# Quantization, learning and games

Sébastien Loustau

Université d'Angers, Larema

C.I.R.M., February 2016

Joint work with B. Guedj, L. Li and W. Farhani

# Contents

# Contents

# Online learning

- $(z_t)_{t=1}^T$, $z_t \in \mathcal{Z}$ a **deterministic** sequence,

# Online learning

- $(z_t)_{t=1}^T$, $z_t \in \mathcal{Z}$ a **deterministic** sequence,
- $\{\mathcal{I}_{\theta,t}, \theta \in \Theta\}_{t=1}^T$ available information.

# Online learning

- $(z_t)_{t=1}^T$, $z_t \in \mathcal{Z}$ a **deterministic** sequence,
- $\{\mathcal{I}_{\theta,t}, \theta \in \Theta\}_{t=1}^T$ available information.

## Game protocol

$\forall t = 1, \ldots, T$ :

# Online learning

- $(z_t)_{t=1}^T$, $z_t \in \mathcal{Z}$ a **deterministic** sequence,
- $\{\mathcal{I}_{\theta,t}, \theta \in \Theta\}_{t=1}^T$ available information.

## Game protocol

$\forall t = 1, \ldots, T$ :

1. Observe $\mathcal{I}_{\theta,t}$, $\theta \in \Theta$

# Online learning

- $(z_t)_{t=1}^T$, $z_t \in \mathcal{Z}$ a **deterministic** sequence,
- $\{\mathcal{I}_{\theta,t}, \theta \in \Theta\}_{t=1}^T$ available information.

## Game protocol

$\forall t = 1, \dots, T$ :

1. Observe $\mathcal{I}_{\theta,t}$, $\theta \in \Theta$ and predict $z_t$ with

$$\hat{z}_t$$

# Online learning

- $(z_t)_{t=1}^T$, $z_t \in \mathcal{Z}$ a **deterministic** sequence,
- $\{\mathcal{I}_{\theta,t}, \theta \in \Theta\}_{t=1}^T$ available information.

## Game protocol

$\forall t = 1, \ldots, T$ :

1. Observe $\mathcal{I}_{\theta,t}$, $\theta \in \Theta$ and predict $z_t$ with

$$\hat{z}_t := \hat{z}_t\left((z_s)_{s=1}^{t-1}, \{\mathcal{I}_{\theta,s}, \theta \in \Theta\}_{s=1}^{t}\right).$$

# Online learning

- $(z_t)_{t=1}^T$, $z_t \in \mathcal{Z}$ a **deterministic** sequence,
- $\{\mathcal{I}_{\theta,t}, \theta \in \Theta\}_{t=1}^T$ available information.

## Game protocol

$\forall t = 1, \ldots, T$ :

1. Observe $\mathcal{I}_{\theta,t}$, $\theta \in \Theta$ and predict $z_t$ with

$$\hat{z}_t := \hat{z}_t \left( (z_s)_{s=1}^{t-1}, \{\mathcal{I}_{\theta,s}, \theta \in \Theta\}_{s=1}^{t} \right).$$

2. Observe $z_t$ and pay
   - $\ell(\hat{z}_t, z_t)$ for your algorithm,
   - $\ell(\mathcal{I}_{\theta,t}, z_t)$ for each piece of information $\theta$.

# Example : prediction with (finite) expert advices

- $(z_t)_{t=1}^{T}$, $z_t \in \mathbb{R}$ a **deterministic** sequence,

# Example : prediction with (finite) expert advices

- $(z_t)_{t=1}^T$, $z_t \in \mathbb{R}$ a **deterministic** sequence,
- $\{p_{k,t}, k = 1, \ldots, N\}_{t=1}^T$ expert advices.

# Example : prediction with (finite) expert advices

- $(z_t)_{t=1}^T$, $z_t \in \mathbb{R}$ a **deterministic** sequence,
- $\{p_{k,t}, k = 1, \ldots, N\}_{t=1}^T$ expert advices.

## Game protocol

$\forall t = 1, \ldots, T$ :

1. Observe $p_{k,t}$, $k = 1, \ldots, N$ and predict

$$\hat{z}_t := \hat{z}_t \left( \{\ell(p_{k,s}, z_s), k = 1, \ldots, N\}_{s=1}^{t-1}, \{p_{k,t}, k = 1, \ldots, N\} \right) .$$

# Example : prediction with (finite) expert advices

- $(z_t)_{t=1}^T$, $z_t \in \mathbb{R}$ a **deterministic** sequence,
- $\{p_{k,t}, k = 1, \ldots, N\}_{t=1}^T$ expert advices.

## Game protocol

$\forall t = 1, \ldots, T$ :

1. Observe $p_{k,t}$, $k = 1, \ldots, N$ and predict

$$\hat{z}_t := \hat{z}_t \left( \{\ell(p_{k,s}, z_s), k = 1, \ldots, N\}_{s=1}^{t-1}, \{p_{k,t}, k = 1, \ldots, N\} \right).$$

2. Observe $z_t$ and pay
   - $\ell(\hat{z}_t, z_t)$ for your algorithm,
   - $\ell(p_{k,t}, z_t)$ for expert number $k$.

# First results

- ($\{0, 1\}$ case)

# First results

- ($\{0, 1\}$ case) If $\exists k^\star : p_{k^\star, t} = z_t \forall t$

# First results

- ($\{0,1\}$ case) If $\exists k^\star : p_{k^\star,t} = z_t \forall t$ then $\exists \hat{z} := (\hat{z}_t)_{t=1}^T$ :

$$\mathrm{err}(\hat{z}) \leq C \log N.$$

# First results

- ($\{0,1\}$ case) If $\exists k^\star : p_{k^\star,t} = z_t \forall t$ then $\exists \hat{z} := (\hat{z}_t)_{t=1}^T$ :
$$\mathrm{err}(\hat{z}) \leq C \log N.$$
- Proof using a weighted majority vote algorithm

# First results

- ($\{0,1\}$ case) If $\exists k^\star : p_{k^\star,t} = z_t \forall t$ then $\exists \hat{z} := (\hat{z}_t)_{t=1}^T$ :

$$\mathrm{err}(\hat{z}) \leq C \log N.$$

- Proof using a weighted majority vote algorithm and the inequality $W_m \leq W_{m-1}/2$ where $W_m$ sum of weights after $m$ errors.

