Published in Image Processing On Line on YYYY–MM–DD.
Submitted on YYYY–MM–DD, accepted on YYYY–MM–DD.
ISSN 2105–1232 © YYYY IPOL & the authors CC–BY–NC–SA
This article is available online with supplementary materials,
software, datasets and online demo at
https://doi.org/10.5201/ipol

Clustering Multivariate Ordinal Data

Thomas Michel[1], Ali Ramlaoui[2], Théo Rudkiewicz[3]

[1] ENS Paris-Saclay, France (thomas.michel1@ens-paris-saclay.fr)
[2] ENS Paris-Saclay, France (ali.ramlaoui@ens-paris-saclay.fr) [3] TODO (TODO)

PREPRINT March 3, 2024

Contents

1 Introduction 2

2 Method 3
2.1 AECM algorithm. 4
2.2 Univariate model 4
2.2.1 Notations and goal 4
2.2.2 Optimization 5
2.3 Stochastic Binary Ordinal Search 6
2.3.1 Probabilistic model 6
2.4 BOS Model 7
2.5 Globally Ordered Data model 13
2.5.1 Probabilistic model 13
2.5.2 Parameter estimation 14

3 Experiments 15
3.1 Experimental setup 15
3.2 Estimation methods 17
3.3 Experiments with real-life datasets 17

4 Conclusion 18

A AECM tests for BOS and GOD distributions 20

B Metrics on real-world datasets 21

C t-SNE plots 21

D Assignment Matrices 24

E Histograms of the different clustering **25**

F GOD Model proofs **25**

Abstract

The abstract should contain about 100 to 150 words, and should be identical to the abstract text submitted electronically. An abstract must be able to stand alone, independent of the paper. Written in plain text, it cannot contain citations to the paper's references or equations or footnotes and should not, if possible, include special characters like math notations or greek letters, or hyperlinks. The abstract must be a single paragraph; multiple parts can be split with a single line break.

Source Code

The source code section briefly explains what the source code published with the article contains, all in a single paragraph. For example: The reviewed source code and documentation for this algorithm are available from the web page of this article[1]. Compilation and usage instruction are included in the `README.txt` file of the archive.

Keywords: first, second, third, fourth

1 Introduction

The exploration of hidden structures within datasets is a crucial task for data scientists, and clustering serves as a valuable tool in this endeavor. Mixture models have emerged as a standard approach for clustering due to their capacity to provide a well-defined mathematical framework for parameter estimation and model selection. These models, instrumental in determining the number of clusters, not only encapsulate classical geometric methods but also find successful application in diverse practical scenarios.

In the realm of model-based clustering, the classification of data hinges on the availability of a suitable probability distribution tailored to the nature of the data at hand—be it numerical, rankings, functional, or categorical. Notably, ordinal data, where categories possess a specific order, represent a common occurrence, especially in fields like marketing where product evaluations are solicited through ordinal scales. Despite their prevalence, ordinal data have received comparatively less attention in the context of model-based clustering. Often, practitioners resort to transforming ordinal data into quantitative or nominal formats to align with readily applicable distributions, neglecting valuable order information.

This paper explores the less-investigated domain of model-based clustering for ordinal data, specifically focusing on ordinal data derived from ordered categories. Ordinal data find widespread application in fields such as social sciences, psychology, marketing, healthcare, and more. They enable researchers to capture nuanced information, such as preferences, attitudes, or severity levels, in cases where continuous measures are neither significant nor possible. For example, when assessing tumor severity, the precise size may not be as crucial as the current state of development of the disease as it is assessed by specialists. The use of ordinal data enriches the comprehension of subjective opinions, behaviors, and hierarchical relationships across diverse research contexts. Over the years, various approaches have been proposed to define probability distributions for ordinal data, including modeling cumulative probabilities, constraining multinomial models to reflect ordinality, assuming ordinal data as discretization of continuous latent variables, and constructing distributions to meet specific properties.

[1] https://doi.org/10.5201/ipol

Among these approaches, the work by Biernacki and Jacques [2016], studied in the context of this project, delves into an original strategy—modeling the hypothetical data generating process for ordinal data. While this general principle has found success in ranking data scenarios, the distinction in the data generating process becomes apparent for ordinal data. In this context, a search algorithm, specifically the binary search algorithm, emerges as a fitting choice, respecting the ordinal nature of data through comparisons without necessitating links to nominal or continuous distributions.

The proposed model, parameterized with a position parameter (modal category) and a precision parameter, exhibits desirable properties such as a unique mode, probability distribution decrease on either side of the mode, and the flexibility to accommodate uniform or Dirac distributions. Maximum likelihood estimation using an EM algorithm is employed, leveraging the binary search algorithm's latent variable interpretation. While combinatorial complexity limits straightforward estimation for models based on latent Gaussian variables, the proposed approach remains tractable for ordinal data with up to eight categories—a common scenario for most ordinal variables.

In this project, we aim to replicate and build upon the findings of Biernacki and Jacques [2016]. We re-implemented their suggested probabilistic model, parameter estimation method, and model-based clustering algorithm in Python. Drawing inspiration from their approach, we propose an alternative probabilistic model with similar properties. The goal is to address computational limitations, enabling the clustering of more extensive datasets with potentially more categories than the previous method allows. We also present a preliminary analysis of this new method to justify the decreased computational cost of estimating parameters for this model. Additionally, we test Biernacki and Jacques [2016]'s approach on real-world datasets and compare it to the proposed approach, along with baseline models. This is done on different datasets of multiple nature in order to check whether the proposed methods are successful in these settings in practice and what their advantages are. The ultimate goal is to check whether the gains are significant against methods that are not adapted for ordinal datasets, in order to decide whether these approaches are interesting to use in general as a default method of choice for this type of variable.

2 Method

We suppose that we aim to cluster multivariate data. Each dimension has data generated by a random process parametrized by parameters that are dependant of the cluster. To estimate the cluster it is possible to use the AECM algorithm introduced in [Meng and Van Dyk, 1997]. This algorithm is quite generic and only requires to be able to estimate the univariate parameters of a weighted set of data points. We present this algorithm in section 2.1.

Similarly to Biernacki and Jacques [2016] we focus on ordinal data. Therefore we suppose that each dimension follow a process that generates ordinal categories. In section 2.2, we present two random processes to model ordinal data the binary ordinal search (BOS) model from Biernacki and Jacques [2016] and the globally ordered data (GOD) model.

To apply the proposed methods the each dimension of the data should represent an ordinal categorie. These categories must respect the following properties:

- The categories must be well-ordered (ordinal data): The categories are linearly ordered, and each non-empty subset contains the least element. This implies that any element can be compared to any other, and we can enumerate all the categories in increasing order.

- The set of categories is finite. This simplifies the previous assumption to the existence of a linear ordering. This assumption is necessary to ensure that the stochastic search terminates after a fixed number of steps, implying a finite number of possible runs of the search.

2.1 AECM algorithm.

Similarly to the EM algorithm, Alternating Expectation-Conditional Maximization (AECM) [Meng and Van Dyk, 1997] is separated in two steps. However, in this case, we consider multivariate ordinal data with different possible distributions (clusters) priors for the data. This is done, just like for the Gaussian Mixture Model case, using latent variables w_{ik} which describe whether the data x_i belongs to the cluster k or not, and parameters $(\alpha_k)_{k\in\{1,\dots,p\}}$ which describe the probability of belonging to each cluster.

1. Expectation step: In this case, the expectation step consists in just computing the probability for every data point to belong to each cluster:

$$\mathbb{P}(w_{ik}=1|x_i,\alpha^{(t)},\mu^{(t)},\pi^{(t)}) = \frac{\alpha_k^{(t)}\mathbb{P}(x_i|w_{ik}=1,\mu_k^{(t)},\pi_k^{(t)})}{\sum_{l=1}^{p}\alpha_l^{(t)}\mathbb{P}(x_i|w_{il}=1,\mu_l^{(t)},\pi_l^{(t)})}. \quad (1)$$

2. Maximization step: The parameters are updated using the new expected values for belonging to the different cluster. Since there are two groups of latent variables, the clusters variables $\alpha_k^{(t)}$ are updated first to maximize the log-likelihood:

$$\alpha_k^{(t+1)} = \frac{1}{n}\sum_{i=1}^{n}\mathbb{P}(w_{ik}=1|x_i,\alpha^{(t)},\mu^{(t)},\pi^{(t)}). \quad (2)$$

   And then the parameters $(\mu_k^{(t+1)},\pi_k^{(t+1)})$ are updated after using an EM algorithm in the univariate case for every cluster k and for every dimension of the multivariate variables independently using the data on the corresponding dimension.

2.2 Univariate model

We now wan't a random process to model univariate ordinal data among a finite numbers of categories. As we suppose that we only care for the order of the categories we can can without loss of generality consider our categories as $[\![1,m]\!]$ (when there is m categories). Therefore if we have θ the parameters of our model, a model is gives $\forall i\in[\![1,m]\!]\,, P(X=i|\theta)$ if X was generated from our random process.

As we should represent data have having a common source we can suppose that there is an underlying true category $\mu\in[\![1,m]\!]$ and put it as a parameter. In addition it is natural to add a precision parameter π. This is the case for the BOS model and the GOD model.

In the following sections we present how to estimate those parameters for a generic law, then we present the BOS model and how to apply this estimation technique then we do the same with the GOD model.

