for Poset Families and Approximate Pattern Mining". In: *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2020, pp. 2165–2174.

Matteo Riondato and Eli Upfal. "Mining Frequent Itemsets through Progressive Sampling with Rademacher Averages". In: *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '15*. Sydney, NSW, Australia: ACM Press, 2015, pp. 1005–1014.

Matteo Riondato and Eli Upfal. "VC-Dimension and Rademacher Averages: From Statistical Learning Theory to Sampling Algorithms". In: *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: Association for Computing Machinery, Aug. 2015, pp. 2321–2322.

Geoffrey Roeder, Yuhuai Wu, and David K Duvenaud. "Sticking the Landing: Simple, Lower-Variance Gradient Estimators for Variational Inference". In: *Advances in Neural Information Processing Systems*

30. Ed. by I. Guyon et al. Curran Associates, Inc., 2017, pp. 6925–6934.

Francisco J. R. Ruiz, Michalis K. Titsias, and David M. Blei. "Overdispersed Black-Box Variational Inference". In: *Proceedings of the Thirty-Second Conference on Uncertainty in Artificial Intelligence*. UAI'16. Jersey City, New Jersey, USA: AUAI Press, June 2016, pp. 647–656.

L. K. Saul, T. Jaakkola, and M. I. Jordan. "Mean Field Theory for Sigmoid Belief Networks". In: *Journal of Artificial Intelligence Research* 4 (Mar. 1996), pp. 61–76.

Minh-Ngoc Tran, David J. Nott, and Robert Kohn. "Variational Bayes With Intractable Likelihood". In: *Journal of Computational and Graphical Statistics* 26.4 (Oct. 2017), pp. 873–882.