# First results

- ($\{0,1\}$ case) If $\exists k^\star : p_{k^\star,t} = z_t \forall t$ then $\exists \hat{z} := (\hat{z}_t)_{t=1}^T$ :

$$\mathrm{err}(\hat{z}) \leq C \log N.$$

- Proof using a weighted majority vote algorithm and the inequality $W_m \leq W_{m-1}/2$ where $W_m$ sum of weights after $m$ errors.
- More generally, we want **regret bounds**:

$$\sum_{t=1}^{T} \ell(\hat{z}_t, z_t) - \min_{k=1,\dots N} \sum_{t=1}^{T} \ell(p_{k,t}, z_t) \leq \Delta_N(T),$$

# First results

- ($\{0,1\}$ case) If $\exists k^\star : p_{k^\star,t} = z_t \forall t$ then $\exists \hat{z} := (\hat{z}_t)_{t=1}^T$ :
$$\mathrm{err}(\hat{z}) \leq C \log N.$$

- Proof using a weighted majority vote algorithm and the inequality $W_m \leq W_{m-1}/2$ where $W_m$ sum of weights after $m$ errors.
- More generally, we want **regret bounds**:
$$\sum_{t=1}^T \ell(\hat{z}_t, z_t) - \min_{k=1,\dots N} \sum_{t=1}^T \ell(p_{k,t}, z_t) \leq \Delta_N(T),$$
where $\Delta_N(T) \ll T$ (consistency)

# First results

- ($\{0,1\}$ case) If $\exists k^\star : p_{k^\star,t} = z_t \forall t$ then $\exists \hat{z} := (\hat{z}_t)_{t=1}^T$ :

$$\mathrm{err}(\hat{z}) \leq C \log N.$$

- Proof using a weighted majority vote algorithm and the inequality $W_m \leq W_{m-1}/2$ where $W_m$ sum of weights after $m$ errors.
- More generally, we want **regret bounds**:

$$\sum_{t=1}^{T} \ell(\hat{z}_t, z_t) - \min_{k=1,\dots,N} \sum_{t=1}^{T} \ell(p_{k,t}, z_t) \leq \Delta_N(T),$$

where $\Delta_N(T) \ll T$ (consistency) or $\Delta_N(T) = \mathcal{O}(1)$ (fast rates).

# Exponential Weighted Average Forecaster

### Theorem

If $\ell(\cdot, z)$ is convex and $[0, 1]$-bounded :

$$\sum_{t=1}^{T} \ell(\hat{z}_t, z_t) - \min_{k=1,\dots N} \sum_{t=1}^{T} \ell(p_{k,t}, z_t) \leq \frac{\log N}{\lambda} + \frac{\lambda T}{8},$$

# Exponential Weighted Average Forecaster

### Theorem

If $\ell(\cdot, z)$ is convex and $[0,1]$-bounded :

$$\sum_{t=1}^{T} \ell(\hat{z}_t, z_t) - \min_{k=1,\ldots N} \sum_{t=1}^{T} \ell(p_{k,t}, z_t) \leq \frac{\log N}{\lambda} + \frac{\lambda T}{8},$$

where :

$$\hat{z}_t = \sum_{k=1}^{N} \frac{e^{-\lambda \sum_{u=1}^{t-1} \ell(p_{k,u}, z_u)}}{W_{t-1}} p_{k,t}, \;\; \forall t = 1, \ldots, T.$$

# Exponential Weighted Average Forecaster

- Proof based on Hoeffding's lemma.

# Exponential Weighted Average Forecaster

- Proof based on Hoeffding's lemma.
- For $\lambda = \sqrt{(8 \log N)/T}$:

$$\frac{\log N}{\lambda} + \frac{\lambda T}{8} = \sqrt{(T \log N)/2}.$$

# Exponential Weighted Average Forecaster

- Proof based on Hoeffding's lemma.
- For $\lambda = \sqrt{(8 \log N)/T}$:

$$\frac{\log N}{\lambda} + \frac{\lambda T}{8} = \sqrt{(T \log N)/2}.$$

- If for some $\lambda > 0$:

$$\hat{z} \mapsto e^{-\lambda \ell(\hat{z}, z)}$$

  is concave, the bound becomes $\log N/\lambda$

# Exponential Weighted Average Forecaster

- Proof based on Hoeffding's lemma.
- For $\lambda = \sqrt{(8 \log N)/T}$:

$$\frac{\log N}{\lambda} + \frac{\lambda T}{8} = \sqrt{(T \log N)/2}.$$

- If for some $\lambda > 0$:

$$\hat{z} \mapsto e^{-\lambda \ell(\hat{z}, z)}$$

  is concave, the bound becomes $\log N/\lambda \Rightarrow$ Fast rates phenomenon.

# Exponential Weighted Average Forecaster

- Proof based on Hoeffding's lemma.
- For $\lambda = \sqrt{(8 \log N)/T}$:

$$\frac{\log N}{\lambda} + \frac{\lambda T}{8} = \sqrt{(T \log N)/2}.$$

- If for some $\lambda > 0$:

$$\hat{z} \mapsto e^{-\lambda \ell(\hat{z}, z)}$$

  is concave, the bound becomes $\log N/\lambda \Rightarrow$ Fast rates phenomenon.
  Example : square loss, $\lambda \leq 1/2B^2$, $\max |z_t| \leq B$.

# Online regression setting

- $(y_t)_{t=1}^T$, $y_t \in \mathbb{R}$,

# Online regression setting

- $(y_t)_{t=1}^T$, $y_t \in \mathbb{R}$,
- $\{f_\theta(x_t), \theta \in \Theta \subseteq \mathbb{R}^p\}_{t=1}^T$.

# Online regression setting

- $(y_t)_{t=1}^T$, $y_t \in \mathbb{R}$,
- $\{f_\theta(x_t), \theta \in \Theta \subseteq \mathbb{R}^p\}_{t=1}^T$.

## Game protocol

$\forall t = 1, \ldots, T$ :

# Online regression setting

- $(y_t)_{t=1}^T$, $y_t \in \mathbb{R}$,
- $\{f_\theta(x_t), \theta \in \Theta \subseteq \mathbb{R}^p\}_{t=1}^T$.

## Game protocol

$\forall t = 1, \ldots, T$ :

1. Observe $x_t$ and predict:

$$\hat{y}_t := \hat{y}_t \left( \{(f_\theta(x_s) - y_s)^2, \theta \in \Theta\}_{s=1}^{t-1}, \{f_\theta(x_t), \theta \in \Theta\} \right) .$$

# Online regression setting

- $(y_t)_{t=1}^T$, $y_t \in \mathbb{R}$,
- $\{f_\theta(x_t), \theta \in \Theta \subseteq \mathbb{R}^p\}_{t=1}^T$.

## Game protocol

$\forall t = 1, \ldots, T$ :

1. Observe $x_t$ and predict:

$$\hat{y}_t := \hat{y}_t \left( \{(f_\theta(x_s) - y_s)^2, \theta \in \Theta\}_{s=1}^{t-1}, \{f_\theta(x_t), \theta \in \Theta\} \right).$$

2. Observe $y_t$ and pay
   - $(\hat{y}_t - y_t)^2$ for your algorithm,
   - $(f_\theta(x_t) - y_t)^2$ for expert $\theta$.

# High dimensional setting

- $\Theta = \mathbb{R}^p$, $p \gg T$ and $f_\theta(x_t) = \langle \theta, x_t \rangle$ or $\langle \theta, \Psi(x_t) \rangle$ for some dictionnary $\Psi := \{\psi_1, \ldots, \psi_p\}$.

