Augmented quantization: a general approach to mixture models

Charlie SIRE[1,2,3]

Supervisors: R. LE RICHE[3], D. RULLIERE[3], J. ROHMER[2], L. PHEULPIN[1], Y. RICHET[1]

[1]IRSN

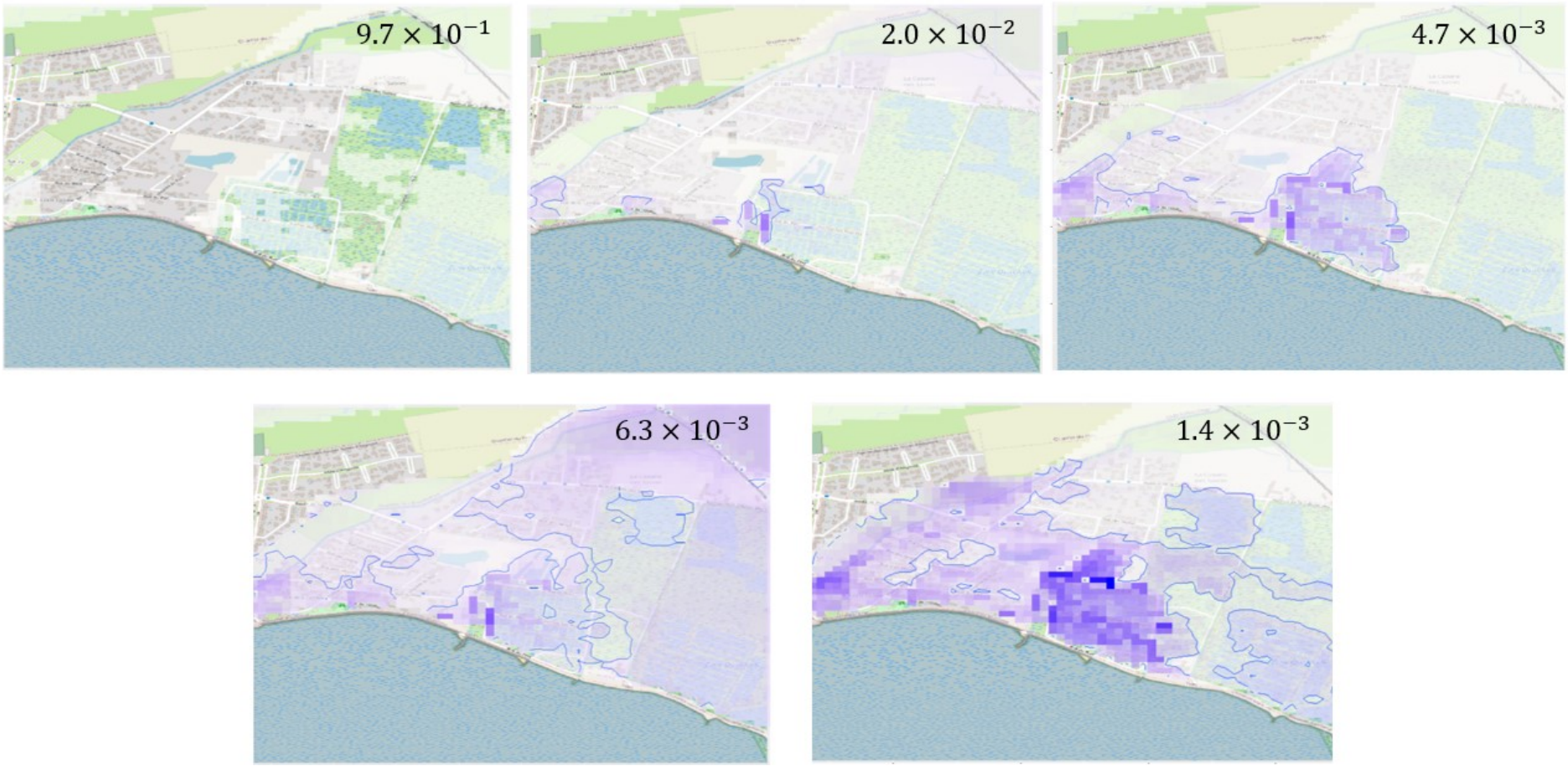
[2]BRGM

[3]Mines Saint-Etienne and CNRS,LIMOS

April 4th, 2023

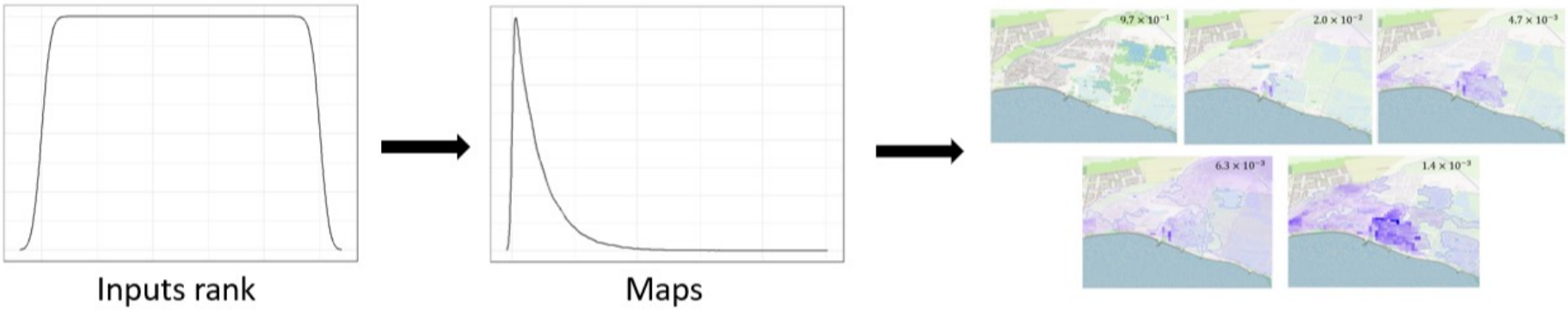
Content

Quantizing rare random maps

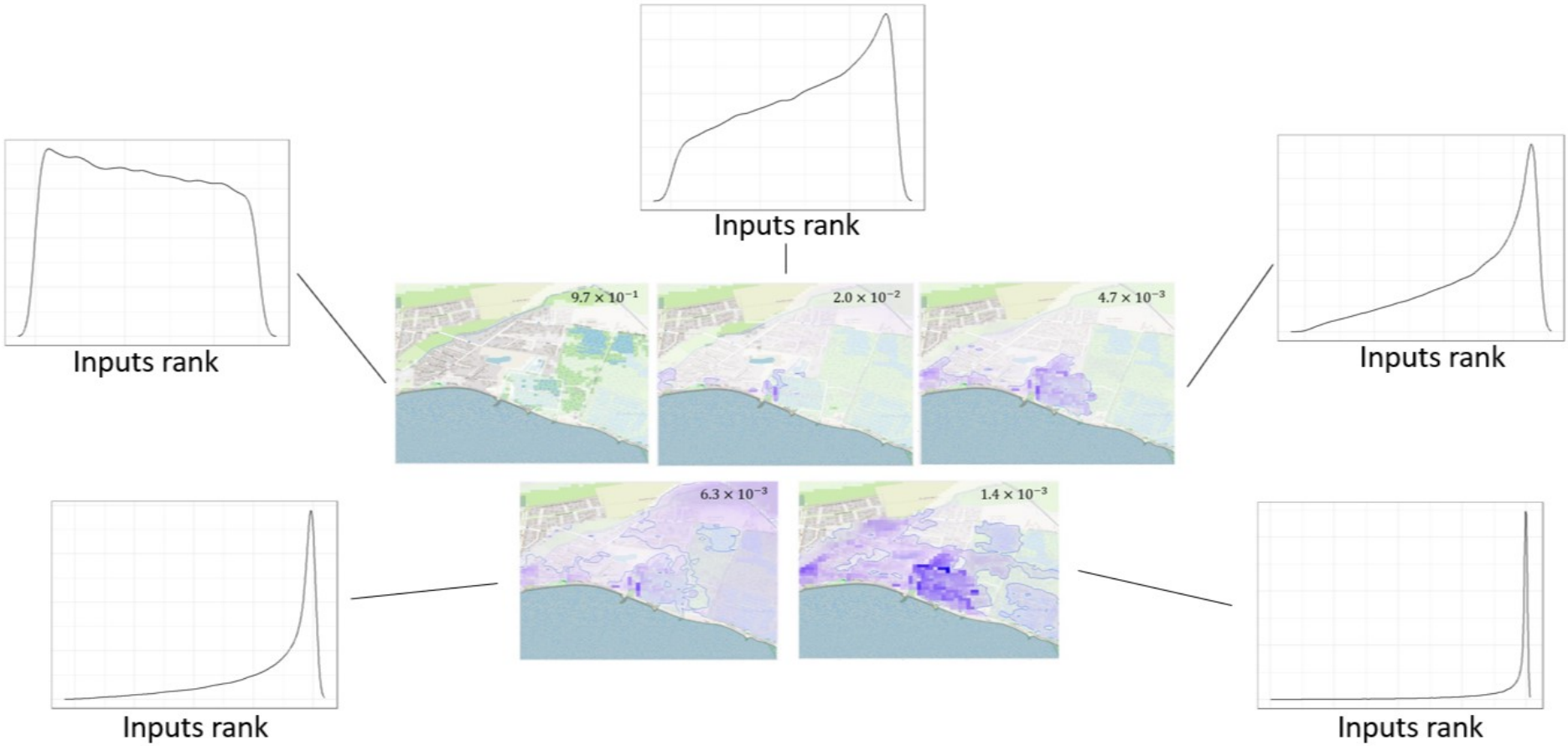


Details in [5]

