



HAL
open science

Augmented Quantization: a General Approach to Mixture Models

Charlie Sire, Rodolphe Le Riche, Didier Rullière, Jérémy Rohmer, Lucie Pheulpin, Yann Richet

► **To cite this version:**

Charlie Sire, Rodolphe Le Riche, Didier Rullière, Jérémy Rohmer, Lucie Pheulpin, et al.. Augmented Quantization: a General Approach to Mixture Models. UQ 2024 - SIAM Conference on Uncertainty Quantification, Society for Industrial and Applied Mathematics, Feb 2024, Trieste, Italy. hal-04527349

HAL Id: hal-04527349

<https://hal.science/hal-04527349>

Submitted on 30 Mar 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Copyright

Augmented Quantization: a General Approach to Mixture Models

Charlie SIRE¹

Supervisors: R. LE RICHE³, D. RULLIERE³, J. ROHMER², L. PHEULPIN⁴, Y. RICHET⁴

¹Inria Saclay - Ecole Polytechnique

²BRGM

³Mines Saint-Etienne and CNRS,LIMOS

⁴IRSN

Content

- 1 Mixture Models
- 2 From K-means to Augmented Quantization
- 3 Algorithm steps
 - Find clusters
 - Perturb clusters
 - Find representative
- 4 Toy problems
- 5 Conclusion

Mixture models

Consider \mathcal{R} a family of probability measures, $\ell \in \mathbb{N}^*$ and $\mathcal{J} = \{1, \dots, \ell\}$.

Objective: Approximate the distribution of $(x_i)_{i=1}^n$ by the mixture R_J with

- $R_J = \sum_{j \in \mathcal{J}} p_j R_j$,
- $R_j \in \mathcal{R}, \quad j \in \mathcal{J}$,
- J a discrete random variable with weights $p_j = P(J = j), j \in \mathcal{J}$.

EM methods

Classical approaches for mixture models use the notion of likelihood (Dellaert 2003, Sridharan 2014, Delyon et al. 1999, McLachlan et al. 2019, Nguyen et al. 2020) but they are not adapted to all distributions:

- Problem of definition for Dirac distributions
- Problem of support for uniform distributions

⇒ Introduce an approach based on the quantization problem

Quantization error with Wasserstein

Sample $(x_i)_{i=1}^n \in \mathcal{X}^n$

Principle: Find $\Gamma_\ell = (\gamma_1, \dots, \gamma_\ell) \in \mathcal{X}^\ell$ minimizing

$$\mathcal{E}_p(\Gamma_\ell) := \left(\frac{1}{n} \sum_{i=1}^n \left\| x_i - \arg \min_{\gamma \in \Gamma_\ell} \|x_i - \gamma\| \right\|^p \right)^{\frac{1}{p}}$$

It can be written

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

with $C_j = \{x \in (x_i)_{i=1}^n : j = \arg \min_{j' \in \mathcal{J}} \|x - \gamma_{j'}\|\}$

Content

- 1 Mixture Models
- 2 From K-means to Augmented Quantization
- 3 Algorithm steps
 - Find clusters
 - Perturb clusters
 - Find representative
- 4 Toy problems
- 5 Conclusion

Augmented quantization

Objective: Find $\mathbf{R} = (R_1, \dots, R_\ell) \in \mathcal{R}^\ell$ and $\mathbf{C} = (C_1, \dots, C_\ell)$ minimising the *quantization error*

Quantization error: $\mathcal{E}_p(\mathbf{C}, \mathbf{R}) := \left(\sum_{j=1}^{\ell} \frac{\text{card}(C_j)}{n} \mathcal{W}_p(C_j, R_j)^p \right)^{\frac{1}{p}}$

Global error: $\epsilon_p(\mathbf{C}, \mathbf{R}) := \mathcal{W}_p \left(\bigcup_{j=1}^{\ell} C_j, R_J \right) = \mathcal{W}_p \left((x_i)_{i=1}^n, R_J \right)$
with $J \in \mathcal{J}$ a random variable such that $P(J = j) = \frac{\text{card}(C_j)}{n}$.