# High dimensional setting

- $\Theta = \mathbb{R}^p$, $p \gg T$ and $f_\theta(x_t) = \langle \theta, x_t \rangle$ or $\langle \theta, \Psi(x_t) \rangle$ for some dictionnary $\Psi := \{\psi_1, \dots, \psi_p\}$.
- We want a **sparsity regret bound** as:

$$\sum_{t=1}^{T} \ell(\hat{y}_t, y_t) \leq \inf_{\theta \in \mathbb{R}^p} \left\{ \sum_{t=1}^{T} (f_\theta(x_t) - y_t)^2 + \Delta_{p,\theta}(T) \right\},$$

# High dimensional setting

- $\Theta = \mathbb{R}^p$, $p \gg T$ and $f_\theta(x_t) = \langle \theta, x_t \rangle$ or $\langle \theta, \Psi(x_t) \rangle$ for some dictionnary $\Psi := \{\psi_1, \dots, \psi_p\}$.
- We want a **sparsity regret bound** as:

$$\sum_{t=1}^{T} \ell(\hat{y}_t, y_t) \leq \inf_{\theta \in \mathbb{R}^p} \left\{ \sum_{t=1}^{T} (f_\theta(x_t) - y_t)^2 + \Delta_{p,\theta}(T) \right\},$$

where $\Delta_{p,\theta}(T)$ grows linearly in $|\theta|_0$, logarithmically in $p$ and $T$. If $|\theta|_0^\star = s^\star \ll p$, the remainder term becomes $s^\star \log p \log T$.

# Solution : as before !

## Algorithm

**Parameters** prior $\pi \in \mathcal{P}(\Theta)$, temperature $\lambda > 0$.

# Solution : as before !

## Algorithm

**Parameters** prior $\pi \in \mathcal{P}(\Theta)$, temperature $\lambda > 0$.
**Initialization** $\hat{p}_1 := \pi$.

# Solution : as before !

## Algorithm

**Parameters** prior $\pi \in \mathcal{P}(\Theta)$, temperature $\lambda > 0$.
**Initialization** $\hat{p}_1 := \pi$.
**At each round** $t \geq 1$:

1. Observe $x_t$ and predict as:

$$\hat{y}_t := \mathbb{E}_{\hat{p}_t} f_\theta(x_t).$$

2. Observe $y_t$ and compute:

$$\mathrm{d}\hat{p}_{t+1}(\theta) := \frac{e^{-\lambda \sum_{s=1}^{t} (y_s - f_\theta(x_s))^2}}{W_t} \mathrm{d}\pi(\theta).$$

# Solution : as before !

- PAC-Bayesian bound inequality:

$$\sum_{t=1}^{T}(\hat{y}_t - y_t)^2 \leq \inf_{\rho \in \mathcal{P}(\Theta)} \mathbb{E}_{\theta\sim\rho}\left\{\sum_{t=1}^{T}(f_\theta(x_t) - y_t)^2 + \frac{\mathcal{K}(\rho,\pi)}{\lambda}\right\},$$

# Solution : as before !

- PAC-Bayesian bound inequality:

$$\sum_{t=1}^{T}(\hat{y}_t - y_t)^2 \leq \inf_{\rho \in \mathcal{P}(\Theta)} \mathbb{E}_{\theta \sim \rho} \left\{ \sum_{t=1}^{T}(f_\theta(x_t) - y_t)^2 + \frac{\mathcal{K}(\rho, \pi)}{\lambda} \right\},$$

  where $\mathcal{K}(\cdot, \cdot)$ satisfies a duality formula.

# Solution : as before !

- PAC-Bayesian bound inequality:

$$\sum_{t=1}^{T} (\hat{y}_t - y_t)^2 \leq \inf_{\rho \in \mathcal{P}(\Theta)} \mathbb{E}_{\theta \sim \rho} \left\{ \sum_{t=1}^{T} (f_\theta(x_t) - y_t)^2 + \frac{\mathcal{K}(\rho, \pi)}{\lambda} \right\},$$

  where $\mathcal{K}(\cdot, \cdot)$ satisfies a duality formula.
- Next, by choosing a **sparsity prior**:

$$\mathrm{d}\pi_\tau(\theta) := \prod_{j=1}^{p} \left( \frac{\mathrm{const.}}{\tau + \theta_j^2} \right)^2 \mathrm{d}\theta_j,$$

# Solution : as before !

- PAC-Bayesian bound inequality:

$$\sum_{t=1}^{T}(\hat{y}_t - y_t)^2 \leq \inf_{\rho \in \mathcal{P}(\Theta)} \mathbb{E}_{\theta \sim \rho}\left\{\sum_{t=1}^{T}(f_\theta(x_t) - y_t)^2 + \frac{\mathcal{K}(\rho, \pi)}{\lambda}\right\},$$

  where $\mathcal{K}(\cdot, \cdot)$ satisfies a duality formula.

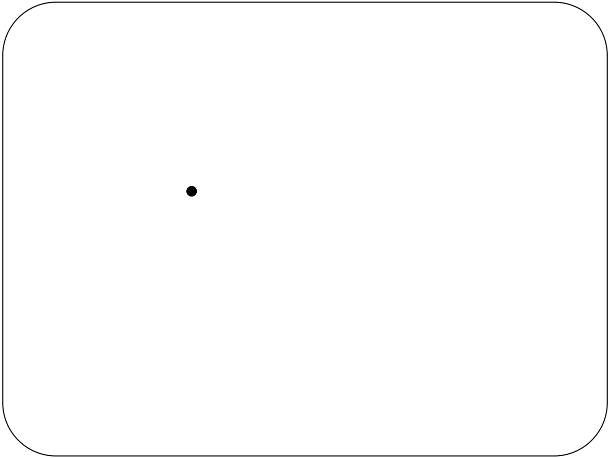
- Next, by choosing a **sparsity prior**:

$$\mathrm{d}\pi_\tau(\theta) := \prod_{j=1}^{p}\left(\frac{\text{const.}}{\tau + \theta_j^2}\right)^2 \mathrm{d}\theta_j,$$

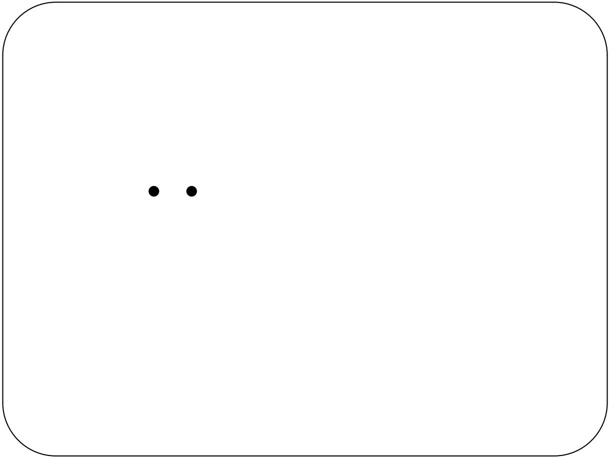
  a sparsity regret bound follows easily.