Schematic method

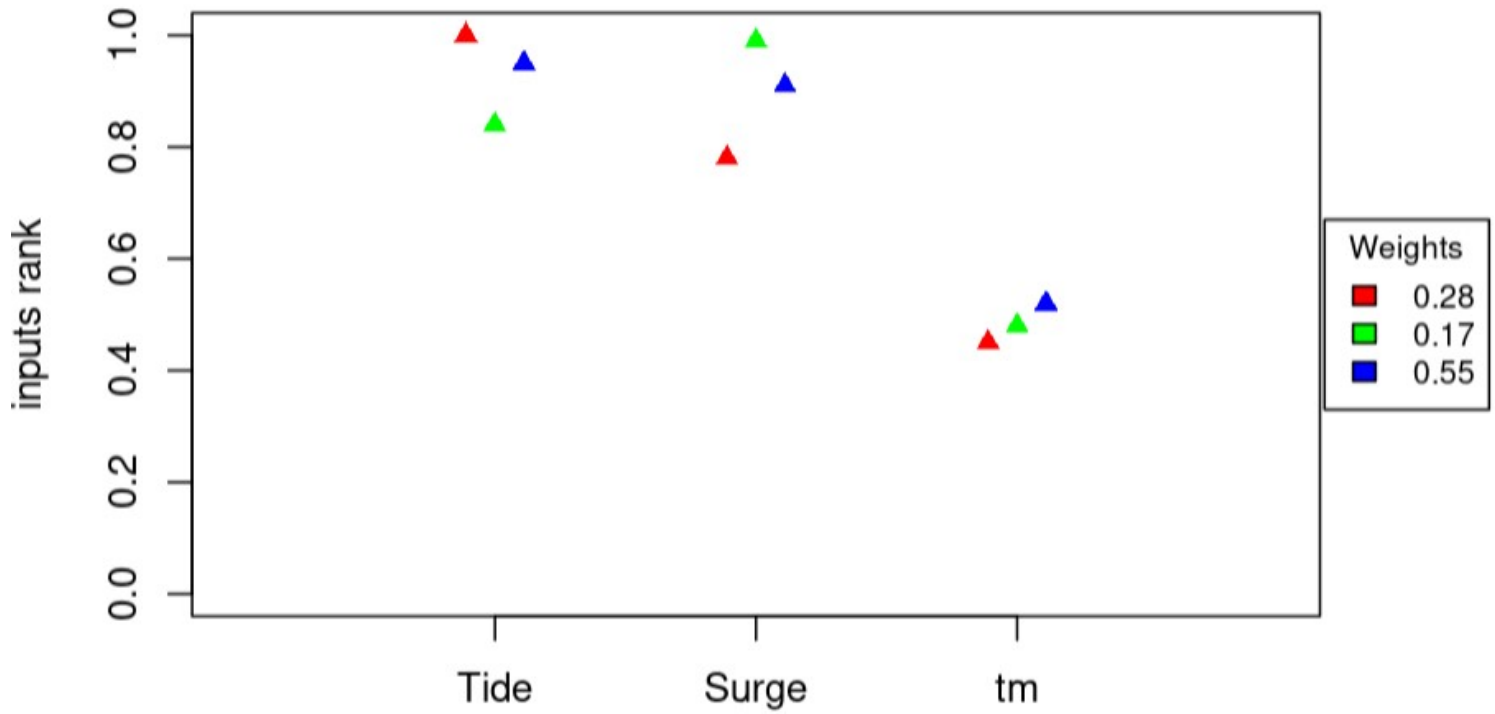


Work with the input space



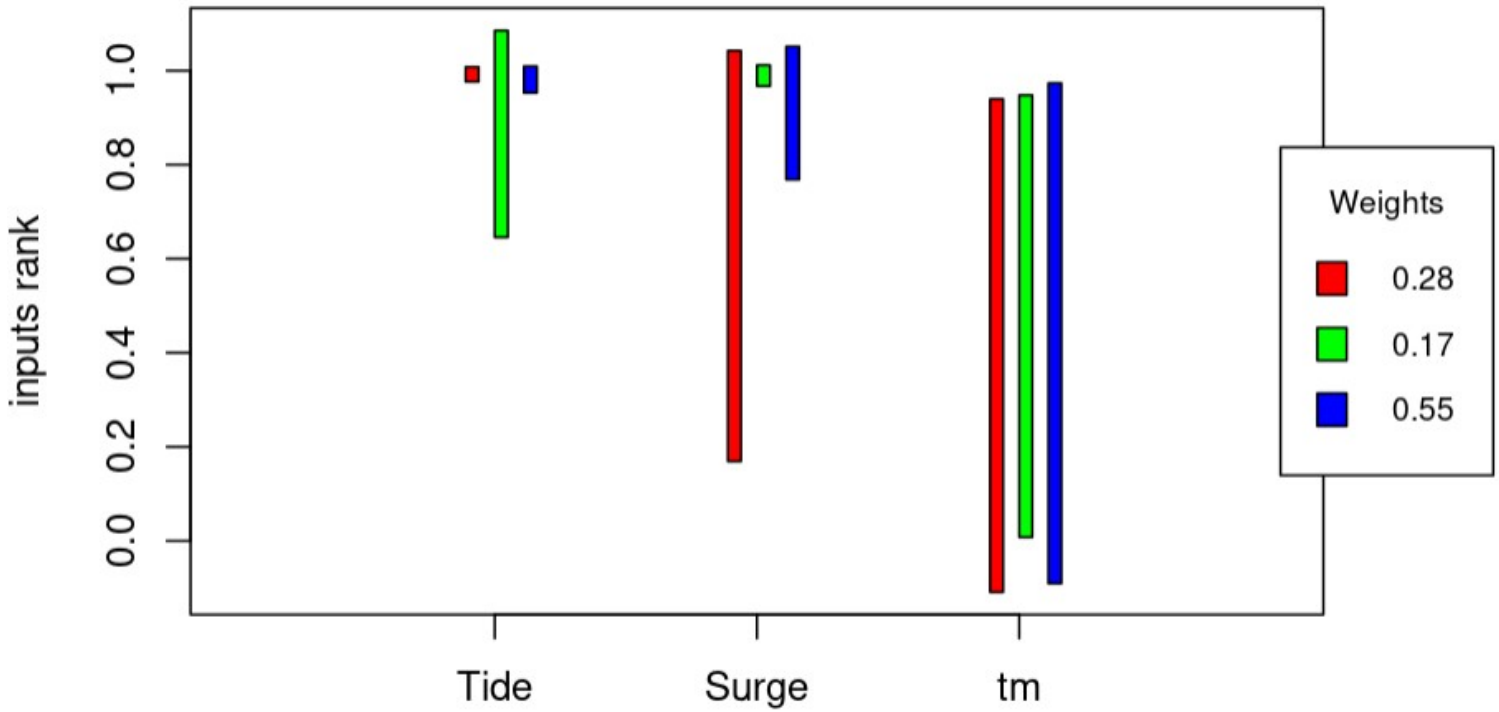
Sub-quantization

Perform a sub-quantization in every Voronoi cell to represent the inputs related to every prototype flooding



Mixture models

Work with uniform distribution instead of diracs on the marginals



Content

Augmented representation: mixture models

The classical clustering approach provides ℓ diracs with associated weights

Objective: Provide more complex representation with prototypes being continuous distributions

Idea: We investigate an approximation $\tilde{X}_\ell$ of a sample $(x_i)_{i=1}^n$

- $\tilde{X}_\ell = R^{(J)}$
- J a discrete random variable $\in \{1, \dots, \ell\}$ with weights denoted $(\omega_j)_{j=1}^\ell$
- $\forall j \in \{1, \dots, \ell\}, R^{(j)} \in \mathcal{R}$ a given family of distributions