Proposition

The global error between a clustering \mathbf{C} and a set of representatives \mathbf{R} is lower than the quantization error between them:

$$\epsilon_p(\mathbf{C}, \mathbf{R}) \leq \mathcal{E}_p(\mathbf{C}, \mathbf{R}).$$

Lloyd's algorithm

Algorithm Lloyd's algorithm

Input: $(\gamma_1, \dots, \gamma_\ell) \in \mathcal{X}^\ell$, sample $(x_i)_{i=1}^n$

while stopping criterion not met **do**

Update clusters: $C_j \leftarrow \{x \in (x_i)_{i=1}^n : j = \arg \min_{j' \in \mathcal{J}} \|x - \gamma_{j'}\|\}$

Update representatives: $\gamma_j = \frac{1}{\text{card}(C_j)} \sum_{x \in C_j} x$

end while

J r.v. with $p_j = \mathbb{P}(J = j) = \frac{\text{card}(C_j)}{n}$, $j \in \mathcal{J}$

Output: $\sum_{j=1}^{\ell} p_j \delta_{\gamma_j}$

General Lloyd's algorithm

Algorithm Rewritten Lloyd's algorithm

Input: $R = (R_1, \dots, R_\ell) \in \mathcal{R}^\ell$, Sample $(x_i)_{i=1}^n$

while stopping criterion not met **do**

Update clusters: $C \leftarrow \text{FindC}(R)$

Update representatives: $R \leftarrow \text{FindR}(C)$

end while

J r.v. with $p_j = \mathbb{P}(J = j) = \frac{\text{card}(C_j)}{n}$, $j \in \mathcal{J}$

Output: R_J

Exploration problem

What we need:

- *FindC* providing clusters from representatives
- *FindR* providing representatives from clusters

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

Augmented quantization algorithm

Algorithm Augmented Quantization algorithm

Input: $R = (R_1, \dots, R_\ell) \in \mathcal{R}^\ell$, samples $(x_i)_{i=1}^n$

$J \in \mathcal{J}$ r.v. with $\mathbb{P}(J = j) = \frac{1}{\ell}$
 $(R^*, C^*, \mathcal{E}^*) \leftarrow (\emptyset, \emptyset, +\infty)$

while stopping criterion not met do

 Update clusters: $C \leftarrow \text{FindC}(R, J)$

 Perturb clusters: $C \leftarrow \text{Perturb}(C)$

 Update mixture: $R \leftarrow \text{FindR}(C)$, J r.v. with $\mathbb{P}(J = j) = \frac{\text{card}(C_j)}{n}$, $j \in \mathcal{J}$

 Update the best configuration:

 if $\mathcal{E}_p(C, R) < \mathcal{E}^*$ then $\mathcal{E}^* \leftarrow \mathcal{E}$, $C^* \leftarrow C$, $R^* \leftarrow R$, $J^* \leftarrow J$

end while

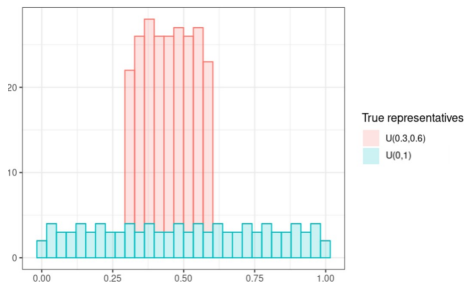
Output:

- the membership discrete random variable J^* with $\mathbb{P}(J^* = j) = \frac{\text{card}(C_j^*)}{n}$, $j \in \mathcal{J}$
 - the mixture $R_{J^*}^*$
-

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

$$\begin{cases} R_1^{\text{true}} &= R_U(0, 1), & P(J = 1) = \frac{1}{3} \\ R_2^{\text{true}} &= R_U(0.3, 0.6), & P(J = 2) = \frac{2}{3} \end{cases}$$

with $R_U(a, b)$ the measure associated to $\mathcal{U}(a, b)$



We start with $R_1 = R_U(0, 0.5)$ and $R_2 = R_U(0.5, 1)$
 We investigate $\mathcal{R} = \{R_U(a, b), a \leq b\}$

Content

- 1 Mixture Models
- 2 From K-means to Augmented Quantization
- 3 Algorithm steps
 - Find clusters
 - Perturb clusters
 - Find representative
- 4 Toy problems
- 5 Conclusion

FindC

Algorithm *FindC*

Input: Sample $(x_i)_{i=1}^n$, $\mathbf{R} = (R_1, \dots, R_\ell)$, N , J r.v. $\in \mathcal{J}$