# The problem of Online Clustering

# The problem of Online Clustering

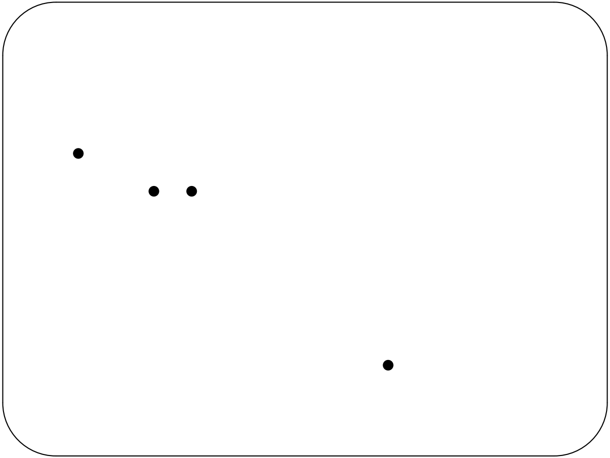


# The problem of Online Clustering

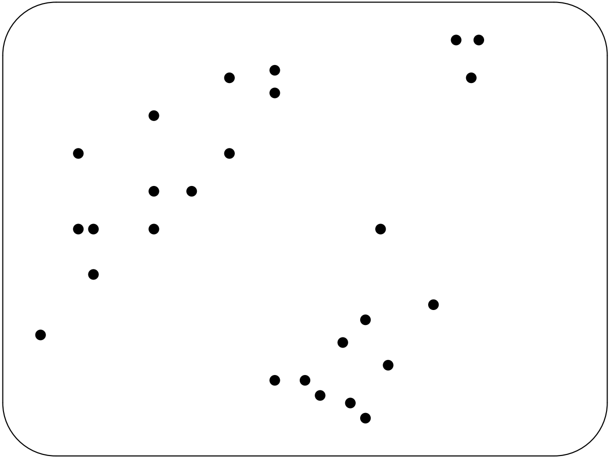


# The problem of Online Clustering

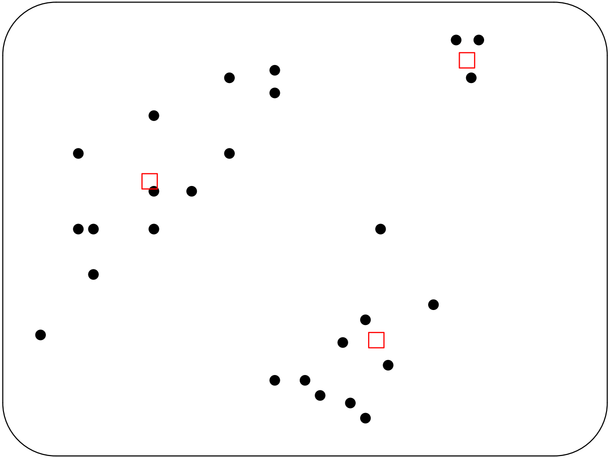
# The problem of Online Clustering



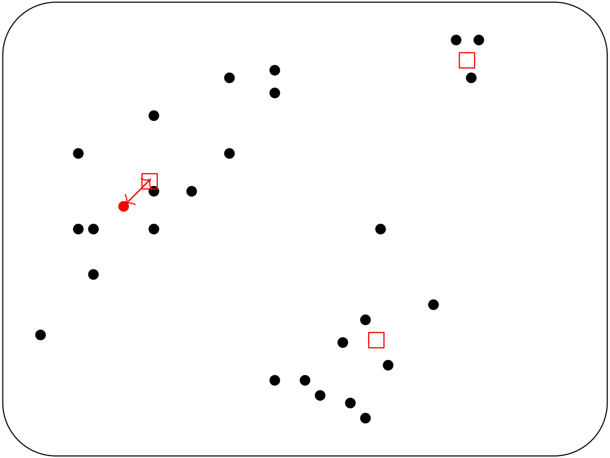
# The problem of Online Clustering



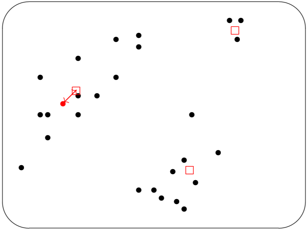
# The problem of Online Clustering



# The problem of Online Clustering

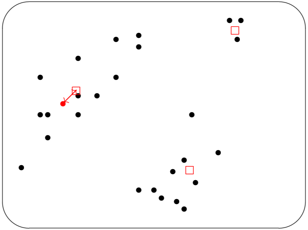


# The problem of Online Clustering



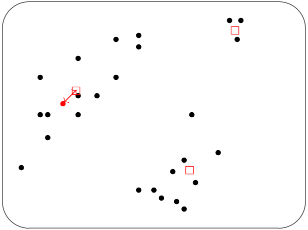
- Principle : prediction in a high dimensional setting.

# The problem of Online Clustering



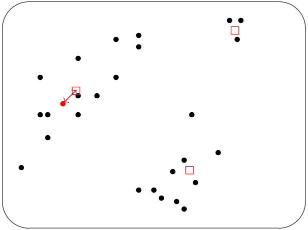
- Principle : prediction in a high dimensional setting.
- Sparsity assumption : points are grouped into $s$ clusters.

# The problem of Online Clustering



- Principle : prediction in a high dimensional setting.
- Sparsity assumption : points are grouped into $s$ clusters.
- Use sparsity priors to choose the number of clusters.

# The problem of Online Clustering



- Principle : prediction in a high dimensional setting.
- Sparsity assumption : points are grouped into $s$ clusters.
- Use sparsity priors to choose the number of clusters.

We want to prove new kind of sparsity regret bounds:

$$\sum_{t=1}^{T} \ell(\hat{\mathbf{c}}_t, x_t) - \inf_{\mathbf{c}\in\mathbb{R}^{dp}} \left\{ \sum_{t=1}^{T} \ell(\mathbf{c}, x_t) + \lambda |\mathbf{c}|_0 \right\},$$

where $|\mathbf{c}|_0 = \mathrm{card}\{j = 1, \dots, p : c_j \neq 0_{\mathbb{R}^d}\}$ and

$$\ell(\mathbf{c}, x) = \min_{j=1,\dots,p} \|c_j - x\|_2^2.$$

# Algorithm : L. 2014

**Parameters :** $p \geq 1$, $\pi \in \mathcal{P}(\mathbb{R}^{dp})$, $\lambda > 0$.

# Algorithm : L. 2014

**Parameters :** $p \geq 1$, $\pi \in \mathcal{P}(\mathbb{R}^{dp})$, $\lambda > 0$.
**Initialization** $\hat{c}_1 \sim \hat{p}_1 := \pi$.

# Algorithm : L. 2014

**Parameters :** $p \geq 1$, $\pi \in \mathcal{P}(\mathbb{R}^{dp})$, $\lambda > 0$.
**Initialization** $\hat{c}_1 \sim \hat{p}_1 := \pi$. Denote $S_0(\cdot) \equiv 0$.

# Algorithm : L. 2014

**Parameters :** $p \geq 1$, $\pi \in \mathcal{P}(\mathbb{R}^{dp})$, $\lambda > 0$.
**Initialization** $\hat{\mathbf{c}}_1 \sim \hat{p}_1 := \pi$. Denote $S_0(\cdot) \equiv 0$.

**At each round** $t = 1, \ldots, T$:

- Observe $x_t$ and compute

$$S_t(\mathbf{c}) = S_{t-1}(\mathbf{c}) + \ell(\mathbf{c}, x_t) + \frac{\lambda}{2}[\ell(\mathbf{c}, x_t) - \ell(\hat{\mathbf{c}}_t, x_t)]^2, \ \forall \mathbf{c} \in \mathbb{R}^{dp}.$$

# Algorithm : L. 2014

**Parameters :** $p \geq 1$, $\pi \in \mathcal{P}(\mathbb{R}^{dp})$, $\lambda > 0$.
**Initialization** $\hat{\mathbf{c}}_1 \sim \hat{p}_1 := \pi$. Denote $S_0(\cdot) \equiv 0$.