2.2.1 Notations and goal

Let suppose that we have a set of n independant observations $X=(x_i)_{i\in[n]}$, wehre $x_i\in[\![1,m]\!]$ follow a distribution P with parameters μ,π with $\mu\in[\![1,m]\!]$ and $\pi\in[[a,b]]$. We want to estimate μ and π. We choose the estimate that maximize the likelihood of the data.

$$(\mu,\pi) = \underset{(\mu,\pi)\in[\![1,m]\!]\times[a,b]}{\operatorname{argmax}} P(X|\mu,\pi)$$

As the data are independant, we have:

$$P(X|\mu,\pi) = \prod_{i=1}^{n}P(x_i|\mu,\pi)$$

As the number of possible values for x is finite, we can group the data by values and count the number of occurences of each value. Let n_i be the number of occurences of i in the data. We have:

$$P(X|\mu,\pi) = \prod_{i=1}^{m} P(i|\mu,\pi)^{n_i}$$

In the AECM algorithm, each data point has a weight w_i. As previously, we can suppose without loss of generality that we have only one observation of each value with a specific weight (we can always group the data by values and sum the weights of the observations). With the weights $W \in \mathbb{R}_+^n$ where w_i is the weight of the value i, we can write the weighted likelihood as:

$$P(W|\mu,\pi) = \prod_{i=1}^{m} P(i|\mu,\pi)^{w_i}$$

We can also write the weighted log-likelihood as:

$$L_W(\mu,\pi) := \log P(W|\mu,\pi) = \sum_{i=1}^{m} w_i \log P(i|\mu,\pi)$$

2.2.2 Optimization

To estimate μ and π, the idea proposed for the BOS-model in REF is to use the Expectaion-Maximization algorithm. However they note that it is easier to first estimate π for every possible value of μ and then to estimate μ using the estimated π. In forulas, we have:

$$\hat{\pi_\mu} = \underset{\pi\in[[a,b]]}{\operatorname{argmax}} P(W|\mu,\pi)$$

$$\hat{\mu} = \underset{\mu\in[[1,m]]}{\operatorname{argmax}} \max_{\pi\in[[a,b]]} P(W|\mu,\pi) = \underset{\mu\in[[1,m]]}{\operatorname{argmax}} P(W|\mu,\hat{\pi_\mu})$$

Once we have the estimates $\hat{\pi_\mu}$ for every possible value of μ, it is easy to estimate μ by choosing the value of μ that maximize the weighted likelihood as it requires only to compute the likelihood for every possible value of μ and to choose the maximum.

Estimating π for a given μ is a one-dimensional optimization problem. We can use the EM algorithm to solve it but it may be possible to use a direct optimization algorithm. For example if the function $\pi \mapsto L_W(\mu,\pi)$ is stricly concave (or constant), we can the ternary search algorithm (or trisection algorithm) to find the maximum.

Concavity of the weighted likelihood Suppose we have:

$$\forall x \in [[1,m]], \pi \mapsto P(x|\mu,\pi) \text{ is strcictly log-concave}$$

Then we have, $\pi \mapsto L_W(\mu,\pi)$ is stricly concave as positive linear combination of concave functions are concave.

Ternary search algorithm

TR: TODO

Evaluating the likelihood To run the ternary search algorithm, we need to be able to evaluate the log-likelihood for a given value of π efficiently. The log-likelihood is the sum of m log of the likelihoods of individual values. Hence the complexity will be $\Theta(mC_E(m))$ where $C_E(m)$ is the complexity of computing the likelihood for a single value of x.

In the case of the BOS model, we can notice that $\forall x \in [[1,m]], \pi \mapsto P(x|\mu,\pi)$ is polynomial of degree $\Theta(m)$ with coefficients that depend on m, i and μ. This is alsmost alos the case for the GOD mdoel in the sense that the following reasonning holds. As the function is polynomial, we can precompute these coefficients and then evaluate the likelihood of one value in $\Theta(m)$ operations. This gives a complexity of $\Theta(m^2)$ to evaluate the total likelihood of W for a given μ and π.

An important point is we excluded the cost of the precomputations of the coefficients. This cost is not necessary negligible but countrary to the evaluation of the likelihood, it is only done once for one m and can be then used for all iterations of the AECM algorithm and the ternary search algorithm.

Complexity We note n the numbre of observations, m the number of possible values for x and $\varepsilon > 0$ the precision of the estimate of π.

We can first group the data by values and sum the weights of the observations. This can be done in $\Theta(n)$ operations.

Then to estimate π for a given μ, we can use the ternary search algorithm. The number of iterations of the algorithm is $\Theta(\log \frac{b-a}{\varepsilon})$. For each iteration, we need to compute the likelihood for two values of π. If we note $C_E(m)$ the complexity of computing the likelihood for a given value of π, we have a complexity of $\Theta(\log \frac{b-a}{\varepsilon} C_E(m))$.

Finally, to estimate μ, we need to compute the likelihood for every possible value of μ. This gives a total complexity of $\Theta(mC_E(m) \log \frac{b-a}{\varepsilon})$.

In our case we have $C_E(m) = \Theta(m^2)$ and $[a,b] \subset [0,1]$ which gives a complexity of $\Theta(m^3 \log \frac{1}{\varepsilon})$ (without taking into account the precomputations of the coefficients).

2.3 Stochastic Binary Ordinal Search

The BOS model is inspired by a standard binary search with added noise in the comparison. Consequently, the algorithm may at times misidentify the next subset for the search, ultimately causing it to overlook the sought-after value.

2.3.1 Probabilistic model

The stochastic binary ordinal search unfolds as follows: Let m be the number of categories. Then, for at most $m-1$ steps, we perform the following three operations. We start with the full set of categories, denoted as $e_1 = [\![1,m]\!]$. Then we perform the following steps:

At step j, we start with a subset of all the categories, denoted as $e_j = [\![l_j, u_j - 1]\!] \subseteq [\![1,m]\!]$.

1. Sample a breakpoint y_j uniformly in e_j ($y_j \sim \mathcal{U}(e_j)$).
2. Draw an accuracy indicator z_j from a Bernoulli distribution with parameter π ($z_j \sim$ Bernoulli(π)). A value of $z_j = 1$ indicates that the comparison is perfect, and the next step will be computed optimally. A value of $z_j = 0$ implies a blind comparison at the next step.
3. Determine the new subset e_{j+1} for the next iteration. Firstly, split the subset into three intervals, namely $e_j^- = [\![l_j, y_j - 1]\!]$, $e_j^= = \{y_j\}$, and $e_j^+ = [\![y_j + 1, u_j - 1]\!]$. e_{j+1} will be chosen among these intervals. If the comparison is blind ($z_j = 0$), randomly select the interval with a probability proportional to its size. Alternatively, if $z_j = 1$ and the comparison is perfect, select the interval containing μ (or, by default, the one closest to it).

After $m-1$ steps, the resulting interval contains a single value, which is the observed result $e_m = \{x\}$ of the BOS model.

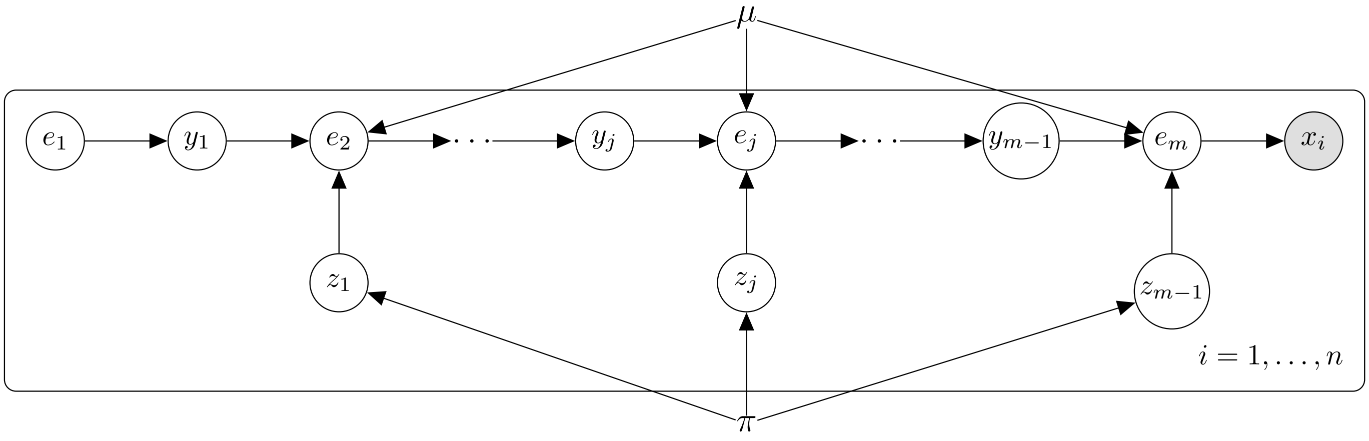

Figure 1: Graphical model of the stochastic Binary Ordinal Search.