Classical approach I

We have a sample $(x_i)_{i=1}^n \in \mathcal{X}^n$
Principle: Find $\Gamma_\ell = (\gamma_1, \dots, \gamma_\ell) \in \mathcal{X}^\ell$ minimizing [3]

$$\epsilon_p(\Gamma_\ell) = \left(\frac{1}{n} \sum_{i=1}^{n} \| x^{(i)} - \arg\min_{\gamma \in \Gamma_\ell} \| x^{(i)} - \gamma \| \|^p \right)^{\frac{1}{p}}$$

We have

$$\epsilon_p = \left(\sum_{j=1}^{\ell} \frac{\mathrm{card}(C_{\Gamma_\ell}^{(j)})}{n} \frac{1}{\mathrm{card}(C_{\Gamma_\ell}^{(j)})} \sum_{x \in C_{\Gamma_\ell}^{(j)}} \| x - \gamma_j \|^p \right)^{\frac{1}{p}}$$

$$= \left(\sum_{j=1}^{\ell} \frac{\mathrm{card}(C_{\Gamma_\ell}^{(j)})}{n} \mathcal{W}_p(C_{\Gamma_\ell}^{(j)}, \delta_{\gamma_j})^p \right)^{\frac{1}{p}}$$

Classical approach II

Algorithm Lloyd's algorithm

$\Gamma_\ell = \{\gamma^{(1)}, \dots, \gamma^{(\ell)}\} \in \mathcal{X}^\ell$, sample $(x^i)_{i=1}^n$ and X the associated r.v.

while stopping criterion not met **do**

Update clusters: $C_{\Gamma_\ell}^{(j)} = \{x^i, j = \underset{j' \in \{1,\dots,\ell\}}{\arg\min} \|x^i - \gamma^{(j')}\|\}, j = 1, \dots, \ell$

Update centroids: $\gamma^{(j)} \leftarrow \mathbb{E}\left[X \mid X \in C_{\Gamma_\ell}^{(j)}\right], j = 1, \dots, \ell$

end while

Classical approach III

Algorithm Rewritten Lloyd's algorithm

$R = \{R^{(1)}, \ldots, R^{(\ell)}\} \in \mathcal{X}^{\ell}$, sample $(x^i)_{i=1}^n$

while stopping criterion not met **do**

Update clusters: $(C^{(1)}, \ldots, C^{(\ell)}) \leftarrow \mathit{FindClusters}(R)$

Update representatives: $R^{(j)} \leftarrow \mathit{FindRepresentative}(C^{(j)}), j = 1, \ldots, \ell$

end while

Adaptation

Objective: Adapt the method and find $R = (R^{(1)}, \dots, R^{(\ell)}) \in \mathcal{R}$ and $C = (C^{(1)}, \dots, C^{(\ell)})$ minimizing $\epsilon_p(R, C) = \left(\sum_{j=1}^{\ell} \frac{\text{card}(C^{(j)})}{n} \mathcal{W}_p(C^{(j)}, R^{(j)})^p \right)^{\frac{1}{p}}$ with $\mathcal{R}$ a given family of distribution

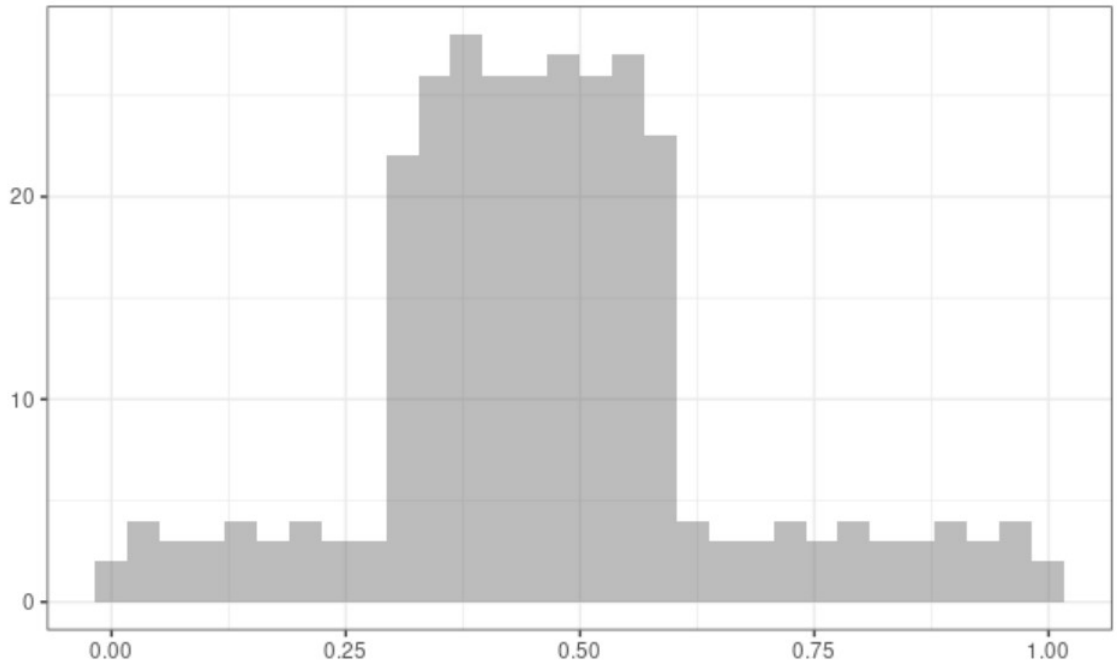
What we need:

- *FindClusters* providing clusters from representatives
- *FindRepresentative* providing representatives from clusters

Problem: Only *FindClusters* and *FindRepresentative* are not sufficient to be exploratory enough in the case of continuous distribution

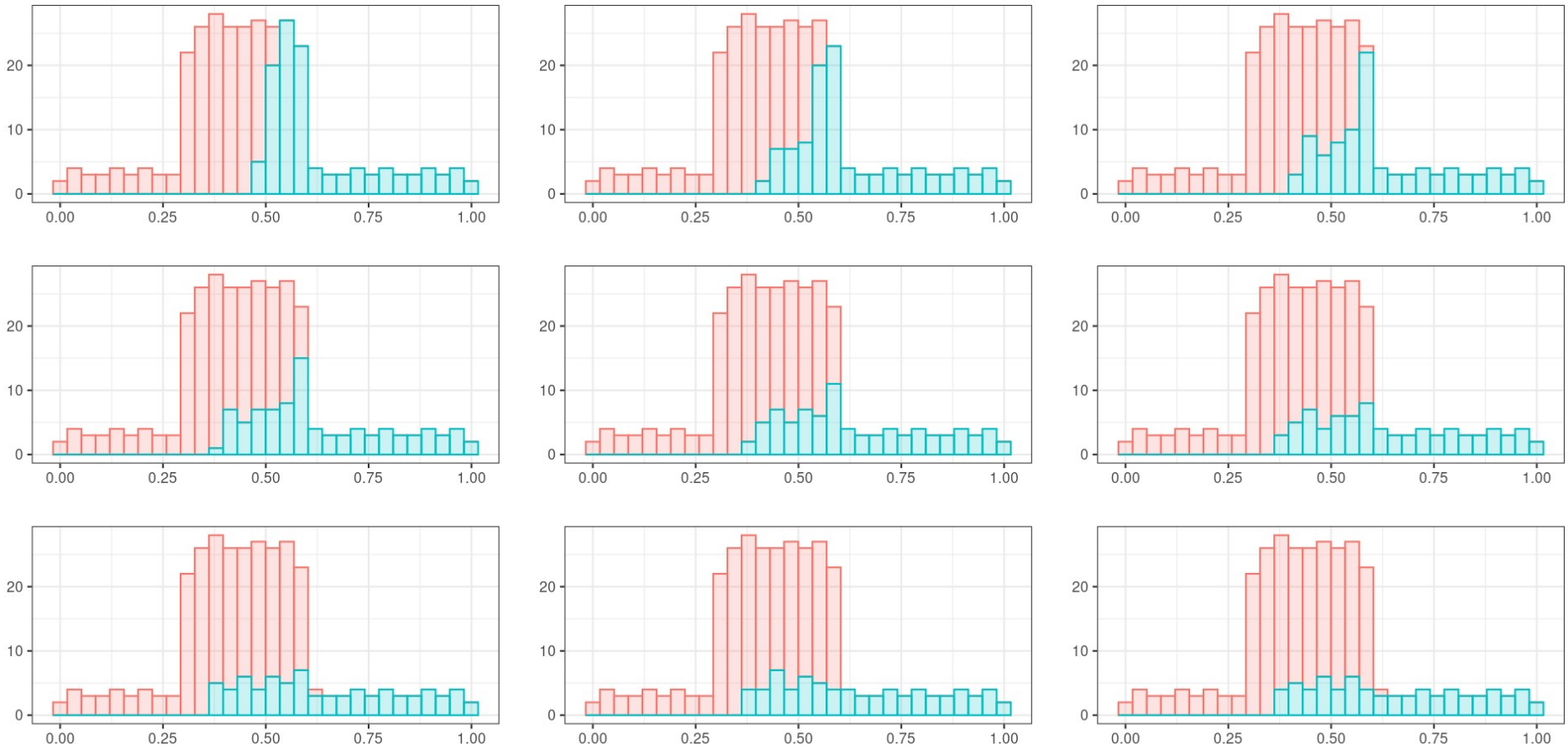
Illustrative sample $(x_i)_{i=1}^n$

$R_{\text{true}}^{(1)} \sim \mathcal{U}_{[0,1]}$ and $R_{\text{true}}^{(2)} \sim \mathcal{U}_{[0.3,0.6]}$
$P(J=1) = \frac{1}{3}$ and $P(J=2) = \frac{2}{3}$