**At each round** $t = 1, \ldots, T$:

- Observe $x_t$ and compute

$$S_t(\mathbf{c}) = S_{t-1}(\mathbf{c}) + \ell(\mathbf{c}, x_t) + \frac{\lambda}{2}[\ell(\mathbf{c}, x_t) - \ell(\hat{\mathbf{c}}_t, x_t)]^2, \ \forall \mathbf{c} \in \mathbb{R}^{dp}.$$

- Let $\hat{p}_{t+1}(d\mathbf{c}) := \frac{e^{-\lambda S_t(\mathbf{c})}}{W_t}\pi(d\mathbf{c}) \in \Delta(\mathbb{R}^{dp})$, where $W_t = \mathbb{E}_{\mathbf{c}\sim\pi} e^{-\lambda S_t(\mathbf{c})}$.

# Algorithm : L. 2014

**Parameters :** $p \geq 1$, $\pi \in \mathcal{P}(\mathbb{R}^{dp})$, $\lambda > 0$.
**Initialization** $\hat{\mathbf{c}}_1 \sim \hat{p}_1 := \pi$. Denote $S_0(\cdot) \equiv 0$.

**At each round** $t = 1, \dots, T$:

- Observe $x_t$ and compute

$$S_t(\mathbf{c}) = S_{t-1}(\mathbf{c}) + \ell(\mathbf{c}, x_t) + \frac{\lambda}{2}[\ell(\mathbf{c}, x_t) - \ell(\hat{\mathbf{c}}_t, x_t)]^2, \ \forall \mathbf{c} \in \mathbb{R}^{dp}.$$

- Let $\hat{p}_{t+1}(d\mathbf{c}) := \frac{e^{-\lambda S_t(\mathbf{c})}}{W_t} \pi(d\mathbf{c}) \in \Delta(\mathbb{R}^{dp})$, where $W_t = \mathbb{E}_{\mathbf{c} \sim \pi} e^{-\lambda S_t(\mathbf{c})}$.
- Let $\hat{\mathbf{c}}_{t+1} \sim \hat{p}_{t+1}$.

# Algorithm : L. 2014

**Parameters :** $p \geq 1$, $\pi \in \mathcal{P}(\mathbb{R}^{dp})$, $\lambda > 0$.
**Initialization** $\hat{\mathbf{c}}_1 \sim \hat{p}_1 := \pi$. Denote $S_0(\cdot) \equiv 0$.

**At each round** $t = 1, \dots, T$:

- Observe $x_t$ and compute

$$S_t(\mathbf{c}) = S_{t-1}(\mathbf{c}) + \ell(\mathbf{c}, x_t) + \frac{\lambda}{2}[\ell(\mathbf{c}, x_t) - \ell(\hat{\mathbf{c}}_t, x_t)]^2, \ \forall \mathbf{c} \in \mathbb{R}^{dp}.$$

- Let $\hat{p}_{t+1}(d\mathbf{c}) := \frac{e^{-\lambda S_t(\mathbf{c})}}{W_t}\pi(d\mathbf{c}) \in \Delta(\mathbb{R}^{dp})$, where $W_t = \mathbb{E}_{\mathbf{c}\sim\pi} e^{-\lambda S_t(\mathbf{c})}$.
- Let $\hat{\mathbf{c}}_{t+1} \sim \hat{p}_{t+1}$.

# PAC-Bayesian bound

### Théorème (L. 2014)

$\forall (x_t)_{t=1}^T$, $\forall \pi \in \mathcal{P}(\mathbb{R}^{dp})$, $\forall \lambda > 0$:

$$\sum_{t=1}^{T} \mathbb{E}_{(\hat{p}_1,\dots,\hat{p}_t)} \ell(\hat{\mathbf{c}}_t, x_t) \leq \inf_{\rho \in \mathcal{P}(\mathbb{R}^{dp})} \left\{ \mathbb{E}_{\mathbf{c}\sim\rho} \sum_{t=1}^{T} \ell(\mathbf{c}, x_t) + \frac{\mathcal{K}(\rho,\pi)}{\lambda} + \frac{\lambda}{2} \mathbb{E}_{(\hat{p}_1,\dots,\hat{p}_T)} \mathbb{E}_{\mathbf{c}\sim\rho} \sum_{t=1}^{T} [\ell(\mathbf{c}, x_t) - \ell(\hat{\mathbf{c}}_t, x_t)]^2 \right\}.$$

# PAC-Bayesian bound

### Théorème (L. 2014)

$\forall (x_t)_{t=1}^T$, $\forall \pi \in \mathcal{P}(\mathbb{R}^{dp})$, $\forall \lambda > 0$:

$$\sum_{t=1}^{T} \mathbb{E}_{(\hat{p}_1,\dots,\hat{p}_t)} \ell(\hat{\mathbf{c}}_t, x_t) \leq \inf_{\rho \in \mathcal{P}(\mathbb{R}^{dp})} \Bigg\{ \mathbb{E}_{\mathbf{c}\sim\rho} \sum_{t=1}^{T} \ell(\mathbf{c}, x_t) + \frac{\mathcal{K}(\rho,\pi)}{\lambda} + \frac{\lambda}{2} \mathbb{E}_{(\hat{p}_1,\dots,\hat{p}_T)} \mathbb{E}_{\mathbf{c}\sim\rho} \sum_{t=1}^{T} [\ell(\mathbf{c}, x_t) - \ell(\hat{\mathbf{c}}_t, x_t)]^2 \Bigg\}.$$

### Proof

Online variance inequality (Audibert 2009):

$$\forall \lambda, \forall \rho, \forall x,\ \mathbb{E}_{\mathbf{c}'\sim\rho} \log \mathbb{E}_{\mathbf{c}\sim\rho} e^{\lambda(\ell(\mathbf{c}',x) - \ell(\mathbf{c},x) - \frac{\lambda}{2}[\ell(\mathbf{c},x)-\ell(\mathbf{c}',x)]^2)} \leq 0.$$

# PAC-Bayesian bound

Then, for $t = 1, \ldots, T$:

$$\forall \lambda > 0, \; \mathbb{E}_{\mathbf{c}' \sim \hat{p}_t} \ell(\mathbf{c}', x_t) \leq -\frac{1}{\lambda} \log \mathbb{E}_{\mathbf{c} \sim \hat{p}_t} e^{-\lambda(S_t(c) - S_{t-1}(c))}.$$

# PAC-Bayesian bound

Then, for $t = 1, \ldots, T$:

$$\forall \lambda > 0,\ \mathbb{E}_{\mathbf{c}' \sim \hat{p}_t} \ell(\mathbf{c}', x_t) \leq -\frac{1}{\lambda} \log \mathbb{E}_{\mathbf{c} \sim \hat{p}_t} e^{-\lambda(S_t(c) - S_{t-1}(c))}.$$

Using the chain rule (Barron, 87):

$$\sum_{t=1}^{T} \mathbb{E}_{(\hat{p}_1, \ldots, \hat{p}_t)} \ell(\hat{\mathbf{c}}_t, x_t) \leq -\mathbb{E}_{(\hat{p}_1, \ldots, \hat{p}_T)} \frac{1}{\lambda} \sum_{t=1}^{T} \log \mathbb{E}_{\mathbf{c} \sim \hat{p}_t} e^{-\lambda(S_t(c) - S_{t-1}(c))}$$

$$= -\frac{1}{\lambda} \mathbb{E}_{(\hat{p}_1, \ldots, \hat{p}_T)} \log \prod_{t=1}^{T} \frac{W_t}{W_{t-1}} = -\frac{1}{\lambda} \mathbb{E}_{(\hat{p}_1, \ldots, \hat{p}_T)} \log W_T$$

$$= -\frac{1}{\lambda} \mathbb{E}_{(\hat{p}_1, \ldots, \hat{p}_T)} \log \mathbb{E}_{\mathbf{c} \sim \pi} e^{-\lambda S_T(\mathbf{c})}.$$