2.4 BOS Model

Definition 1. *We define Correct(μ, e, y, f) as the indicator function that f is the correct subset to choose in case of a perfect comparison i.e. if $\mu \in f$ or by default the closest to μ.*

Definition 2. *We define Next(e, y) as the set of intervals that can be chosen after a comparison at breakpoint y in the interval e i.e. Next$(e, y) = \{e^{-,y}, e^{=,y}, e^{+,y}\}$ with $e = [\![l, u-1]\!]$ and $y \in [\![l, u-1]\!]$: $e^{-,y} = [\![l, y-1]\!]$, $e^{=,y} = \{y\}$ and $e^{+,y} = [\![y+1, u-1]\!]$.*

We now suppose

Lemma 1 (e_j transition)**.** *$\forall m \in \mathbb{N}^*, \forall x \in [\![1, m]\!], \forall \mu \in [\![1, m]\!], \pi \in [0,1], \forall e \subset [\![1, m]\!], \forall f \subset e$:*

$$\Pr(f|x \in e, e, \mu, \pi) = \frac{1}{|e|} \sum_{y \in e} \left[\left(\mathit{Correct}(\mu, e, y, f) - \frac{|f|}{|e|} \right) \pi + \frac{|f|}{|e|} \right] \mathbb{1}\left\{ x \in \mathit{Next}(e, y) \right\}$$

Proof. We have that, by marginalization over the breakpoint y:

$$\begin{aligned}
\Pr(f|x \in e, e, \mu, \pi) &= \sum_{y \in e} \Pr(f, y | x \in e, e, \mu, \pi) && (3)\\
&= \sum_{y \in e} \Pr(f|y, x \in e, e, \mu, \pi) \Pr(y|x \in e, e, \mu, \pi) && (4)\\
&= \sum_{y \in e} \Pr(f|y, x \in e, e, \mu, \pi) \frac{1}{|e|} && (5)\\
&= \frac{1}{|e|} \sum_{y \in e} \Pr(f|y, x \in e, e, \mu, \pi) && (6)
\end{aligned}$$

Then by marginalization over the accuracy indicator z:

$$\Pr(f|x \in e, e, \mu, \pi) = \frac{1}{|e|} \sum_{y \in e} \sum_{z \in \{0,1\}} \Pr(f|y, x \in e, e, \mu, \pi, z) \Pr(z|y, x \in e, e, \mu, \pi) \tag{7}$$

$$= \frac{1}{|e|} \sum_{y \in e} [\Pr(f|y, x \in e, e, \mu, \pi, z = 1)\pi + \Pr(f|y, x \in e, e, \mu, \pi, z = 0)(1 - \pi)] \tag{8}$$

$$= \frac{1}{|e|} \sum_{y \in e} \left[\mathrm{Correct}\,(\mu, e, y, f)\,\pi + \frac{|f|}{|e|}(1 - \pi) \right] \mathbb{1}\left\{x \in \mathrm{Next}\,(e, y)\right\} \tag{9}$$

$$= \frac{1}{|e|} \sum_{y \in e} \left[\left(\mathrm{Correct}\,(\mu, e, y, f) - \frac{|f|}{|e|} \right) \pi + \frac{|f|}{|e|} \right] \mathbb{1}\left\{x \in \mathrm{Next}\,(e, y)\right\} \tag{10}$$

Note that $\mathbb{1}\left\{x \in \mathrm{Next}\,(e, y)\right\} = 1$ only for one value of y and 0 for all the others. □

Lemma 2. *$\forall m \in \mathbb{N}^*, \forall x \in [\![1, m]\!], \forall \mu \in [\![1, m]\!], \pi \in [0, 1], \forall e \subset [\![1, m]\!]$:*

$$\pi \mapsto \Pr(x|x \in e, e, \mu, \pi)$$

is a polynomial function of degree at most $|e| - 1$.

Proof. Let $m \in \mathbb{N}^*, x \in [\![1, m]\!], \mu \in [\![1, m]\!], \pi \in [0, 1]$.

We proceed by strong induction on $|e|$.

- Initialization: $|e| = 1$:

  $$\Pr(x|x \in e, e, \mu, \pi) = \mathbb{1}\left\{e = \{x\}\right\}$$

  which is a polynomial function of degree 0.

- Induction: Suppose the result holds for all $f \subset [\![1, m]\!]$ of size less or equal than $|e| - 1$ and let us prove it for $|e|$.

  We marginalize over the next interval f and we have that:

  $$\Pr(x|x \in e, e, \mu, \pi) = \sum_{f \subset e} \Pr(x, f|x \in e, e, \mu, \pi) \tag{11}$$

  $$= \sum_{f \subset e} \Pr(x|f, x \in e, e, \mu, \pi) \Pr(f|x \in e, e, \mu, \pi) \tag{12}$$

  We can then notice that $\Pr(x|f, x \in e, e, \mu, \pi)$ is 0 if $x \notin f$ and that e does not intervene in the BOS process anymore. Hence we can replace $\Pr(x|f, x \in e, e, \mu, \pi)$ by $\Pr(x|x \in f, f, \mu, \pi)$ and sum only over $f \subset e$ such that $x \in f$:

  $$\Pr(x|x \in e, e, \mu, \pi) = \sum_{f \subset e; x \in f} \Pr(x|x \in f, f, \mu, \pi) \Pr(f|x \in e, e, \mu, \pi) \tag{13}$$

  As $\Pr(f|x \in e, e, \mu, \pi)$ is a polynomial function of degree at most 1 (see lemma 1) and $\Pr(x|f, x \in f, f, \mu, \pi)$ is a polynomial function of degree at most $|f| - 1 \leqslant |e| - 2$ by induction hypothesis, we have that $\Pr(x|x \in e, e, \mu, \pi)$ is a polynomial function of degree at most $|e| - 1$.

Hence the result holds for all e. □

Theorem 1 (Likelihood is polynomial). *$\forall m \in \mathbb{N}^*, \forall x \in [\![1, m]\!], \forall \mu \in [\![1, m]\!],$:*

$$\pi \mapsto \Pr(x|\mu, \pi)$$

is a polynomial function of degree at most $m - 1$.

Proof. Let $m \in \mathbb{N}^*$, $x \in [\![1, m]\!]$ and $\mu \in [\![1, m]\!]$.

First we can introduce redondant knowledge as we start necessarily with the full set of categories, we can add its value as known. We have that $\Pr(x|\mu, \pi) = \Pr(x|e_1, \mu, \pi)$. We also now that $x \in e_1$ therefore $\Pr(x|\mu, \pi) = \Pr(x|x \in e_1, e_1, \mu, \pi)$.

We can now use the previous lemma 2 to conclude that $\Pr(x|\mu, \pi)$ is a polynomial function of degree at most $m - 1$. □

We can now prove that $\forall x \in [\![0, h-1]\!], \forall \mu \in [\![0, h-1]\!], \pi \mapsto \Pr(x|x \in [\![0, h-1]\!], \mu, \pi)$ is concave on $[0, 1]$

Lemma 3 (Log concavity affine times polynomial). *For I a real interval.*

Let P be a strictly log*-concave function or without zeros on I and $a, b \in \mathbb{R}$ with $a \neq 0$ and $\forall t \in I, at + b \neq 0$. Then $f : t \mapsto (at + b)P(t)$ is* log*-concave on I.*

Moreover, if we just have that P is log*-concave on I without the assumption on a and b, we have that f is* log*-concave on I.*

Proof. Let $t \in I$.

We have that:

$$f'(t) = aP(t) + (at + b)P'(t) \tag{14}$$
$$f''(t) = 2aP'(t) + (at + b)P''(t) \tag{15}$$
$$f'(t)^2 = a^2P(t)^2 + 2a(at + b)P(t)P'(t) + (at + b)^2P'(t)^2 \tag{16}$$
$$f(t)f''(t) = 2a(at + b)P(t)P'(t) + (at + b)^2P(t)P''(t) \tag{17}$$

Hence:

$$f'(t)^2 - f(t)f''(t) = a^2P(t)^2 + (at + b)^2 \left[P'(t)^2 - P(t)P''(t)\right]$$

If P is strictly log-concave on I, using the lemma 9 we have that $P'(t)^2 - P(t)P''(t) > 0$. If P is without zeros on I, we have that $\forall t \in I, P(t) > 0$.

As all the terms are $\geqslant 0$ and one of the two is strictly positive, hence we have that $\forall t \in I, f'(t)^2 - f(t)f''(t) > 0$ and using the lemma 9 we have that f is strictly log-concave on I.

If we just have that P is log-concave on I without the assumption on a and b, we have that $\forall t \in I, f'(t)^2 - f(t)f''(t) \geqslant 0$ and using the lemma 9 we have that f is log-concave on I. □

Lemma 4 (Log concavity of the BOS model). *$\forall m \in \mathbb{N}^*, \forall x \in [\![1, m]\!], \forall \mu \in [\![1, m]\!], , \forall e \subset [\![1, m]\!]$:*

$$\pi \mapsto \Pr(x|x \in e, e, \mu, \pi)$$

If $|e| > 2$, this function is log*-concave on $[0, 1]$ and strictly* log*-concave on $]0, 1[$. If $|e| \leqslant 2$ this function is affine on $[0, 1]$ if $|e| \leqslant 2$.*

Proof. Let $m \in \mathbb{N}^*$, $x \in [\![1, m]\!]$ and $\mu \in [\![1, m]\!]$. We proceed by induction on $|e|$.

- Initialization: $|e| \leqslant 2$:

$$\pi \mapsto \Pr(x|x \in e, e, \mu, \pi)$$

  is a polynomial function of degree at most 1 by lemma 2 and is therefore affine on $[0, 1]$.

- Induction: Suppose that $|e| > 2$ and that the result holds for all $f \subset [\![1, m]\!]$ of size less or equal than $|e| - 1$ and let us prove it for $|e|$.

  Using the lemma 2, we have:

$$\Pr(x|x \in e, e, \mu, \pi) = \sum_{f \subset e; x \in f} \Pr(x|x \in f, f, \mu, \pi) \Pr(f|x \in e, e, \mu, \pi) \quad (18)$$

  We have a sum of function. We now have to check that each function is log-concave on $[0, 1]$ and at least one is strictly log-concave on $]0, 1[$. We will use the lemma 3. We first focus on $\Pr(f|x \in e, e, \mu, \pi)$.