Try to identify these two representatives, starting from $R^{(1)} \sim \mathcal{U}_{[0,0.5]}$ and $R^{(2)} \sim \mathcal{U}_{[0.5,1]}$

Exploration problem



Augmented quantization algorithm

To converge to the best R and C, a perturbation of the clusters must be added

Algorithm Augmented quantization algorithm

Input: $R = \{R^{(1)}, \dots, R^{(\ell)}\} \in \mathcal{R}^\ell$, sample $(x_i)_{i=1}^n$

Output: $\tilde{X}_\ell$

$(R_\star, C_\star, \epsilon_\star)) \leftarrow (\emptyset, \emptyset, +\infty)$

1: `while stopping criterion not met do`

`Update clusters:` $(C^{(1)}, \dots, C^{(\ell)}) \leftarrow \mathit{FindClusters}(R)$

`Perturb clusters:` $(C^{(1)}, \dots, C^{(\ell)}) \leftarrow \mathit{perturb}(C^{(1)}, \dots, C^{(\ell)})$

`Update representatives:` $\forall j \in \{1, \dots, \ell\}, R^{(j)} \leftarrow \mathit{FindRepresentative}(C^{(j)})$

`Update best configuration:` $(R_\star, C_\star, \epsilon_\star) = \mathit{UpdateBest}(R, C, R_\star, C_\star, \epsilon_\star)$

2: `end while`

3: $(\omega_1, \dots, \omega_\ell) = (\frac{\mathrm{card}(C_\star^{(1)})}{n}, \dots, \frac{\mathrm{card}(C_\star^{(\ell)})}{n})$

4: J `r.v.` $\in \{1, \dots, \ell\}$ `with` $\forall j \in \{1, \dots, \ell\}, \mathbb{P}(J = j) = \omega_j$

5: $\tilde{X} = R_\star^{(J)}$

Content

FindClusters

Objective: Associates a partition of ℓ clusters to the ℓ representatives

Inputs: $(x_i)_{i=1}^n$ and ℓ representatives $(R^{(1)}, \dots, R^{(\ell)})$

Outputs: Partition $C^{(1)}, \dots, C^{(\ell)}$

General idea: Greedily build the clusters $C^{(1)}, \dots, C^{(\ell)}$.

Consering S_j a large sample with distribution $R^{(j)}$

Add every x_i to cluster $j(x_i)$ that minimizes

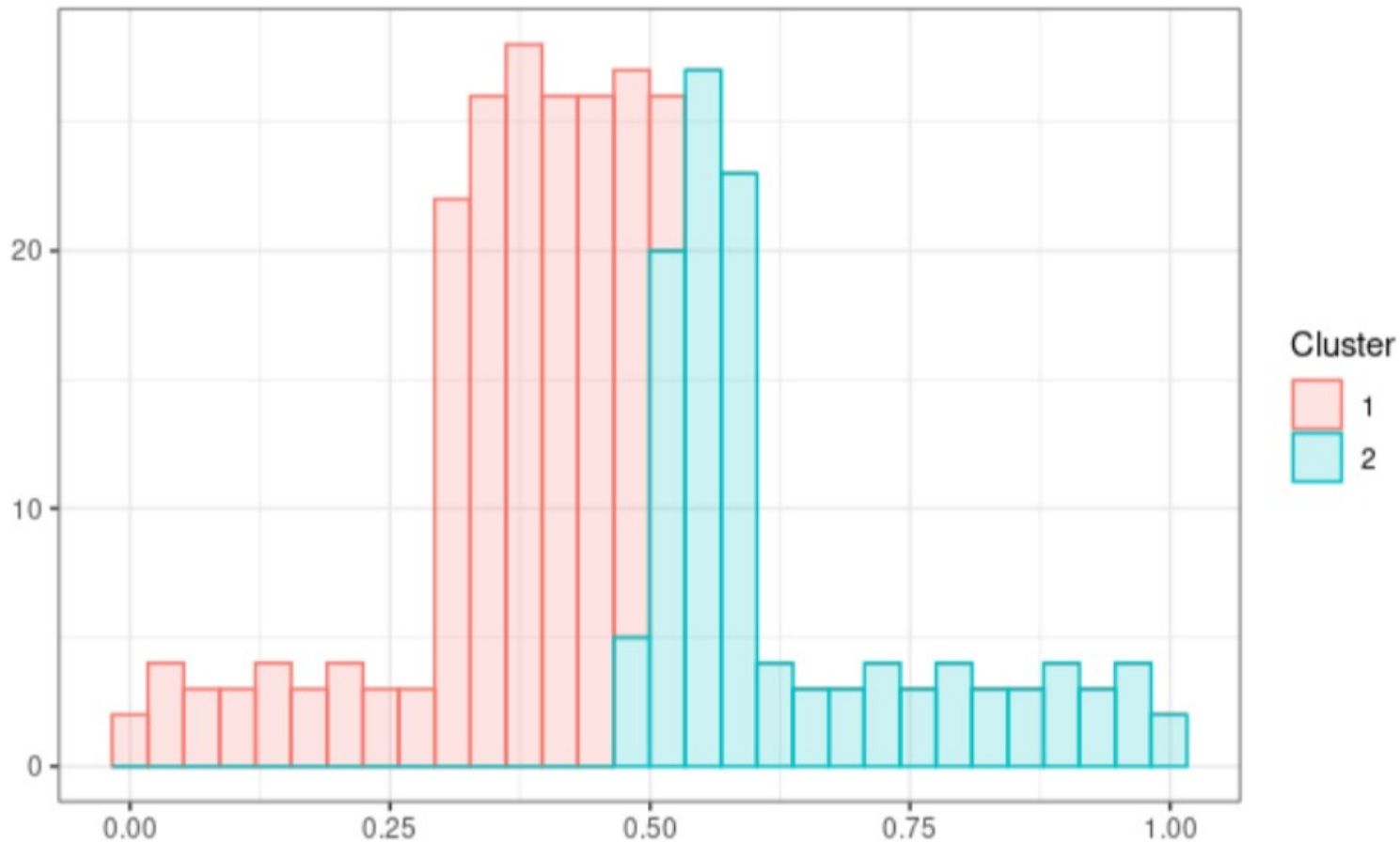
$$\delta(x_i, j) = \mathcal{W}_p(S_j \cup x_i, R^{(j)})^p - \mathcal{W}_p(S_j, R^{(j)})^p$$

$\delta(x_i, j)$ measures how x_i makes S_j different to $R^{(j)}$

Remark: At each iteration, $S_{j(x_i)} = S_{j(x_i)} \cup x_i$

FindClusters illustration

Start with $R^{(1)} \sim \mathcal{U}_{[0,0.5]}$ and $R^{(2)} \sim \mathcal{U}_{[0.5,1]}$



Content

Perturb part 1: Split

Objective: Split some of the clusters by identifying their worst elements to place in "bin" clusters

Inputs: $(x_i)_{i=1}^n$, a partition, $(C^{(1)}, \dots, C^{(\ell)})$, proportion of elements to remove p_{bin}, clusters to split $\text{indexes}_{\text{bin}}$