# Choice of prior $\pi$

- Choice of $\pi \in \Delta(\mathbb{R}^{dp})$ to get a sparsity regret bound:

$$\pi_\tau(d\mathbf{c}) := C_{R,\tau} \prod_{j=1}^{p} \left\{ \left(1 + \frac{|c_j|_2^2}{\tau^2}\right)^{-\frac{3+d}{2}} \mathbb{1}(|c_j|_2 \leq 2R) \right\} d\mathbf{c}.$$

# Sparsity regret bound

## Théorème (L. 2014)

$\forall (x_t)_{t=1}^T$, *for* $\lambda = \sqrt{(3+d)/T}$:

$$\sum_{t=1}^{T} \mathbb{E}_{(\hat{p}_1,\dots,\hat{p}_t)} \ell(\hat{\boldsymbol{c}}_t, x_t) \leq \inf_{\boldsymbol{c} \in \mathcal{B}(R)} \left\{ \sum_{t=1}^{T} \ell(\boldsymbol{c}, x_t) + |\boldsymbol{c}|_0 \sqrt{T} \log T \right\} + \mathcal{O}(\sqrt{T}).$$

# Sparsity regret bound

## Théorème (L. 2014)

*$\forall (x_t)_{t=1}^T$, for $\lambda = \sqrt{(3+d)/T}$:*

$$\sum_{t=1}^{T} \mathbb{E}_{(\hat{p}_1,\ldots,\hat{p}_t)} \ell(\hat{\mathbf{c}}_t, x_t) \le \inf_{\mathbf{c} \in \mathcal{B}(R)} \left\{ \sum_{t=1}^{T} \ell(\mathbf{c}, x_t) + |\mathbf{c}|_0 \sqrt{T} \log T \right\} + \mathcal{O}(\sqrt{T}).$$

## Proof

Our sparsity induced prior satisfies:

$$\begin{aligned} \mathcal{K}(\pi_\tau, \pi_\tau^{\text{trans}}) &\le (3+d) \sum_{j=1}^{p} \log\left(1 + \frac{|c_j|_2}{\sqrt{6}\tau}\right) + \frac{12 p d \tau^2}{R^2} \\ &\le (3+d)|\mathbf{c}|_0 \log\left(1 + \frac{\sum_{j=1}^{p} |c_j|_2}{\sqrt{6}\tau |\mathbf{c}|_0}\right) + \frac{12 p d \tau^2}{R^2}. \end{aligned}$$

# Extensions

## Online to batch conversion

- $\mathcal{D}_n = \{X_1, \ldots, X_n\}$

# Extensions

## Online to batch conversion

- $\mathcal{D}_n = \{X_1, \ldots, X_n\}$
- $\{\hat{p}_1, \ldots, \hat{p}_{n+1}\}$ as before.

# Extensions

## Online to batch conversion

- $\mathcal{D}_n = \{X_1, \ldots, X_n\}$
- $\{\hat{p}_1, \ldots, \hat{p}_{n+1}\}$ as before. Then:

$$\hat{\mathbf{c}}_{\mathrm{Mirror}} \sim \hat{\mu} := \mathcal{U}(\{\hat{\mathbf{c}}_1, \ldots, \hat{\mathbf{c}}_{n+1} | \mathcal{D}_n\})$$

satisfies a sparsity oracle inequality.

# Extensions

## Online to batch conversion

- $\mathcal{D}_n = \{X_1, \ldots, X_n\}$
- $\{\hat{p}_1, \ldots, \hat{p}_{n+1}\}$ as before. Then:

$$\hat{\mathbf{c}}_{\mathrm{Mirror}} \sim \hat{\mu} := \mathcal{U}(\{\hat{\mathbf{c}}_1, \ldots, \hat{\mathbf{c}}_{n+1} | \mathcal{D}_n\})$$

satisfies a sparsity oracle inequality.

## High dimensional clustering

- $x_t \in \mathbb{R}^p$, $p \gg T$,

# Extensions

## Online to batch conversion

- $\mathcal{D}_n = \{X_1, \ldots, X_n\}$
- $\{\hat{p}_1, \ldots, \hat{p}_{n+1}\}$ as before. Then:

$$\hat{\mathbf{c}}_{\mathrm{Mirror}} \sim \hat{\mu} := \mathcal{U}(\{\hat{\mathbf{c}}_1, \ldots, \hat{\mathbf{c}}_{n+1} | \mathcal{D}_n\})$$

  satisfies a sparsity oracle inequality.

## High dimensional clustering

- $x_t \in \mathbb{R}^p$, $p \gg T$,
- $\ell(\mathbf{c}, x_t) = \min_{j=1,\ldots,k} \|c_j - x_t\|^2$ for a fixed $k$,

# Extensions

## Online to batch conversion

- $\mathcal{D}_n = \{X_1, \ldots, X_n\}$
- $\{\hat{p}_1, \ldots, \hat{p}_{n+1}\}$ as before. Then:

$$\hat{\mathbf{c}}_{\mathrm{Mirror}} \sim \hat{\mu} := \mathcal{U}(\{\hat{\mathbf{c}}_1, \ldots, \hat{\mathbf{c}}_{n+1} | \mathcal{D}_n\})$$

satisfies a sparsity oracle inequality.

## High dimensional clustering

- $x_t \in \mathbb{R}^p$, $p \gg T$,
- $\ell(\mathbf{c}, x_t) = \min_{j=1,\ldots,k} \|c_j - x_t\|^2$ for a fixed $k$,
- we have sparsity (in $p$) regret bounds for:

$$\pi_\tau(d\mathbf{c}) := C_{R,\tau} \prod_{j=1}^{p} \left\{ \left(1 + \frac{|c_j|_2^2}{\tau^2}\right)^{-\frac{3+k}{2}} \mathbb{1}(|c_j|_2 \leq 2R) \right\} d\mathbf{c}.$$

# Extensions

- We want a dynamic choice of clusters:

# Extensions

- We want a dynamic choice of clusters: $\mathcal{C} = \bigcup_{k=1}^{p} \mathbb{R}^{dk}$ instead of $\mathbb{R}^{dp}$.