$C_j = \emptyset$, $j \in \mathcal{J}$

$(j_i)_{i=1}^N$ N independent realizations of J

$(y_i)_{i=1}^N$ N independent realizations, y_i sampled with associated measure R_{j_i}

for $x \in (x_i)_{i=1}^n$ do

$l(x) \leftarrow \arg \min_{i=1, \dots, N} \|x - y_i\|$

$C_{j_{l(x)}} \leftarrow C_{j_{l(x)}} \cup x$

end for

Output: Partition $\mathbf{C}^*(\mathbf{R}, J, n, N) = (C_1, \dots, C_\ell)$

FindC

Algorithm *FindC*

Input: Sample $(x_i)_{i=1}^n$, $\mathbf{R} = (R_1, \dots, R_\ell)$, N , J r.v. $\in \mathcal{J}$

$C_j = \emptyset$, $j \in \mathcal{J}$

$(j_i)_{i=1}^N$ N independent realizations of J

$(y_i)_{i=1}^N$ N independent realizations, y_i sampled with associated measure R_{j_i}

for $x \in (x_i)_{i=1}^n$ do

$l(x) \leftarrow \arg \min_{i=1, \dots, N} \|x - y_i\|$

$C_{j_{l(x)}} \leftarrow C_{j_{l(x)}} \cup x$

end for

Output: Partition $\mathbf{C}^*(\mathbf{R}, J, n, N) = (C_1, \dots, C_\ell)$

FindC

Algorithm *FindC*

Input: Sample $(x_i)_{i=1}^n$, $\mathbf{R} = (R_1, \dots, R_\ell)$, N , J r.v. $\in \mathcal{J}$

$C_j = \emptyset$, $j \in \mathcal{J}$

$(j_i)_{i=1}^N$ N independent realizations of J

$(y_i)_{i=1}^N$ N independent realizations, y_i sampled with associated measure R_{j_i}

for $x \in (x_i)_{i=1}^n$ do

$l(x) \leftarrow \arg \min_{i=1, \dots, N} \|x - y_i\|$

$C_{j_{l(x)}} \leftarrow C_{j_{l(x)}} \cup x$

end for

Output: Partition $\mathbf{C}^*(\mathbf{R}, J, n, N) = (C_1, \dots, C_\ell)$

FindC

Algorithm *FindC*

Input: Sample $(x_i)_{i=1}^n$, $\mathbf{R} = (R_1, \dots, R_\ell)$, N , J r.v. $\in \mathcal{J}$

$C_j = \emptyset$, $j \in \mathcal{J}$

$(j_i)_{i=1}^N$ N independent realizations of J

$(y_i)_{i=1}^N$ N independent realizations, y_i sampled with associated measure R_{j_i}

for $x \in (x_i)_{i=1}^n$ do

$l(x) \leftarrow \arg \min_{i=1, \dots, N} \|x - y_i\|$

$C_{j_{l(x)}} \leftarrow C_{j_{l(x)}} \cup x$

end for

Output: Partition $\mathbf{C}^*(\mathbf{R}, J, n, N) = (C_1, \dots, C_\ell)$

FindC Convergence

Proposition

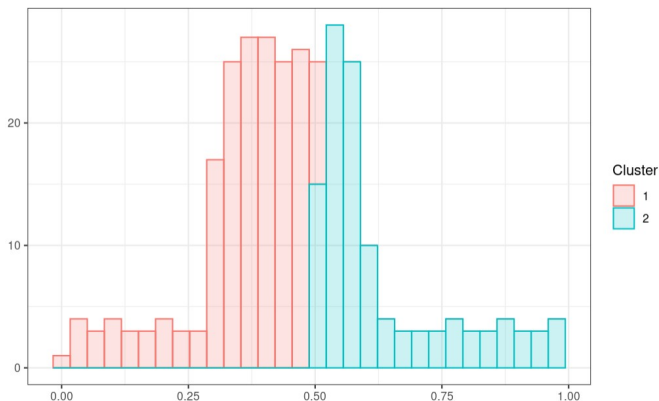
Under general assumptions on \mathcal{R} , if $(X_i)_{i=1}^n$ i.i.d. sample with probability measure R_J then

$$\lim_{n, N \rightarrow +\infty} \mathbb{E}(\mathcal{E}_p(\mathbf{C}^*(\mathbf{R}, J, n, N), \mathbf{R})) = 0.$$

where $\mathbf{C}^*(\mathbf{R}, J, n, N)$ comes from FindC

FindC illustration

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



Content

- 1 Mixture Models
- 2 From K-means to Augmented Quantization
- 3 Algorithm steps**
 - Find clusters
 - Perturb clusters**
 - Find representative
- 4 Toy problems
- 5 Conclusion