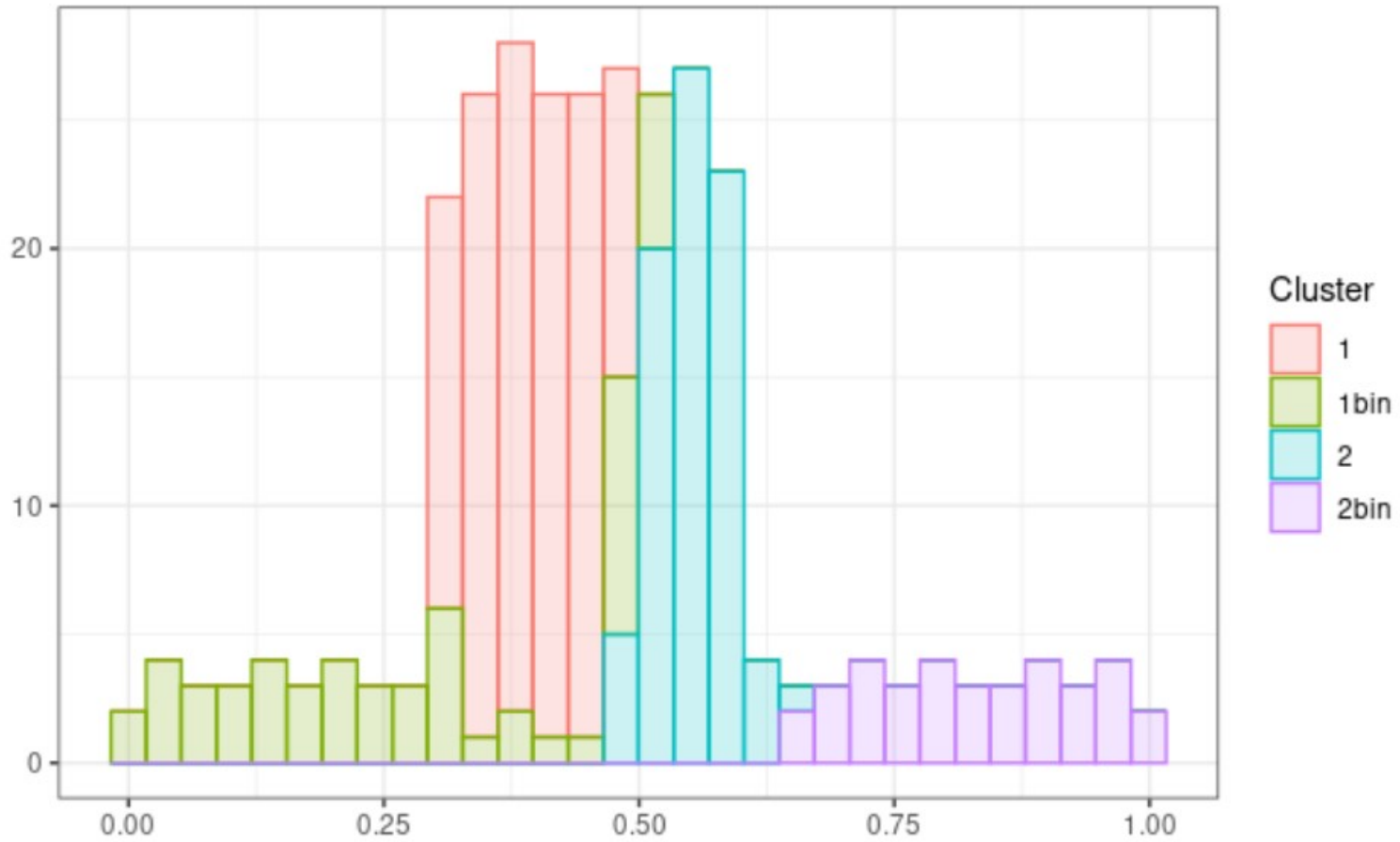
Outputs: Partition $\hat{C} = (C^{(1)}, \dots, C^{(\ell)}, C_{\text{bin}}^{(1)}, \dots, C_{\text{bin}}^{(\ell_{\text{bin}})})$

General idea: To split a cluster $C^{(j)}$, greedily fill $C_{\text{bin}}^{(j)}$ and empty $C^{(j)}$ by selecting

$$x^\star = \underset{x \in C^{(j)}}{\arg\min}\, \mathcal{W}_p(C^{(j)} \setminus x, \mathit{FindRepresentative}(C^{(j)} \setminus x))$$

$x^\star$ makes the cluster $C^{(j)}$ the closest to its representative once removed

Split illustration



Perturb part 2: Merge

Objective: Go back to ℓ clusters by merging some of the $\ell + \ell_{\text{bin}}$ clusters together

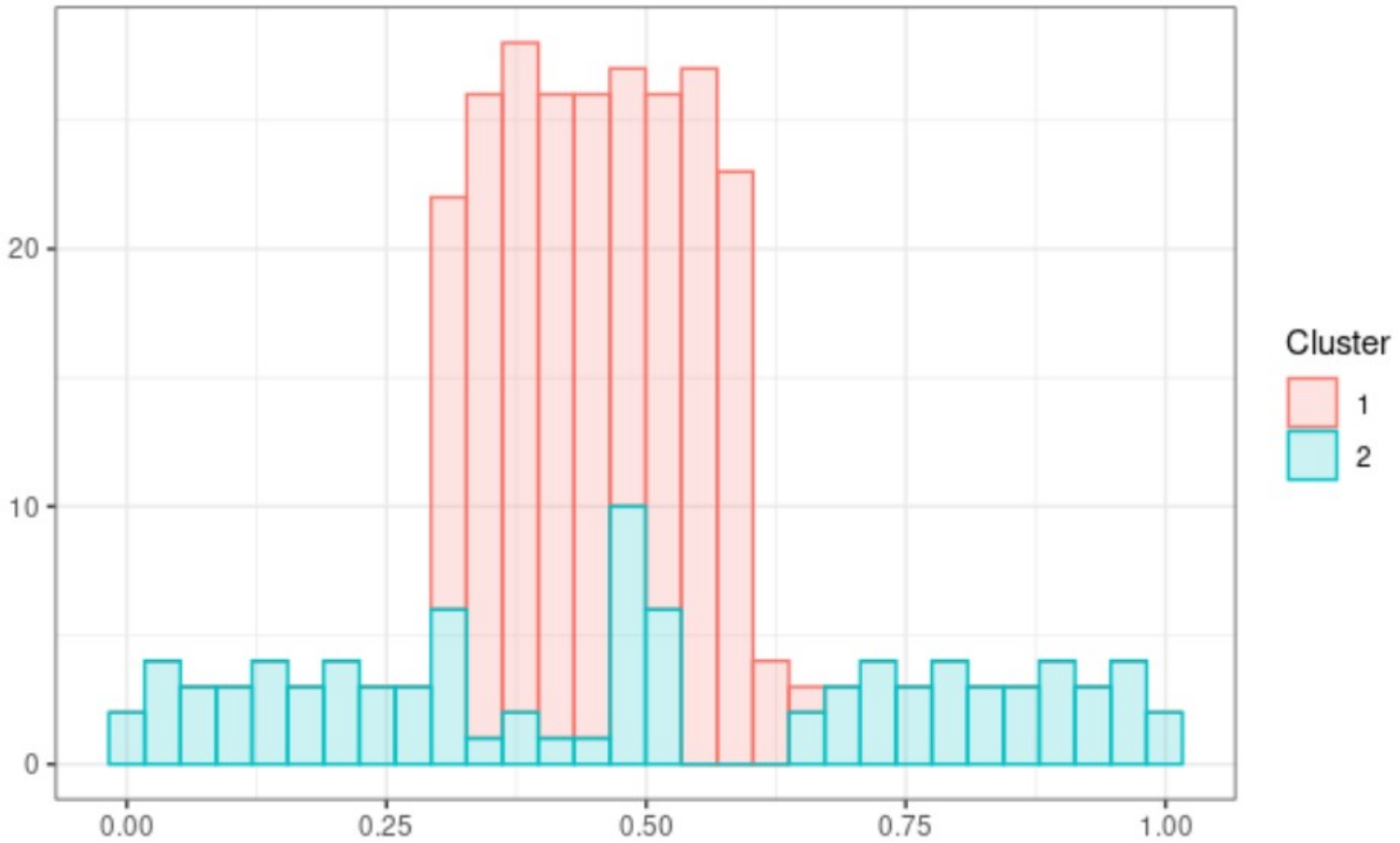
Inputs: $(x_i)_{i=1}^n$, a partition
$\hat{C} = (C^{(1)}, \ldots, C^{(\ell)}, C_{\text{bin}}^{(1)}, \ldots, C_{\text{bin}}^{(\ell_{\text{bin}})}) = (\hat{C}_1, \ldots, \hat{C}_{\ell+\ell_{\text{bin}}})$

Outputs: A partition $C_\star = (C^{(1)}, \ldots, C^{(\ell)})$

General idea: Testing all the possible merging to go from $\ell + \ell_{\text{bin}}$ groups to ℓ groups [1], keep the one with the lowest quantization error

$$\sum_{j=1}^{\ell} \omega_j \mathcal{W}_p(C^{(j)}, \mathit{FindRepresentative}(C^{(j)}))^p$$

Merge illustration



$R^{(1)} = \mathcal{U}_{(0.30,0.61)}$ and $R^{(2)} = \mathcal{U}_{(-0.02,1.01)}$

Content

FindRepresentative

Objective: Associate to a cluster a representative belonging to the parametric family $\mathcal{R} = \{r(\underline{\eta}), \underline{\eta} \in \mathbb{R}^d\}$