# Extensions

- We want a dynamic choice of clusters: $\mathcal{C} = \bigcup_{k=1}^{p} \mathbb{R}^{dk}$ instead of $\mathbb{R}^{dp}$.
- We want to sample from the Gibbs $\hat{p}_t$ :

# Extensions

- We want a dynamic choice of clusters: $\mathcal{C} = \bigcup_{k=1}^{p} \mathbb{R}^{dk}$ instead of $\mathbb{R}^{dp}$.
- We want to sample from the Gibbs $\hat{p}_t$ : MCMC approximation

# Extensions

- We want a dynamic choice of clusters: $\mathcal{C} = \bigcup_{k=1}^{p} \mathbb{R}^{dk}$ instead of $\mathbb{R}^{dp}$.
- We want to sample from the Gibbs $\hat{p}_t$ : MCMC approximation
- With different dimensions for the Markov chain : Reverse Jump MCMC

# Contents

# Motivations

- make life to these theorems

# Motivations

- make life to these theorems
- help the project to keep learning and growing

# Motivations

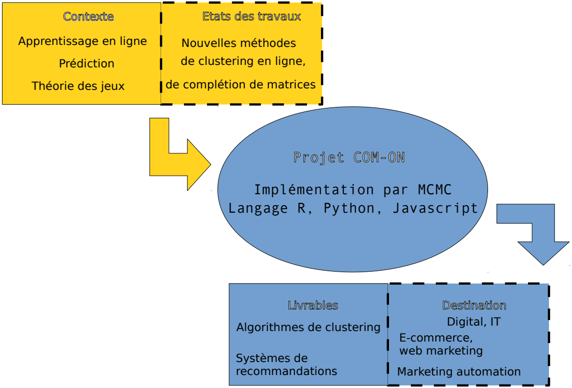
- make life to these theorems
- help the project to keep learning and growing
- impact on the digital revolution

# Motivations

- make life to these theorems
- help the project to keep learning and growing
- impact on the digital revolution

How to do that ?

# A slide to raise fund...



# Some early adopters

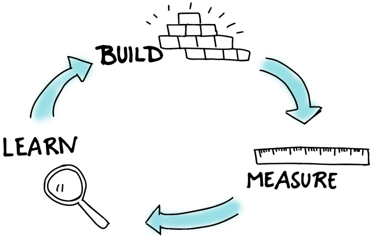
# Some early adopters



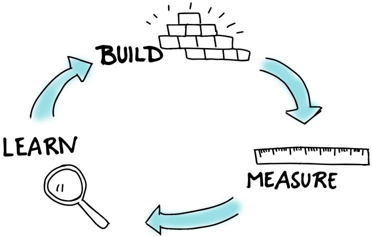
mOment

Better selling with real-time personalization.

# Validated Learning

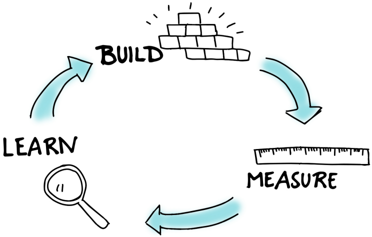


# Validated Learning




- Lean production vs mass production

# Validated Learning




- Lean production vs mass production
- Dropbox, Spotify, Imvu, etc.

# Contents

# Simulations in the batch setting

- $\mathcal{D}_n = \{X_1, \ldots, X_n\}$ from different $k^{\star}$-mixture models,

# Simulations in the batch setting

- $\mathcal{D}_n = \{X_1, \ldots, X_n\}$ from different $k^\star$-mixture models,
- Goal : find the true number of clusters $k^\star$,
- Compare with existing batch techniques.

# Simulations in the batch setting

## 1 group in dimension 5

Observations are sampled from a uniform distribution on the unit hypercube in $\mathbb{R}^5$.

# Simulations in the batch setting

### 1 group in dimension 5

Observations are sampled from a uniform distribution on the unit hypercube in $\mathbb{R}^5$.

### 4 weakly separated Gaussian groups in dimension 2

Observations are sampled from 4 bivariate Gaussian distributions with identity covariance matrix, whose mean vectors are respectively $(0,0), (-2,-1), (0,4), (3,1)$. Each observation is uniformly drawn from one of the four groups.

# Simulations in the batch setting

### 1 group in dimension 5

Observations are sampled from a uniform distribution on the unit hypercube in $\mathbb{R}^5$.

### 4 weakly separated Gaussian groups in dimension 2

Observations are sampled from 4 bivariate Gaussian distributions with identity covariance matrix, whose mean vectors are respectively $(0,0),(-2,-1),(0,4),(3,1)$. Each observation is uniformly drawn from one of the four groups.

### 4 strongly separated Gaussian groups in dimension 2

Same as the previous model but with more separated mean vectors: $(0,0),(-4,-1),(0,7),(5,2)$.

## 7 Gaussian groups in dimension 50

Observations are sampled from 7 multivariate Gaussian distributions in $\mathbb{R}^{50}$ with identity covariance matrix, whose mean vectors are chosen randomly according to a uniform distribution on $[-10, 10]^{50}$. Each observation is uniformly drawn from one of the seven groups.

## 7 Gaussian groups in dimension 50

Observations are sampled from 7 multivariate Gaussian distributions in $\mathbb{R}^{50}$ with identity covariance matrix, whose mean vectors are chosen randomly according to a uniform distribution on $[-10, 10]^{50}$. Each observation is uniformly drawn from one of the seven groups.

## 3 lognormal groups in dimension 3

Observations are sampled from 3 multivariate lognormal distributions in $\mathbb{R}^3$ with identity covariance matrix, whose mean vectors are respectively $(1, 1, 1), (6, 5, 7), (10, 9, 11)$. Each observation is uniformly drawn from one of the three groups.

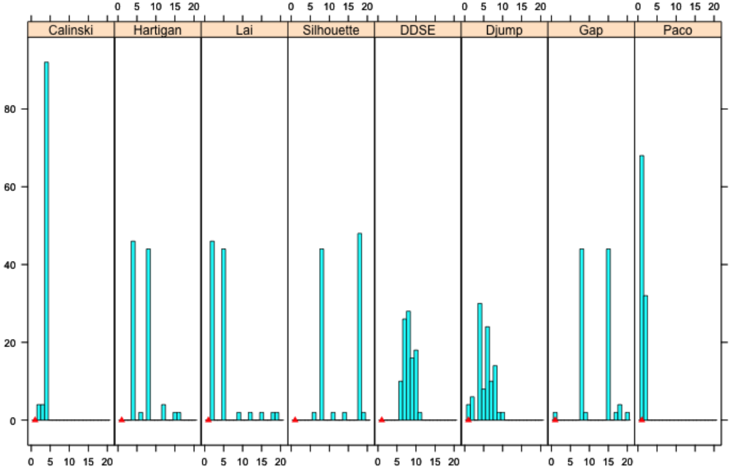
# Clustering methods

- Hartigan (1975),
- Calinski and Harabasz (1974),
- Krzanowski and Lai (1988),
- Kaufman and Rousseeuw (1990),
- Tibshirani (2001),
- DDSE and Djump (Capushe, 2012),

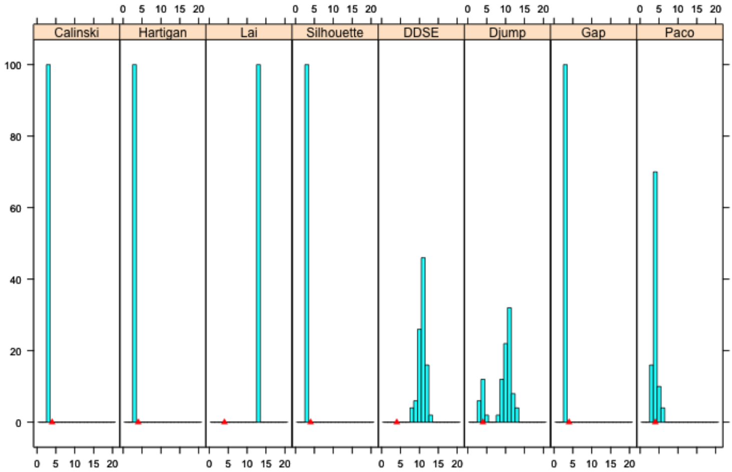
# Clustering methods

- Hartigan (1975),
- Calinski and Harabasz (1974),
- Krzanowski and Lai (1988),
- Kaufman and Rousseeuw (1990),
- Tibshirani (2001),
- DDSE and Djump (Capushe, 2012),
- PAC Online (PACO) with constant parameters (no calibration).