  Using the lemma 1, we have that $\Pr(f|x \in e, e, \mu, \pi)$ is an affine function of π times of the form $a_f \pi + b_f$ where $a_f = 1 - \frac{|f|}{|e|} \neq 0$ (as $|f| < |e|$) or $a_f = -\frac{|f|}{|e|} \neq 0$ and $b_f = \frac{|f|}{|e|}$.

  Let $t \in]0, 1[$. We check that $a_f t + b_f \neq 0$. If $a_f = 1 - \frac{|f|}{|e|}$, we have that $a_f > 0, t > 0$ and $b_f > 0$ and therefore $a_f t + b_f > 0$. If $a_f = -\frac{|f|}{|e|}$, we have that $a_f t + b_f = \frac{|f|}{|e|}(1 - t) > 0$.

  Each $\pi \mapsto \Pr(x|x \in f, f, \mu, \pi)$ is log-concave on $[0, 1]$ by induction hypothesis. Using the lemma 3, we have that each $\pi \mapsto \Pr(x|x \in f, f, \mu, \pi) \Pr(f|x \in e, e, \mu, \pi)$ is log-concave on $[0, 1]$.

  We now need to check that at least one is strictly log-concave on $]0, 1[$. Using the lemma 3, we have that to prove that at least one of the $\pi \mapsto \Pr(x|x \in f, f, \mu, \pi)$ has no zeros on $]0, 1[$. As by induction hypothesis, each $\pi \mapsto \Pr(x|x \in f, f, \mu, \pi)$ is log-concave on $[0, 1]$ it has no zeros on $]0, 1[$ or is constant equal to 0 on $[0, 1]$. As $\Pr(x|x \in e, e, \mu, \pi) > 0$ we have that at least one of the $\pi \mapsto \Pr(x|x \in f, f, \mu, \pi)$ is not constant equal to 0 on $[0, 1]$ and therefore has no zeros on $]0, 1[$. Hence at least one of the $\pi \mapsto \Pr(x|x \in f, f, \mu, \pi) \Pr(f|x \in e, e, \mu, \pi)$ is strictly log-concave on $]0, 1[$.

  We can conclude that the sum is strictly log-concave on $]0, 1[$.

  TR: Need to check why this does not work for |e| = 2

□

Theorem 2 (Log concavity of the BOS model). *$\forall m \in \mathbb{N}^*, \forall x \in [\![1, m]\!], \forall \mu \in [\![1, m]\!]$:*

$$\pi \mapsto \Pr(x|x \in [\![1, m]\!], \mu, \pi)$$

is log-*concave on* $[0, 1]$.

Proof. We just have to apply the lemma 4 to the case where $e = [\![1, m]\!]$. □

Lemma 5 (Symetries the likelihood). $\forall m \in \mathbb{N}^*, \forall x \in [\![1, m]\!], \forall \mu \in [\![1, m]\!], \pi \in [0, 1], \forall e = [\![l, u]\!] \subset [\![1, m]\!]$*:*

$$\Pr(x|x \in [\![l, u]\!], e = [\![l, u]\!], \mu, \pi) = \Pr(x - l|x - l \in [\![0, u - l]\!], e = [\![0, u - l]\!], \max(0, \mu - l), \pi)$$

Proof.

TR: To be done

□

Definition 3. *As justified by the lemma 5, we can define the following notation:*

$$l\left(x, \mu, h\right) := \Pr(x|x \in [\![0, h - 1]\!], e = [\![0, h - 1]\!], \mu, \pi)$$

Theorem 3 (Computing the likelihood)**.** *$\forall m \in \mathbb{N}^*, \forall x \in [\![1, m]\!], \forall \mu \in [\![1, m]\!], \forall \pi \in [0, 1]$:*

$$\begin{aligned} l(x, \mu, h) = & \frac{1}{h} \sum_{y=0}^{x-1} l(x, \mu, y) \left[\left(\mathbb{1}\{\mu < y\} - \frac{y}{h} \right) \pi + \frac{y}{h} \right] \\ & + \frac{1}{h} \left[\left(\mathbb{1}\{\mu = x \vee (x = 0 \wedge \mu \leqslant x) \vee (x = h-1 \wedge \mu \geqslant x)\} - 1 \right) \pi + \frac{1}{h} \right] \\ & + \frac{1}{h} \sum_{y=x+1}^{h-1} l(x-y, \max(0, \mu - y), h - y) \left[\left(\mathbb{1}\{\mu > y\} - \frac{h-y-1}{h} \right) \pi + \frac{h-y-1}{h} \right] \end{aligned} \tag{19}$$

Proof. First we marginalize over the breakpoint y:

$$l(x, \mu, h) = \Pr(x | x \in [\![0, h-1]\!], e = [\![0, h-1]\!], \mu, \pi) \tag{20}$$

$$\begin{aligned} &= \sum_{y=0}^{h-1} \Pr(x, f = e^{-,y} | x \in [\![0, h-1]\!], e = [\![0, h-1]\!], \mu, \pi) \\ &+ \sum_{y=0}^{h-1} \Pr(x, f = e^{=,y} | x \in [\![0, h-1]\!], e = [\![0, h-1]\!], \mu, \pi) \\ &+ \sum_{y=0}^{h-1} \Pr(x, f = e^{+,y} | x \in [\![0, h-1]\!], e = [\![0, h-1]\!], \mu, \pi) \end{aligned} \tag{21}$$

$$\tag{22}$$

Then we use the Bayes rule ($P(A, B) = P(A|B)P(B)$) to get likelihoods of x:

$$\begin{aligned} &= \sum_{y=0}^{h-1} \Pr(x | x \in [\![0, h-1]\!], f = e^{-,y}, \mu, \pi) \Pr(e^{-,y} | x \in [\![0, h-1]\!], e = [\![0, h-1]\!], \mu, \pi) \\ &+ \sum_{y=0}^{h-1} \Pr(x | x \in [\![0, h-1]\!], f = e^{=,y}, \mu, \pi) \Pr(e^{=,y} | x \in [\![0, h-1]\!], e = [\![0, h-1]\!], \mu, \pi) \\ &+ \sum_{y=0}^{h-1} \Pr(x | x \in [\![0, h-1]\!], f = e^{+,y}, \mu, \pi) \Pr(e^{+,y} | x \in [\![0, h-1]\!], e = [\![0, h-1]\!], \mu, \pi) \end{aligned} \tag{23}$$

$$\tag{24}$$

Then we can get rid of the case where $x \notin f$ as it is 0:

$$\begin{aligned} &= \sum_{y=0}^{x-1} \Pr(x | x \in [\![0, y-1]\!], f = [\![0, y-1]\!], \mu, \pi) \Pr([\![0, y-1]\!] | x \in [\![0, h-1]\!], e = [\![0, h-1]\!], \mu, \pi) \\ &+ \Pr(x | x \in \{x\}, f = \{x\}, \mu, \pi) \Pr(\{x\} | x \in [\![0, h-1]\!], e = [\![0, h-1]\!], \mu, \pi) \\ &+ \sum_{y=x+1}^{h-1} \Pr(x | x \in [\![y+1, h-1]\!], f = [\![y+1, h-1]\!], \mu, \pi) \\ &\quad \Pr([\![y+1, h-1]\!] | x \in [\![0, h-1]\!], e = [\![0, h-1]\!], \mu, \pi) \end{aligned} \tag{25}$$

$$\tag{26}$$

We can apply the lemma 5 to the third term:

$$\begin{aligned}
&= \sum_{y=0}^{x-1} \Pr(x | x \in \llbracket 0, y-1 \rrbracket, f = \llbracket 0, y-1 \rrbracket, \mu, \pi) \Pr(\llbracket 0, y-1 \rrbracket | x \in \llbracket 0, h-1 \rrbracket, e = \llbracket 0, h-1 \rrbracket, \mu, \pi) \\
&+ \qquad \Pr(x | x \in \{x\}, f = \{x\}, \mu, \pi) \Pr(\{x\} | x \in \llbracket 0, h-1 \rrbracket, e = \llbracket 0, h-1 \rrbracket, \mu, \pi) \\
&+ \sum_{y=x+1}^{h-1} \Pr(x-y-1 | x-y-1 \in \llbracket 0, h-1-y-1 \rrbracket, f = \llbracket 0, h-1-y-1 \rrbracket, \max(0, \mu-y-1), \pi) \\
&\quad \Pr(\llbracket y+1, h-1 \rrbracket | x \in \llbracket 0, h-1 \rrbracket, e = \llbracket 0, h-1 \rrbracket, \mu, \pi)
\end{aligned} \tag{27}$$

$$\begin{aligned}
&= \sum_{y=0}^{x-1} l(x, \mu, y-1) \Pr(\llbracket 0, y-1 \rrbracket | x \in \llbracket 0, h-1 \rrbracket, e = \llbracket 0, h-1 \rrbracket, \mu, \pi) \\
&+ \qquad l(x, \mu, 1) \Pr(\{x\} | x \in \llbracket 0, h-1 \rrbracket, e = \llbracket 0, h-1 \rrbracket, \mu, \pi) \\
&+ \sum_{y=x+1}^{h-1} l(x-y-1, \max(0, \mu-y-1), h-y-1) \Pr(\llbracket y+1, h-1 \rrbracket | x \in \llbracket 0, h-1 \rrbracket, e = \llbracket 0, h-1 \rrbracket, \mu, \pi)
\end{aligned} \tag{28}$$

$$\tag{29}$$

First we have $l(x, \mu, 1) = 1$. Moreover, we can now use the lemma 1 to get replace the transition probabilities. In our case as we already selected the only possible interval for each breakpoint we have the only term of the sum where $\mathbb{1}\{x \in \text{Next}(e, y)\} = 1$ and the sum is reduced to a single term:

$$\begin{aligned}
&= \sum_{y=0}^{x-1} l(x, \mu, y-1) \frac{1}{h} \left[\left(\text{Correct}(\mu, \llbracket 0, h-1 \rrbracket, y-1, \llbracket 0, y-1 \rrbracket) - \frac{y}{h} \right) \pi + \frac{y}{h} \right] \\
&+ \qquad \frac{1}{h} \left[\left(\text{Correct}(\mu, \llbracket 0, h-1 \rrbracket, x, \{x\}) - \frac{1}{h} \right) \pi + \frac{1}{h} \right] \\
&+ \sum_{y=x+1}^{h-1} l(x-y, \max(0, \mu-y), h-y) \frac{1}{h} \\
&\quad \left[\left(\text{Correct}(\mu-y-1, \llbracket 0, h-1-y-1 \rrbracket, h-y-1, \llbracket y, h-1 \rrbracket) - \frac{h-y-1}{h} \right) \pi + \frac{h-y-1}{h} \right]
\end{aligned} \tag{30}$$

$$\tag{31}$$

We can then replace $\text{Correct}(\mu, \llbracket 0, h-1 \rrbracket, \bullet, \bullet)$ by a logical expression. (We must take into account the special case of the first and last breakpoint):

$$
\begin{aligned}
&= \frac{1}{h}\sum_{y=0}^{x-1} l\left(x,\mu,y\right)\left[\left(\mathbb{1}\left\{\mu<y\right\}-\frac{y}{h}\right)\pi+\frac{y}{h}\right] \\
&+ \frac{1}{h} \qquad \left[\left(\mathbb{1}\left\{\mu=x\vee(x=0\wedge\mu\leqslant x)\vee(x=h-1\wedge\mu\geqslant x)\right\}-1\right)\pi+\frac{1}{h}\right] \qquad (32)\\
&+ \frac{1}{h}\sum_{y=x+1}^{h-1} l\left(x-y,\max(0,\mu-y),h-y\right)\left[\left(\mathbb{1}\left\{\mu>y\right\}-\frac{h-y-1}{h}\right)\pi+\frac{h-y-1}{h}\right] \qquad (33)
\end{aligned}
$$

□

2.5 Globally Ordered Data model

The authors of Biernacki and Jacques [2016] motivated the use of binary search as such: "In order to minimize the number of potentially wrong comparisons, it is necessary to minimize the number of comparisons performed during the search process.". However, we believe that minimizing the number of incorrect comparisons may not be an adequate intuition, and it is more crucial to minimize the probability of making a wrong guess. Motivated by this perspective, we have developed an alternative model where the data is compared with each category with some noise. We will refer to this model as the Globally Ordered Data (GOD) model.

We still consider a search for the parameter μ among the ordered categories. However, instead of conducting a binary search, we compare each category to the parameter μ and with probability π we get the correct answer. After making all these comparisons, we then select the category that corresponds to the minimum number of comparison error. This approach display similar properties to the BOS model such as a unique mode, probability distribution decrease on either side of the mode, and the flexibility to accommodate uniform or Dirac distribution.

2.5.1 Probabilistic model

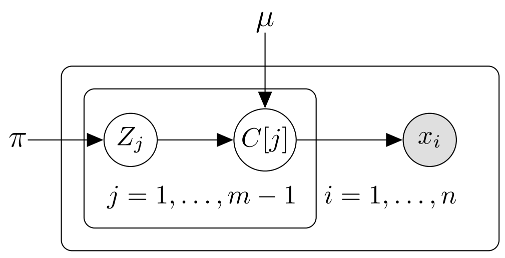

Figure 2: Graphical model of the GOD model.

The GOD model for m categories is characterized by two parameters $\mu \in [\![1, m]\!], \pi \in]\frac{1}{2}, 1]$. The observed data is only the selected category x. The latent variables are the vector $Z = (Z_1, ..., Z_{m-1}) \in \{0,1\}^{m-1}$ and $C \in \{0,1\}^{m-1}$. $(Z_j)_{j\in[\![1,m-1]\!]}$ is a vector of independent Bernoulli variables, with parameter π. For $j \in [\![1, m-1]\!]$, Z_j indicates whether the comparison with the category j is correct ($Z_j = 1$) or not ($Z_j = 0$). C is the vector containing the m results of the comparisons depending on both Z and the parameter μ. It is defined as follows:

$$\forall j \in [\![1, m-1]\!], C[j] = \begin{cases} (\mu < j) & \text{if } Z_j = 1 \\ (\mu \not< j) & \text{if } Z_j = 0 \end{cases}$$

The GOD model will generate $x \in [\![1, m]\!]$ such that $x \sim \mathcal{U}(\mathrm{argmin}_{k \in [\![1,m]\!]} ||c - E_k||_1)$. We can interpret this as a probability maximization as stated in Theorem 4. The graphical model associated with this probabilistic model is depicted on Figure 2.

Definition 4 (Heaviside vector)**.** *For $k \in [\![1, m]\!]$, we define:*

$$E_k := (1)^{k-1}(0)^{m-k} = (\underbrace{1, \dots, 1}_{k-1}, \underbrace{0, \dots, 0}_{m-k}).$$

Theorem 4. *If we suppose that the prior distribution of μ is uniform over $[\![1, m]\!]$ and $\pi > \frac{1}{2}$, then $\forall c \in \{0, 1\}^{m-1}$,*

$$\underset{k \in [\![1,m]\!]}{\mathrm{argmax}}\, \Pr(\mu = k | C = c) = \underset{k \in [\![1,m]\!]}{\mathrm{argmin}}\, ||c - E_k||_1 .$$

The proof can be found in appendix 7.

2.5.2 Parameter estimation

We want to estimate π and μ from a sample $(1, \dots, m)$ with weights $W \in \mathbb{R}_+^m$ generated by the GOD model. We aim at maximizing the likelihood of the sample : $\Pr(W|\pi, \mu)$. We proceed as explained in section 2.2.2.

Likelihood evaluation

Definition 5. *We define for $x \in [\![1, m]\!]$,:*

$$\mathcal{C}_x := \left\{ c \in \{0, 1\}^{m-1} | x \in \underset{k \in [\![1,m]\!]}{\mathrm{argmin}}\, ||c - E_k||_1 \right\}$$

Definition 6. *We define for $x \in [\![1, m]\!], \mu \in [\![1, m]\!], d \in [\![0, m-1]\!]$:*

$$u(\mu, x, d) := \sum_{c \in \mathcal{C}_x / ||c - E_\mu||_1 = d} \left| \underset{k \in [\![1,m]\!]}{\mathrm{argmin}}\, ||c - E_k||_1 \right|^{-1}$$

Using these definitions we have the expression of the likelihood of a single observation.

Theorem 5 (Observation likelihood)**.**

$$\Pr(x|\pi, \mu) = \pi^{m-1} \sum_{d=0}^{m-1} \left(\frac{1 - \pi}{\pi} \right)^d u(x, \mu, d)$$

Proof. See appendix 8. □

We can notice that once u is computed, the likelihood of a single observation can be computed in $\Theta(m)$ operations and the weighted likelihood can be computed in $\Theta(m^2)$ operations.

Computing the coefficients u The computation of the coefficients u can be done in $\mathcal{O}(m^2 2^m)$ time. We believe that it might be possible to compute it in polynomial time, with more research. Although still costly, it only needs to be computed once for a given m and can be stored in $\mathcal{O}(m^3)$ space.

TR: TODO: explain how we compute the coefficients

Log-concavity

Theorem 6. $\forall \mu \in [\![1, m]\!], \forall x \in [\![1, m]\!]$,

$$\pi \mapsto \Pr(x|\pi, \mu)$$

is stricly log-*concave on* $]\frac{1}{2}, 1[$ *and* log-*concave on* $[\frac{1}{2}, 1]$.

Proof. See appendix 9. □

As explained in section 2.2.2, we directly have that $\forall \mu \in [\![1, m]\!], \pi \mapsto L_W(\pi, \mu)$ is concave. Hence we can use a ternary search algorithm to estimate π for a given μ.

Conclusion As presented in section 2.2.2, we can estimate μ, π in $\mathcal{O}(m^3 \log \frac{1}{\varepsilon})$ operations once the coefficients u are computed. The coefficients u can be computed in $\mathcal{O}(m^2 2^m)$ operations and stored in $\mathcal{O}(m^3)$ space. This may seem costly, but m is usually small. (In comparison in Biernacki and Jacques [2016] they assume that $m \leqslant 7$)

3 Experiments

In this section, we try to evaluate the performance of the two models on different datasets. We first present the experimental setup and the datasets used for the experiments. We then present the results obtained for the estimation algorithms on synthetic data and on real-life datasets. We finally discuss the results obtained and the relevance of the models.

The goal of these experiments is to compare the two models but also to individually test their ability to cluster ordinal datasets and to check whether they are able to generalize to real-life datasets.

3.1 Experimental setup

Parameters of the experiment In this section, we propose to test the BOS model on synthetic data and on real-life datasets. To do so, we also use simple clustering algorithms to compare the performance of the BOS model on data that is adapted (ordinal) with algorithms that are not designed for this kind of data such as K-Means [MacQueen et al., 1967] and Gaussian Mixture Models [Reynolds et al., 2009].

We also test the AECM algorithm for the BOS model with both a random initialization of the parameters and an initialization of the parameters using the K-Means algorithm.

Runtimes are also measured on the same machine for all the algorithms to compare their efficiency.