Input: A cluster $C^{(j)}$

Output: A representative distribution $r(\underline{\eta})$

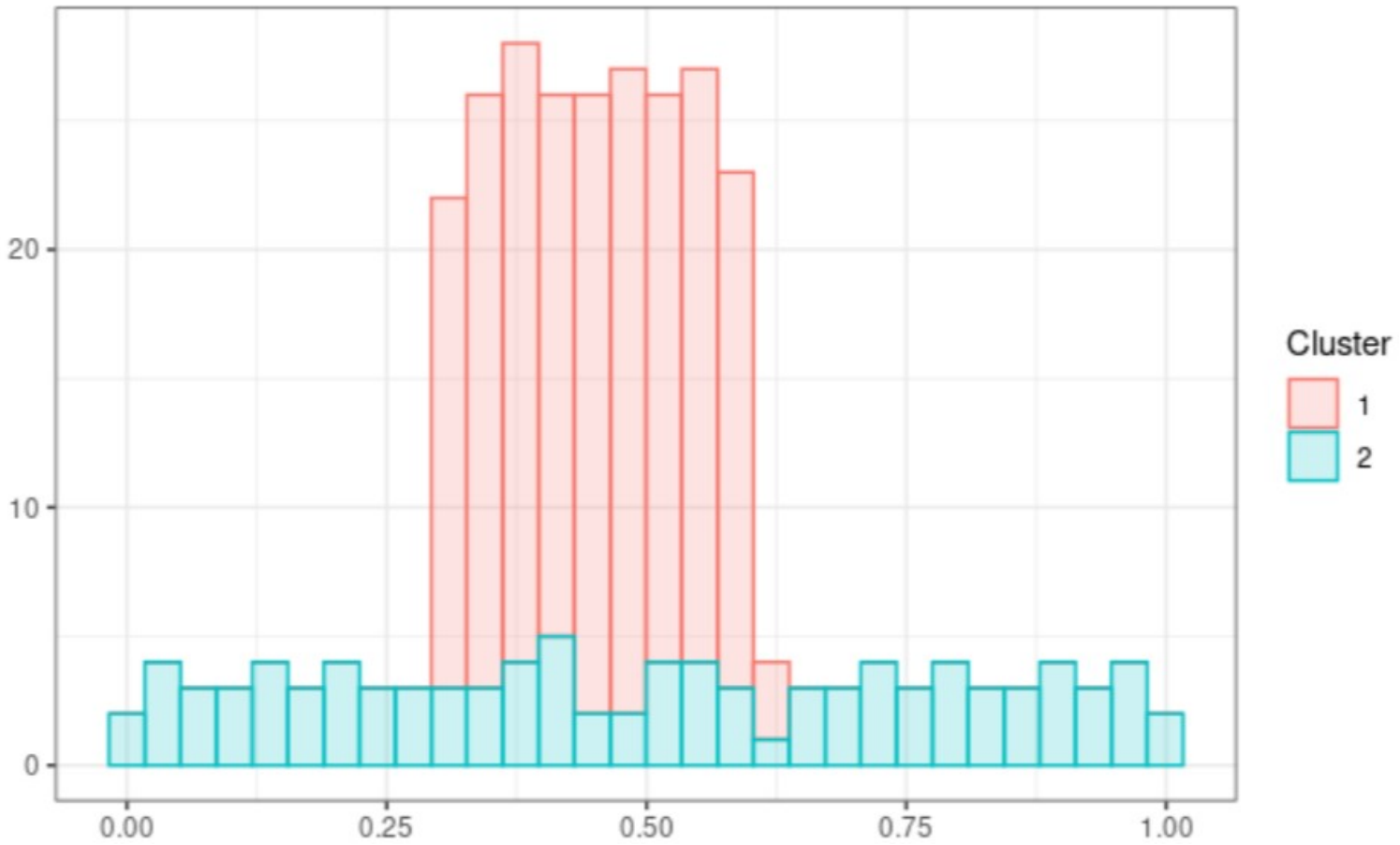
General idea: Minimise the Wasserstein distance between $r(\underline{\eta})$ and the cluster C: $\mathcal{W}_p(r(\underline{\eta}), C)$

Practically: find the best parameters for each marginal. By denoting $C^k = \{x_k, (x_1, \ldots, x_m) \in C\}$, we can optimize

$$\forall k \in \{1, \ldots, m\}, \mathcal{W}_p(r(\underline{\eta_k}), C^k)$$

Why ? In 1D, $\mathcal{W}_p(\mu_1, \mu_2) = \left(\int_0^1 \mid F_1^{-1}(q) - F_2^{-1}(q) \mid^p dq\right)^{\frac{1}{p}}$ [4]

New *FindClusters*



Back to the dirac case

FindRepresentative: Optimising $\mathcal{W}_p(C, \delta_{(x_1,\dots,x_m)})$

Provides $(x_1^\star, \dots, x_m^\star)$ the centroid of C

FindClusters: Build $(C^{(1)}, \dots, C^{(\ell)})$ from $\gamma_1, \dots, \gamma_\ell$

One can show that $x \in C^{(j)} \iff j \in \underset{j' \in \{1,\dots,\ell\}}{\arg\min} \|x - \gamma_{j'}\|$

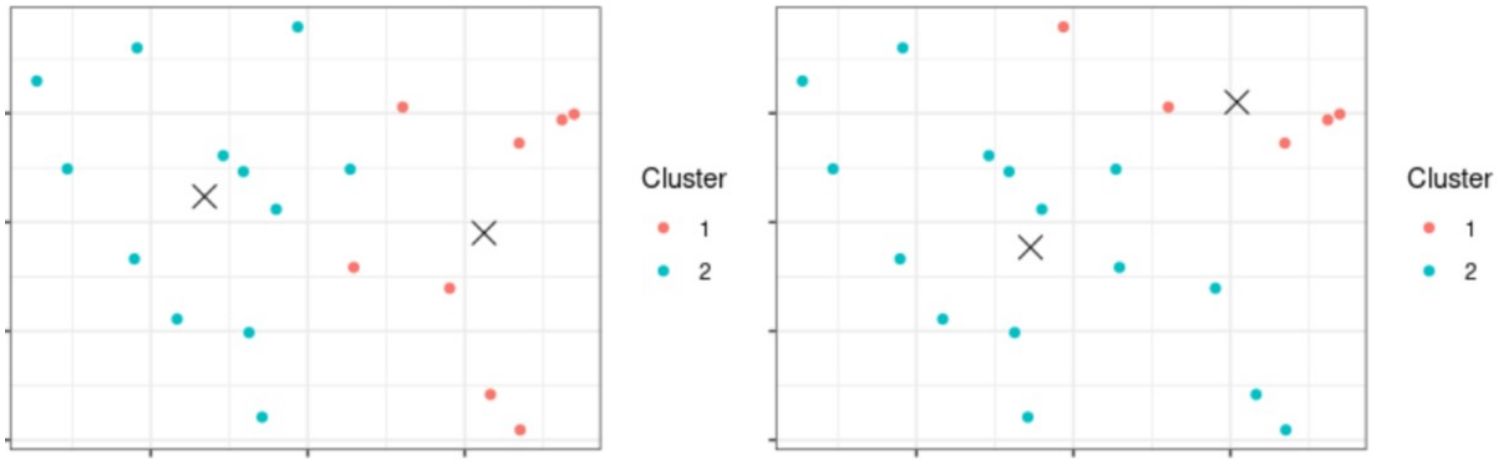
Conclusion These steps are the same as K-means