Perturb step

Objective: Split the clusters by identifying their worst elements, and identify the best merge regarding the quantization error

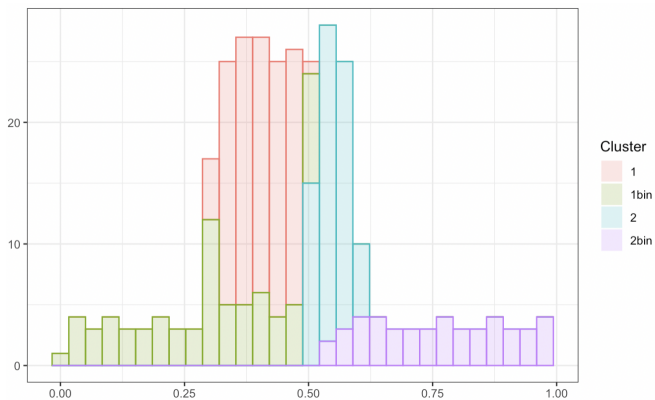
① $\mathbf{C} = (C_1, \dots, C_\ell)$

② $\hat{\mathbf{C}} = \text{split}(\mathbf{C}) = (C_1, \dots, C_\ell, C_{j_1}^{\text{bin}}, \dots, C_{j_{\ell_{\text{bin}}}}^{\text{bin}})$

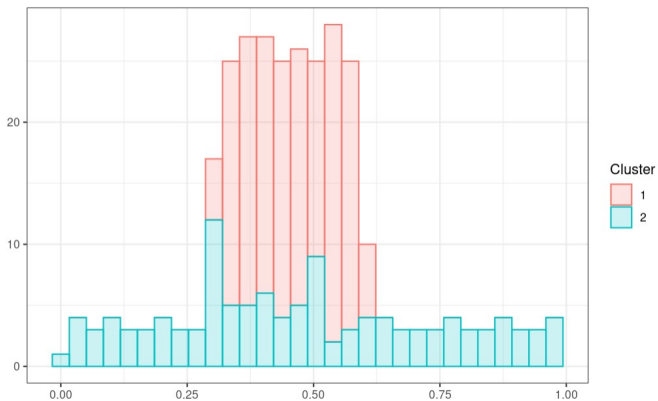
③ $\mathbf{C}^{\text{merge}} = \text{merge}(\hat{\mathbf{C}}) = (C_1^{\text{merge}}, \dots, C_\ell^{\text{merge}})$

Important point: The best merge can return to the clustering before the perturbation step

Split illustration



Merge illustration



Content

- 1 Mixture Models
- 2 From K-means to Augmented Quantization
- 3 Algorithm steps**
 - Find clusters
 - Perturb clusters
 - Find representative**
- 4 Toy problems
- 5 Conclusion

FindR

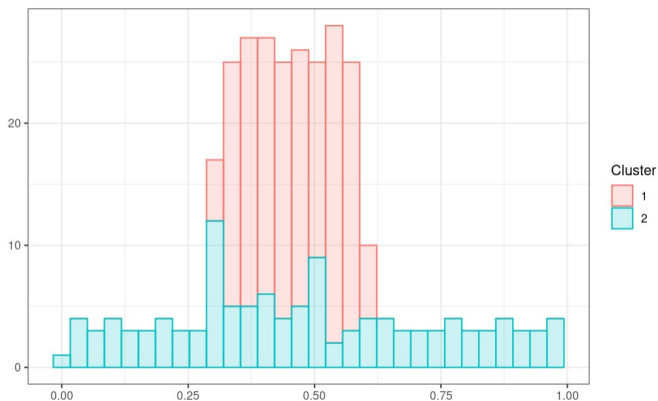
Objective: For a given clustering $\mathbf{C} = (C_1, \dots, C_\ell)$, find its optimal representatives $\mathbf{R}^*(\mathbf{C}) = (R_1^*(C_1), \dots, R_\ell^*(C_\ell))$ with

$$R_j^*(C_j) := \arg \min_{r \in \mathcal{R}} \mathcal{W}_p(C_j, r).$$

General idea: When \mathcal{R} is parametric, i.e. $\mathcal{R} = \{r(\underline{\eta}), \underline{\eta} \in \mathbb{R}^q\}$, find the best parameters η_k^* for each marginal.