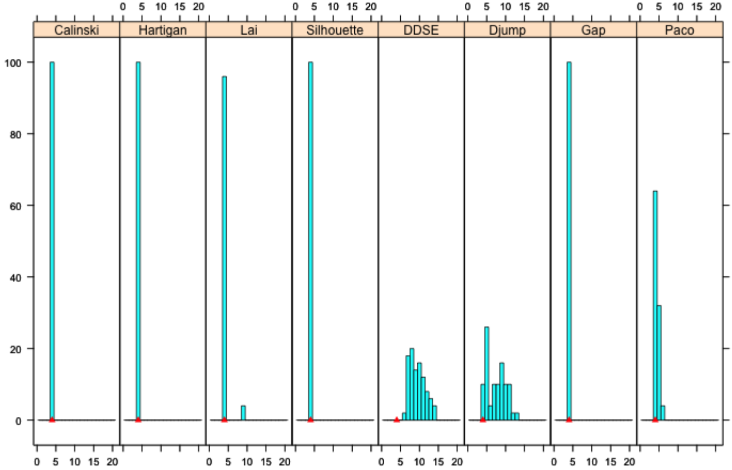
# Histograms (model 1)



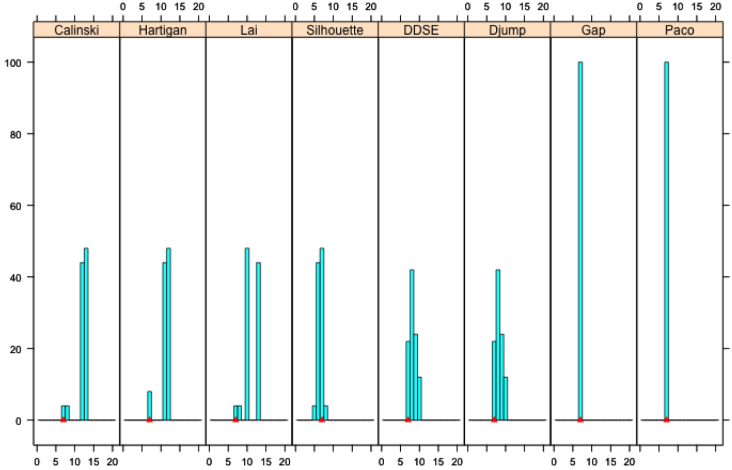
# Histograms (model 2)



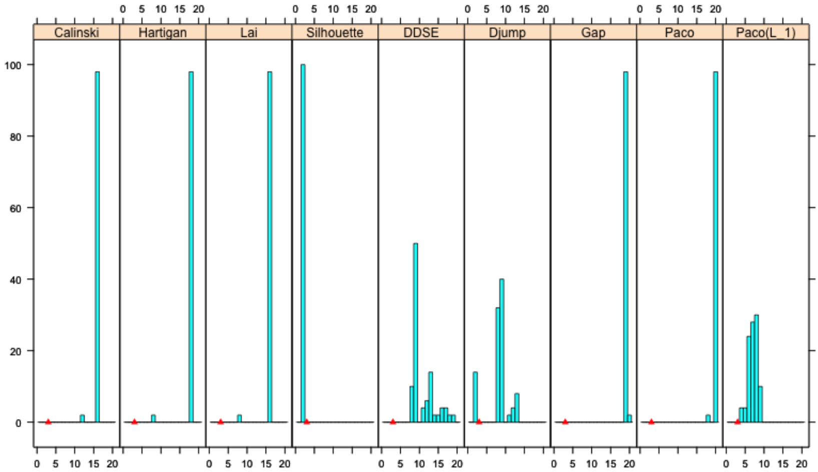
# Histograms (model 3)



# Histograms (model 4)



# Histograms (model 5)



# Online setting

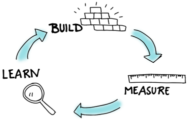
Live demo available here www.artfact-online.fr

- login : **cirm**
- pwd : **2016**

# Discussion

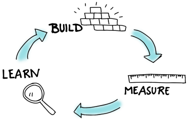
# Discussion

Remember...



# Discussion

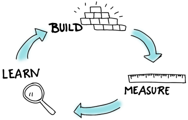
Remember...




- Online learning VS streaming clustering ?

# Discussion

Remember...

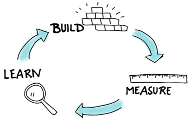



- Online learning VS streaming clustering ?

$$S_t(\mathbf{c}) = S_{t-1}(\mathbf{c}) + \ell(\mathbf{c}, x_t) + \frac{\lambda}{2}[\ell(\mathbf{c}, x_t) - \ell(\hat{\mathbf{c}}_t, x_t)]^2, \ \forall \mathbf{c} \in \mathbb{R}^{dp}.$$

# Discussion

Remember...




- Online learning VS streaming clustering ?

$$S_t(\mathbf{c}) = {\color{red}S_{t-1}(\mathbf{c})} + \ell(\mathbf{c}, x_t) + \frac{\lambda}{2}[\ell(\mathbf{c}, x_t) - \ell(\hat{\mathbf{c}}_t, x_t)]^2, \ \forall \mathbf{c} \in \mathbb{R}^{dp}.$$

- Extension to community detection

# Extensions to community detection

We extend the previous machinery to **community detection** in graphs.

# Extensions to community detection

We extend the previous machinery to **community detection** in graphs.

- We observe a sequence of small graphs $g_t = (\mathrm{edg}_t, \mathrm{ver}_t) \in (\mathcal{E}, \mathcal{V})$, $t = 1, \ldots$ where $\mathrm{card}\{\mathrm{edg}_t\} = m(t)$ the new set of edges

# Extensions to community detection

We extend the previous machinery to **community detection** in graphs.

- We observe a sequence of small graphs $g_t = (\mathrm{edg}_t, \mathrm{ver}_t) \in (\mathcal{E}, \mathcal{V})$, $t = 1, \ldots$ where $\mathrm{card}\{\mathrm{edg}_t\} = m(t)$ the new set of edges
- We want to cluster the graph at time $t$, with $\mathbf{c}_t \in \{0,1\}^{n^2}$ in order to maximize the modularity:

$$\ell(\mathbf{c}, g_t) := \frac{1}{m(t)} \sum_{(i,j) \in g_t} \left[ w_{ij} - \frac{k_i k_j}{2m(t)} \right] \delta_{\mathbf{c}}(v_i, v_j),$$

where $A = (w_{ij})$ is the adjacency matrix of $g_t$, $k_i$, $k_j$ are degrees from $v_i$ and $v_j$ and $\delta_{\mathbf{c}}(v_i, v_j) = 1$ if $v_i$ and $v_j$ are associated with the same community $\mathbf{c}$.

Thanks for your patience :-)