Content

Dirac

Without the clusters perturbation, our algorithm do the same as K-means

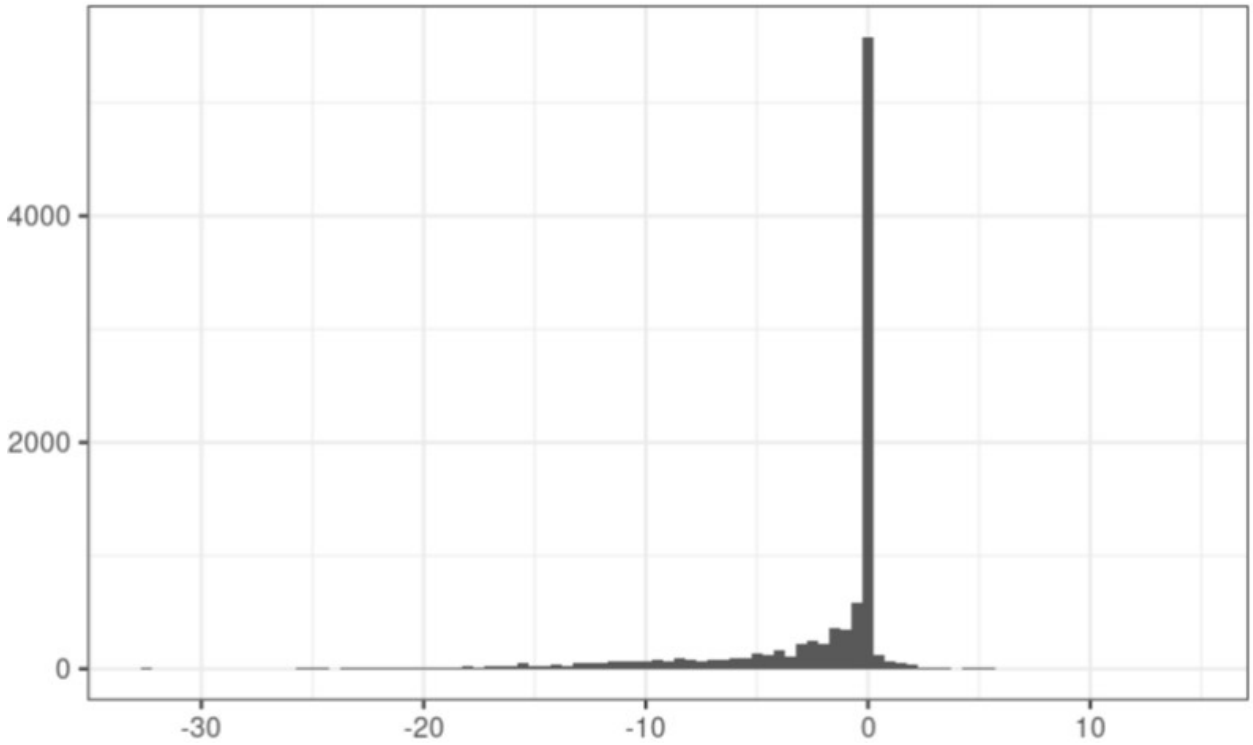
The perturbation can reduce the quantization error



Lloyd's algorithm ($\epsilon_2(\Gamma_2) = 0.28$) Augmented quantization ($\epsilon_2(\Gamma_2) = 0.25$)

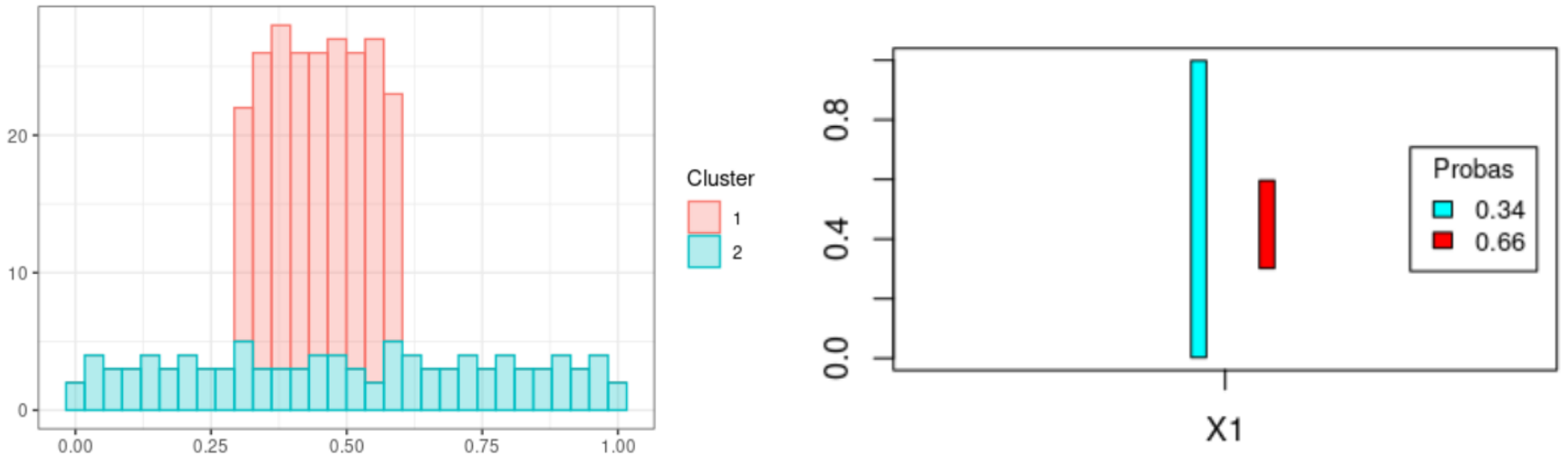
Dirac: statistical tests

Comparison on 500 different samples of 20 points in $[0, 1]^2$, with 20 starts tested for each one. Relative difference between the quantization errors (in %):



Lower quantization error for 43% of the tests
Same quantization error for 53% of the tests

Uniform mixtures

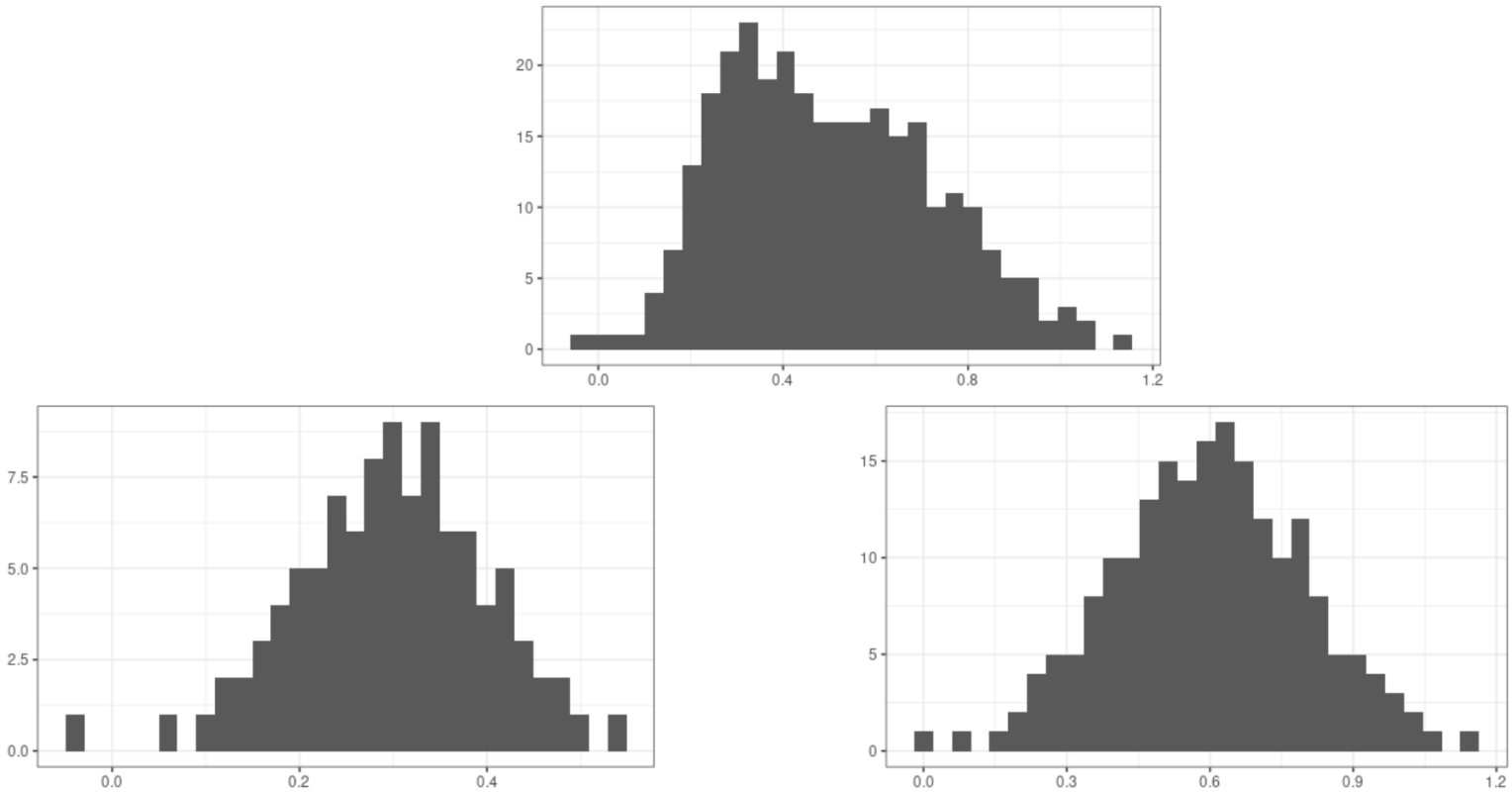


$$\left(\sum_{j=1}^{\ell} \omega_j \mathcal{W}_2(C^{(j)}, R^{(j)})^2\right)^{\frac{1}{2}} = 3.5 \times 10^{-3}$$