Dataset. One of the main goal of the experiments is to test the ability of the models to generalize to real-life datasets. We therefore propose to test the illustrated methods on real world datasets to check the usefulness of the models on different real-life situations. Since the algorithm is specifically designed for ordinal observations, the datasets need to be adapted for the task. One way to apply to obtain real-life datasets is to quantize continuous datasets of observations that can be categorized (e.g. movies, store products, species...) [Skubacz and Hollmén, 2000]. Another interesting approach could be to test the models on tasks that they were not specifically designed for. This could allow seeing how they can generalize and whether they are applicable to a broader class of problems. We therefore propose to test the ability to cluster observations of binary features into different animal species.

Zoo Dataset. The zoo dataset consists of multiple features describing 101 different animals, with most of them being binary variables associated to a characteristic of the animal (hair, feathers, eggs, milk, . . .) [Forsyth, 1990]. Every animal belongs to one of 6 classes.

Car Evaluation Dataset. The car evaluation dataset consists of multiple features describing 1728 different cars, with most of them being ordinal variables associated to a characteristic of the car (buying price, maintenance price, number of doors, . . .) [Bohanec, 1997]. Every car belongs to one of 4 classes.

Hayes-Roth Dataset. The Hayes-Roth dataset consists of multiple features describing 132 different persons, with most of them being binary variables associated to a characteristic of the person (has a PhD, is married, is a professional, . . .) [Hayes-Roth and Hayes-Roth, 1989]. Every person belongs to one of 3 classes.

Caesarian Dataset. The Caesarian dataset is a dataset describing 80 different patients with multiple features associated to the patient (age, delivery number, delivery time, blood pressure, . . .) [Amin and Ali, 2018]. Every patient belongs to one of 2 classes.

The advantage of these datasets is that they are small enough to be able to compute the exact likelihood of the data given the model and the parameters. This allows to check whether the models are able to correctly fit the data.

Evaluation method. For most of the evaluation datasets, we will use classification tasks to check the ability to cluster with respect to pre-existing classes. This allows the evaluation framework to be easier to define and to compare different methods more easily.

In order to correctly associate the predicted clusters with the true clusters, we need to define a strategy that matches each predicted clusters with a true cluster number which will minimize a given criterion. In order to do so, we propose two methods:

- The first one and consists in sorting the histograms of the predicted clusters and the true clusters and then matching the two sorted lists by assigning the predicted clusters to the true cluster in the same sorted order.

  This method is naive because it does not take into account the distribution of the real clusters according to the true labels for the matching.

- The second method consists in solving the Assignment Problem [Kuhn, 1955] with the cost matrix being the distance between the histograms of the predicted clusters and the true clusters. This method takes into account the distribution of the real clusters according to the true labels for the matching. We can easily solve it using any Optimal Transport algorithm (or by defining the Linear Programming problem and solving it using an LP solver).

Figure 3 in appendix B shows that the optimal matching when considering the assignment matrix is a better choice in the case of the Zoo dataset for example and that the classes in the predicted distribution are assigned to the correct true class with respect to their proportions.

The evaluation metrics used to compare the different models are the F1-score, and the Accuracy score in the cases where the datasets are suited for classification and the Wasserstein distance and the Adjusted Rand Index (ARI).

- The F1-score is the harmonic mean of the precision and the recall for classification problems.

- The Wasserstein distance is a measure of the distance between two probability distributions [Ramdas et al., 2017]. It measures the cost of transforming one distribution into the other using the optimal transport plan which in this case is the matching obtained as described above.

$$W(\hat{y}, y) = \min_{\gamma \in \Gamma(\hat{y}, y)} \sum_{i,j} \gamma_{i,j} \left|\left| i - j \right|\right|, \tag{34}$$

  where $\Gamma(\hat{y}, y)$ is the set of all possible matchings between the predicted clusters and the true clusters and $\gamma_{i,j}$ is the probability of matching the predicted cluster i with the true cluster j (i.e. it is the proportion of the samples in the predicted cluster i that are in the true cluster j) for the matching.

- The ARI is a measure of the similarity between two clusterings of the same dataset. It is a function that outputs a value between -0.5 and 1, where 1 means that the two clusterings are identical, 0 means that the two clusterings are independent (random) and -0.5 means that the two clusterings are as different as possible. The ARI is symmetric and therefore does not take into account the order of the clusters [Steinley, 2004].

$$\mathrm{ARI}(\hat{y}, y) = \frac{\sum_{i,j} \binom{n_{i,j}}{2} - \left[\sum_i \binom{\hat{n}_i}{2} \sum_j \binom{n_j}{2}\right] / \binom{n}{2}}{\frac{1}{2}\left[\sum_i \binom{\hat{n}_i}{2} + \sum_j \binom{n_j}{2}\right] - \left[\sum_i \binom{\hat{n}_i}{2} \sum_j \binom{n_j}{2}\right] / \binom{n}{2}}, \tag{35}$$

  where $n_{i,j}$ is the number of samples that are in the predicted cluster i and in the true cluster j, $\hat{n}_i$ is the number of samples in the predicted cluster i and n_j is the number of samples in the true cluster j.

3.2 Estimation methods

In this section, we present the results obtained after running the AECM estimation algorithms for different parameters used to generate data from the BOS distribution and from the GOD model in order to compare the different estimators for their respective distributions. We will then also try the different models on real-life datasets in order to check how they perform for clustering on real data.

BOS distribution. We first test the AECM algorithm on the BOS distribution with experiments similar to Biernacki and Jacques [2016] in order to confirm that our implementation is coherent with the algorithm proposed and yields results that are close. We generate data from the BOS distribution with different parameters and then run the AECM algorithm on the generated data to estimate the parameters. We then compare the estimated parameters with the true parameters using the L_1 distance between the two vectors. We repeat this process multiple times for different parameters and average the results to obtain the results presented in Table 1 in Appendix A.

GOD model. We then test the AECM algorithm on the GOD model with experiments similar to Biernacki and Jacques [2016] in order to confirm that the estimation algorithms and the estimators proposed are able to estimate the parameters generated from a GOD distribution. We use data generated from the GOD model with similar parameters as for the BOS distribution. The results are presented in Table 2 in the appendix section.

3.3 Experiments with real-life datasets

In this section, we present the results obtained after running the AECM algorithm on the real-life datasets presented above. We then compare the results obtained with the BOS model with the results obtained with the GOD model and with the results obtained with the K-Means algorithm and the Gaussian Mixture Models. We also compare the results obtained with the two different methods to match the predicted clusters with the true clusters. The results are presented in Table 3 in the appendix B.

We notice that although K-Means allows to significantly reduce the runtime of both the BOS and the GOD models estimations, it does not necessarily increase the clustering score and the classification score. The BOS model, because of its complexity, is also the longest to run but seems to be competitive with the other models on most datasets.

In order to get a better idea of the differences between the clustering methods, we also plot t-SNE visualizations [Van der Maaten and Hinton, 2008] for different datasets and the multiple models. Figures 4 in appendix C shows the plots of all the features and the associated true labels and clusters.

Histograms and assignment matrices of some datasets are provided in appendix E and appendix D in order to get a better understanding of the different assignments obtained in these settings.

4 Conclusion

In this study, we analyzed model-based clustering for ordinal data, with a specific focus on the Binary Ordinal Search (BOS) and a proposed alternative we called Globally Ordinal Distribution (GOD) models. We aimed to understand and evaluate their efficiency in clustering and classifying ordinal data compared to more traditional methods like K-Means and Gaussian Mixture Models. Our exploration spanned both synthetic and real-world datasets, providing a comprehensive view of the models' performance in various scenarios.

The experiments on synthetic data confirmed the theoretical foundations of the BOS and GOD models. When parameters were known, both models showed an ability to recover the underlying structure of the generated data. Particularly, the BOS model, despite its computational intensity due to its design, performed well in clustering tasks, highlighting its potential for applications with ordinal data. The GOD model, with its more manageable computational requirements, also demonstrated promising results, making it a practical alternative for larger datasets.

When applied to real-world datasets, the results were more nuanced. While both BOS and GOD models performed competitively in certain scenarios, they did not universally outperform the traditional methods. This suggests that while specialized ordinal models are interesting, especially in scenarios where the ordinal nature of data is pronounced, they are not a default solution. It is essential to consider the specific characteristics of the dataset and the computational resources available when choosing the appropriate clustering method.

Different visualizations also provide further insights into how the models partition the data space. They revealed that while the clusters identified by the BOS and GOD models often made intuitive sense, they sometimes differ significantly from those identified by K-Means and Gaussian Mixture Models. This highlights the different assumptions and approaches these models take when learning the structure within data.

In conclusion, the study reaffirms the potential of model-based clustering for ordinal data, particularly highlighting the BOS and GOD models as valuable tools. However, it also demonstrates that the choice of model should be informed by both the nature of the data and the practical constraints of the problem at hand. Further research could explore further refinements to these models, more extensive comparisons with other methods, and applications to a broader range of real-world scenarios.

References

Muhammad Amin and Amir Ali. Caesarian Section Classification Dataset. UCI Machine Learning Repository, 2018. DOI: https://doi.org/10.24432/C5N59X.

Christophe Biernacki and Julien Jacques. Model-based clustering of multivariate ordinal data relying on a stochastic binary search algorithm. *Statistics and Computing*, 26:929–943, 2016.

Marko Bohanec. Car Evaluation. UCI Machine Learning Repository, 1997. DOI: https://doi.org/10.24432/C5JP48.

Richard Forsyth. Zoo. UCI Machine Learning Repository, 1990. DOI: https://doi.org/10.24432/C5R59V.