Why ? In 1D, $\mathcal{W}_p(\mu_1, \mu_2) = \left(\int_0^1 |F_1^{-1}(q) - F_2^{-1}(q)|^p dq \right)^{\frac{1}{p}}$ Panaretos et al. 2019

FindR illustration



$$R_1 = R_U(0.30, 0.61) \text{ and } R_2 = R_U(-0.02, 0.94)$$

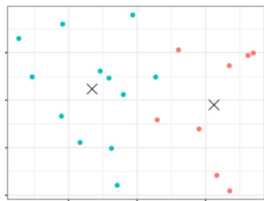
Content

- 1 Mixture Models
- 2 From K-means to Augmented Quantization
- 3 Algorithm steps
 - Find clusters
 - Perturb clusters
 - Find representative
- 4 Toy problems
- 5 Conclusion

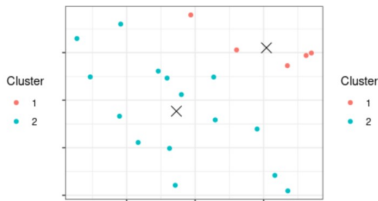
Dirac

- *FindC* creates Voronoï cells: $x \in C_j \iff j \in \arg \min_{j' \in \{1, \dots, \ell\}} \|x - \gamma_{j'}\|$
- *FindR* identifies the centroids of the clusters \mathbf{C}

Without the clusters perturbation, AQ is equivalent to K-means



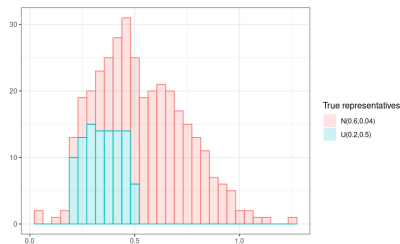
Lloyd's algorithm ($\epsilon_2(\Gamma_2) = 0.28$)



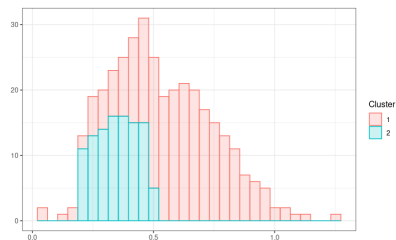
Augmented quantization ($\epsilon_2(\Gamma_2) = 0.25$)

Hybrid mixture

True representatives: $\mathcal{U}(0.2, 0.5)$ and $\mathcal{N}(0.60, 0.20^2)$.



(a) Distribution of a hybrid sample



(b) Estimated clusters

Estimated representatives: $R_U(0.21, 0.49)$ and $R_N(0.60, 0.20^2)$.

Content

- 1 Mixture Models
- 2 From K-means to Augmented Quantization
- 3 Algorithm steps
 - Find clusters
 - Perturb clusters
 - Find representative
- 4 Toy problems
- 5 Conclusion




Possible extensions

Article submitted to Statistics and Computing





<https://arxiv.org/abs/2309.08389>

- Method to optimize the covariance structure of the representatives
- Adapt AQ with alternative metric (e.g. MMD, Smola et al. 2006)

Bibliography I

-  Dellaert, Frank (July 2003). “The Expectation Maximization Algorithm”. In.
-  Delyon, Bernard, Marc Lavielle, and Eric Moulines (1999). “Convergence of a stochastic approximation version of the EM algorithm”. In: *Annals of statistics*, pp. 94–128.
-  McLachlan, Geoffrey J., Sharon X. Lee, and Suren I. Rathnayake (2019). “Finite Mixture Models”. In: *Annual Review of Statistics and Its Application* 6.1, pp. 355–378. DOI: [10.1146/annurev-statistics-031017-100325](https://doi.org/10.1146/annurev-statistics-031017-100325). eprint: <https://doi.org/10.1146/annurev-statistics-031017-100325>. URL: <https://doi.org/10.1146/annurev-statistics-031017-100325>.

Bibliography II

-  Nguyen, Hien D, Florence Forbes, and Geoffrey J McLachlan (2020). “Mini-batch learning of exponential family finite mixture models”. In: *Statistics and Computing* 30, pp. 731–748.
-  Panaretos, Victor M. and Yoav Zemel (2019). “Statistical Aspects of Wasserstein Distances”. In: *Annual Review of Statistics and Its Application* 6.1, pp. 405–431. DOI: 10.1146/annurev-statistics-030718-104938.
-  Smola, Alexander J, A Gretton, and K Borgwardt (2006). “Maximum mean discrepancy”. In: *13th international conference, ICONIP*, pp. 3–6.
-  Sridharan, Ramesh (2014). “Gaussian mixture models and the EM algorithm”. In.