$$\mathcal{W}_2((x_i)_{i=1}^n, \tilde{X}) = 2.3 \times 10^{-3}$$

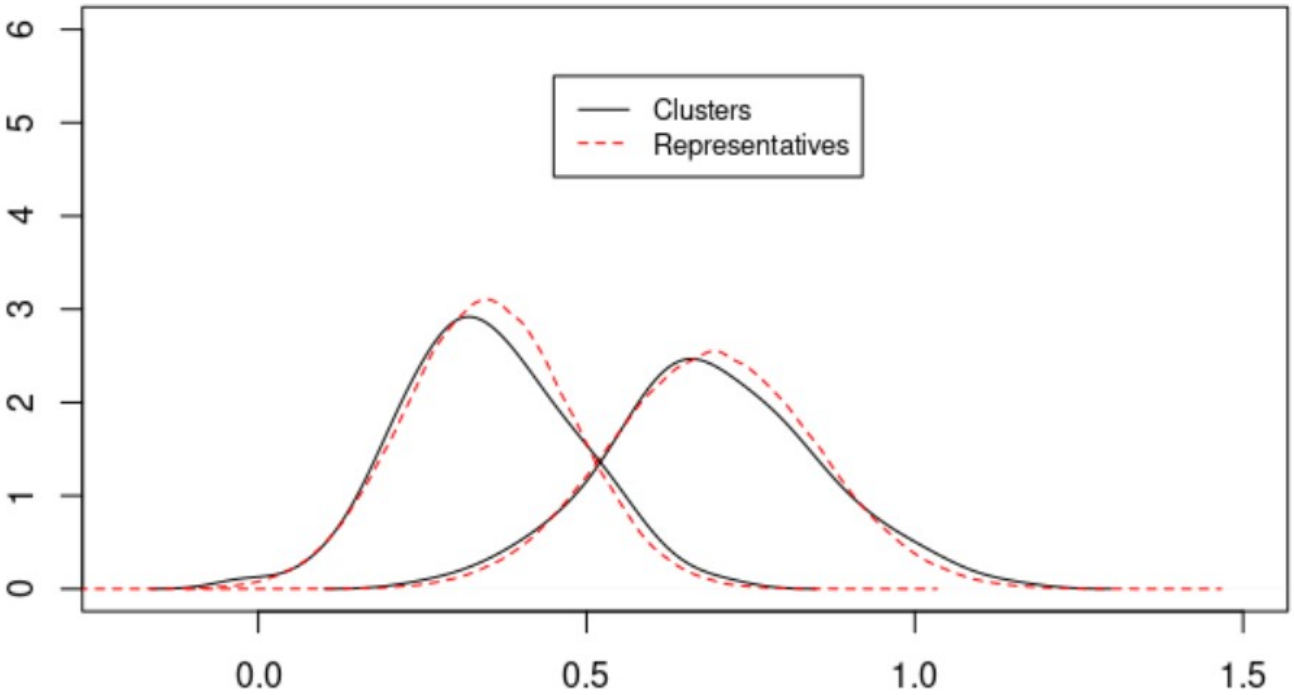
Gaussian mixtures

$$f_X = \frac{1}{3} f_{\mathcal{N}}\left(\frac{x - 0.3}{0.1}\right) + \frac{2}{3} f_{\mathcal{N}}\left(\frac{x - 0.6}{0.2}\right)$$



Gaussian mixtures II

Density of the 2 obtained clusters and the associated representatives

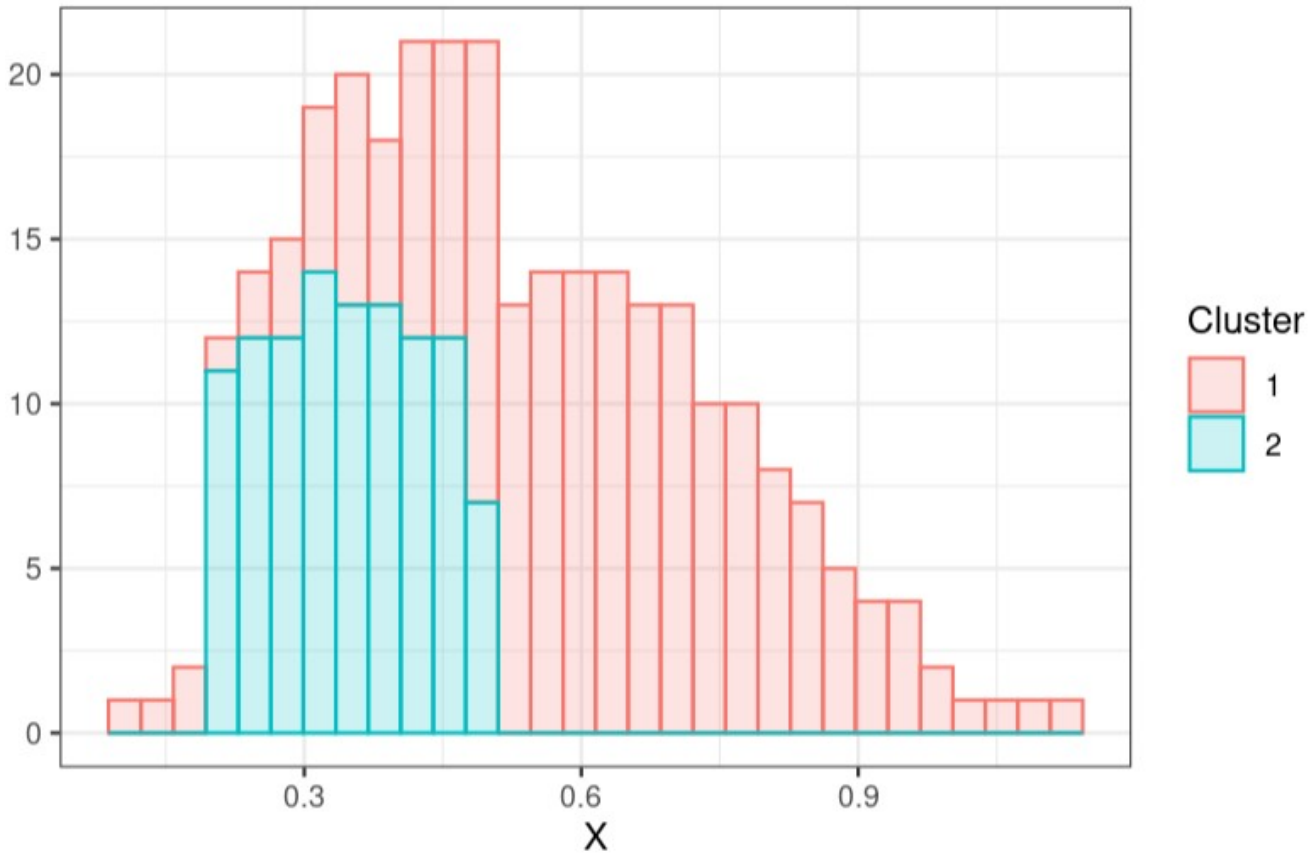


$$\left(\sum_{j=1}^{\ell} \omega_j \mathcal{W}_2(C^{(j)}, R^{(j)})^2\right)^{\frac{1}{2}} = 1 \times 10^{-2}$$

$$\mathcal{W}_2((x_i)_{i=1}^n, \tilde{X}) = 8.5 \times 10^{-3}$$

$$\mathcal{W}_2((x_i)_{i=1}^n, \tilde{X}_{\mathrm{GMM}}) = 7.5 \times 10^{-3} \text{ [2]}$$

Hybrid mixture



Summary and future work

Summary:

- Very general method to investigate mixture models
- Possibility to include different types of distributions
- Innovative approach but time consuming

Further developments:

- Investigate the optimization of p_{bin} in the perturb step
- Active learning of the number of representatives

Bibliography

[1] O-Yeat Chan and Dante V. Manna. "Congruences for Stirling Numbers of the Second Kind". In: 2009.

[2] Frank Dellaert. "The Expectation Maximization Algorithm". In: (July 2003).

[3] Gilles Pagès and Jun Yu. "Pointwise convergence of the Lloyd algorithm in higher dimension". working paper or preprint. Dec. 2013. URL: https://hal.archives-ouvertes.fr/hal-00922957.

[4] Victor M. Panaretos and Yoav Zemel. "Statistical Aspects of Wasserstein Distances". In: *Annual Review of Statistics and Its Application* 6.1 (2019), pp. 405–431. DOI: 10.1146/annurev-statistics-030718-104938.

[5] Charlie Sire et al. "Quantizing rare random maps: application to flooding visualization". In: July 2022.