Barbara Hayes-Roth and Frederick Hayes-Roth. Hayes-Roth. UCI Machine Learning Repository, 1989. DOI: https://doi.org/10.24432/C5501T.

Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

James MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA, 1967.

Xiao-Li Meng and David Van Dyk. The em algorithm—an old folk-song sung to a fast new tune. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 59(3):511–567, 1997.

Aaditya Ramdas, Nicolás García Trillos, and Marco Cuturi. On wasserstein two-sample testing and related families of nonparametric tests. *Entropy*, 19(2):47, 2017.

Douglas A Reynolds et al. Gaussian mixture models. *Encyclopedia of biometrics*, 741(659-663), 2009.

Michal Skubacz and Jaakko Hollmén. Quantization of continuous input variables for binary classification. volume 1983, pages 42–47, 01 2000. ISBN 978-3-540-41450-6. doi: 10.1007/3-540-44491-2_7.

Douglas Steinley. Properties of the hubert-arable adjusted rand index. *Psychological methods*, 9(3): 386, 2004.

Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

A AECM tests for BOS and GOD distributions

Init.	n	$n_{clusters}$	d	n_{cats}	$\Delta\alpha$	$\Delta\mu$	$\Delta\pi$
K-Means	50	3	3	2	0.143	0.167	0.295
				3	0.135	0.667	0.116
			5	2	0.090	0.200	0.172
				3	0.083	0.333	0.105
		5	3	2	0.099	0.667	0.337
				3	0.142	0.778	0.356
			5	2	0.099	0.400	0.210
				3	0.125	0.400	0.157
	250	3	3	2	0.111	0.167	0.344
				3	0.021	0.444	0.206
			5	2	0.152	0.200	0.162
				3	0.043	0.267	0.106
		5	3	2	0.149	0.667	0.423
				3	0.315	0.444	0.246
			5	2	0.134	0.400	0.290
				3	0.174	0.467	0.129
Random	50	3	3	2	0.112	0.167	0.074
				3	0.085	0.222	0.025
			5	2	0.011	0.000	0.043
				3	0.039	0.067	0.038
		5	3	2	0.058	0.167	0.081
				3	0.073	0.111	0.079
			5	2	0.095	0.300	0.130
				3	0.068	0.067	0.117
	250	3	3	2	0.095	0.000	0.035
				3	0.018	0.222	0.007
			5	2	0.031	0.000	0.016
				3	0.017	0.067	0.019
		5	3	2	0.037	0.167	0.022
				3	0.057	0.222	0.037
			5	2	0.044	0.000	0.022
				3	0.020	0.067	0.052

Table 1: Results of the experiments for the AECM algorithm no synthetic data with the BOS distribution. The parameters are the number of samples n, the number of clusters $n_{clusters}$, the dimension d and the number of categories n_{cats}. The deltas are the average of the L_1 distances between the true and estimated parameters after applying optimal transport to find the correct clusters.

Init.	n	$n_{clusters}$	d	n_{cats}	$\Delta\alpha$	$\Delta\mu$	$\Delta\pi$
K-Means	50	3	3	2	0.103	0.167	0.166
				3	0.132	0.444	0.095
			5	2	0.063	0.300	0.088
				3	0.074	0.067	0.020
		5	3	2	0.082	0.333	0.207
				3	0.149	0.778	0.118
			5	2	0.102	0.300	0.122
				3	0.194	0.400	0.089
	250	3	3	2	0.093	0.167	0.152
				3	0.079	0.222	0.076
			5	2	0.088	0.200	0.071
				3	0.121	0.267	0.073
		5	3	2	0.083	0.833	0.211
				3	0.104	0.889	0.125
			5	2	0.097	0.300	0.110
				3	0.113	0.400	0.064
Random	50	3	3	2	0.063	0.167	0.104
				3	0.113	0.333	0.084
			5	2	0.050	0.100	0.057
				3	0.075	0.267	0.053
		5	3	2	0.058	0.333	0.145
				3	0.177	0.667	0.092
			5	2	0.083	0.400	0.112
				3	0.153	0.467	0.056
	250	3	3	2	0.032	0.167	0.093
				3	0.197	0.222	0.042
			5	2	0.014	0.200	0.050
				3	0.072	0.133	0.017
		5	3	2	0.052	0.167	0.118
				3	0.098	0.889	0.146
			5	2	0.058	0.400	0.085
				3	0.164	0.400	0.046

Table 2: Results of the experiments for the AECM algorithm no synthetic data with the GOD model. The parameters are the number of samples n, the number of clusters $n_{clusters}$, the dimension d and the number of categories n_{cats}. The deltas are the average of the L_1 distances between the true and estimated parameters after applying optimal transport to find the correct clusters.

B Metrics on real-world datasets

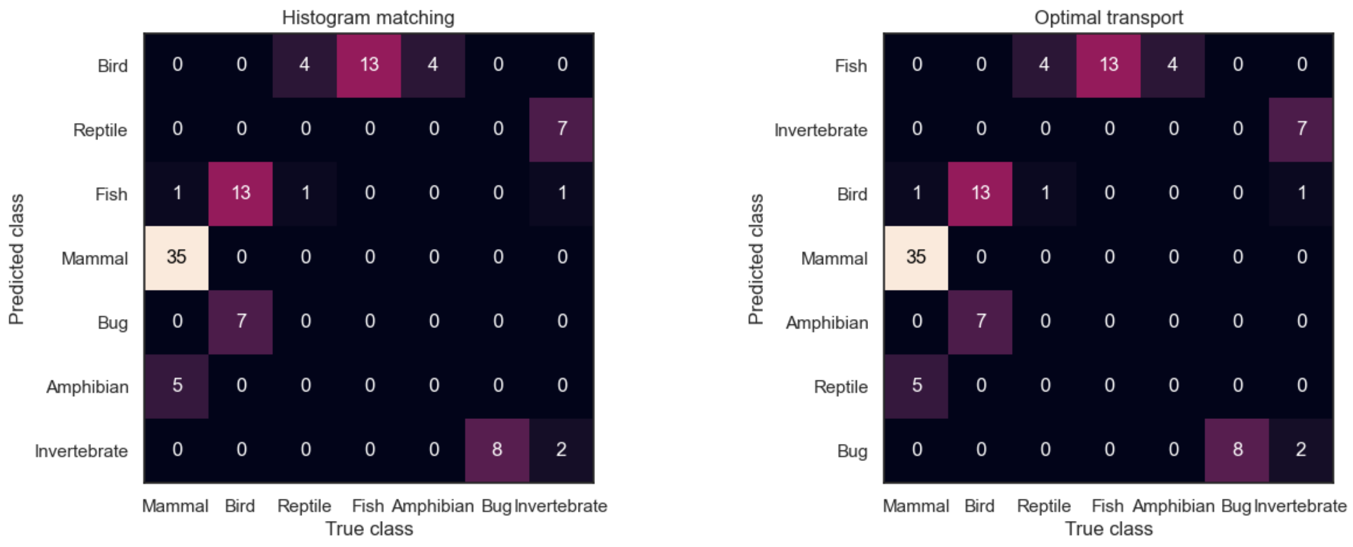

Figure 3: Illustration of the two assignment matrices from the different methods after clustering the Zoo dataset. On the left, the naive method and on the right, the optimal assignment method. The numbers in the matrices represent the number of samples in the predicted cluster i that are in the true cluster j.

C t-SNE plots

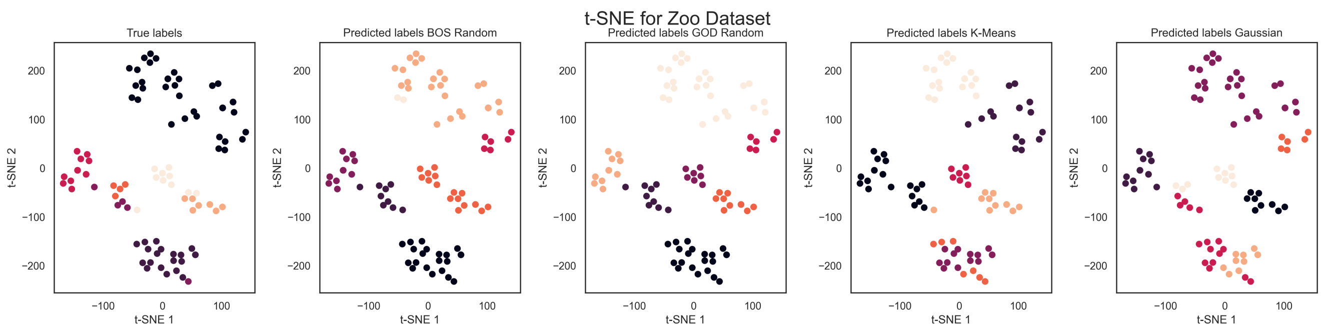

Figure 4: t-SNE visualization of the Zoo dataset with the true labels and the predicted clusters for the BOS model with random initialization, the GOD model with random initialization, the K-Means algorithm and the Gaussian Mixture Models.

Dataset	Method	**Runtime (s)**	**F1**	**Accuracy**	**Wasserstein**	**ARI**
Zoo	**BOS Random**	47.28	0.77	0.78	0.35	**<u>0.76</u>**
	BOS K-Means	4.75	**<u>0.86</u>**	**<u>0.84</u>**	0.17	0.90
	GOD Random	52.64	0.87	0.86	0.23	0.83
	GOD K-Means	15.12	0.87	0.85	**<u>0.14</u>**	0.90
	K-Means	**<u>0.01</u>**	0.70	0.64	0.84	0.58
	Gaussian	0.75	0.79	0.76	0.33	0.73
Car Evaluation	**BOS Random**	481.93	0.46	0.40	0.66	0.02
	BOS K-Means	415.80	0.42	0.37	0.73	-0.01
	GOD Random	28.95	0.43	0.40	0.41	-0.04
	GOD K-Means	19.26	0.46	0.39	0.94	0.05
	K-Means	**<u>0.01</u>**	0.41	0.34	0.91	0.01
	Gaussian	0.04	0.44	0.36	1.07	0.02
Hayes-Roth	**BOS Random**	307.60	0.37	0.39	0.25	0.00
	BOS K-Means	101.11	0.36	0.36	0.30	-0.01
	GOD Random	11.22	0.37	0.39	**<u>0.25</u>**	-0.01
	GOD K-Means	8.49	0.37	0.36	0.23	-0.01
	K-Means	**<u>0.00</u>**	0.34	0.33	0.16	-0.01
	Gaussian	0.02	**<u>0.45</u>**	**<u>0.45</u>**	0.11	**<u>0.07</u>**
Caesarian	**BOS Random**	52.78	0.41	0.56	0.41	-0.01
	BOS K-Means	35.48	0.53	0.53	0.05	-0.01
	GOD Random	3.47	0.54	0.54	**<u>0.04</u>**	-0.01
	GOD K-Means	3.51	0.59	0.59	0.09	0.02
	K-Means	**<u>0.01</u>**	0.56	0.56	0.09	0.00
	Gaussian	0.02	**<u>0.60</u>**	**<u>0.60</u>**	0.00	**<u>0.03</u>**

Table 3: Results of the classification task for the different datasets and the proposed methods. The metrics are the F1-score, the accuracy, the Wasserstein distance and the adjusted rand index (ARI). The runtime is also reported. The best results for each dataset and metric are highlighted in bold and underlined.

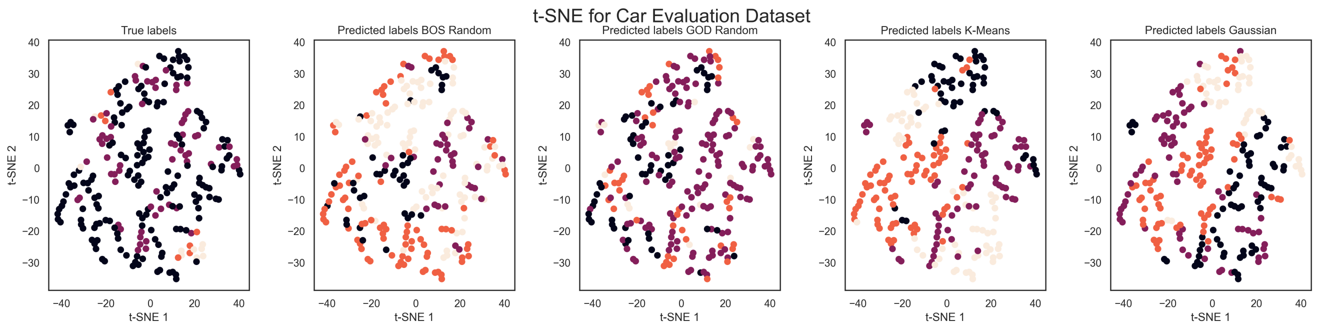

Figure 5: t-SNE visualization of the Car Evaluation dataset with the true labels and the predicted clusters for the BOS model with random initialization, the GOD model with random initialization, the K-Means algorithm and the Gaussian Mixture Models.

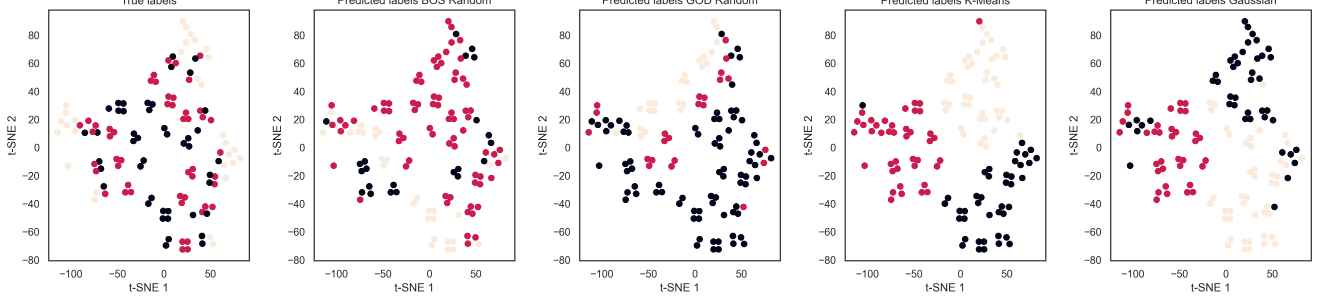

Figure 6: t-SNE visualization of the Hayes-Roth dataset with the true labels and the predicted clusters for the BOS model with random initialization, the GOD model with random initialization, the K-Means algorithm and the Gaussian Mixture Models.

D Assignment Matrices

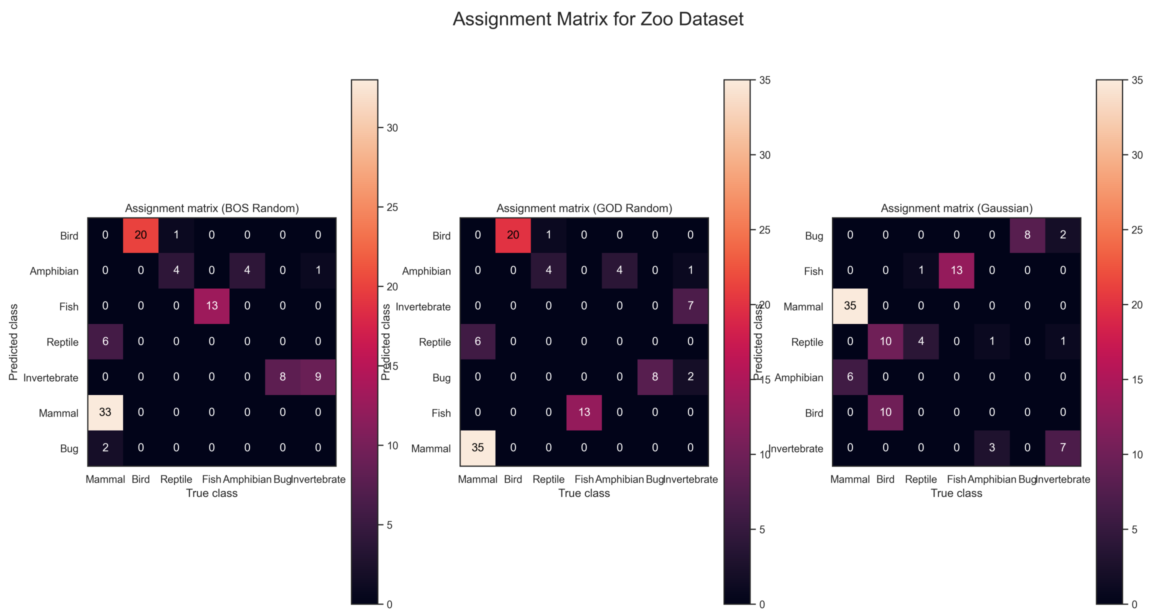

Figure 7: Assignment matrix for the Zoo dataset with different methods.

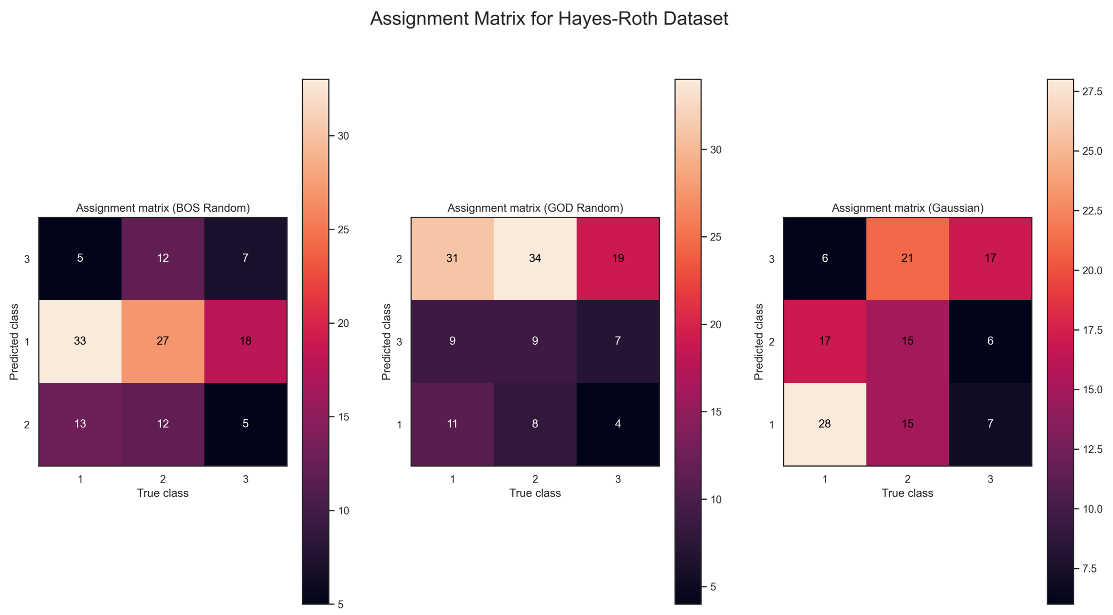

Figure 8: Assignment matrix for the Hayes-Roth dataset with different methods.

E Histograms of the different clustering

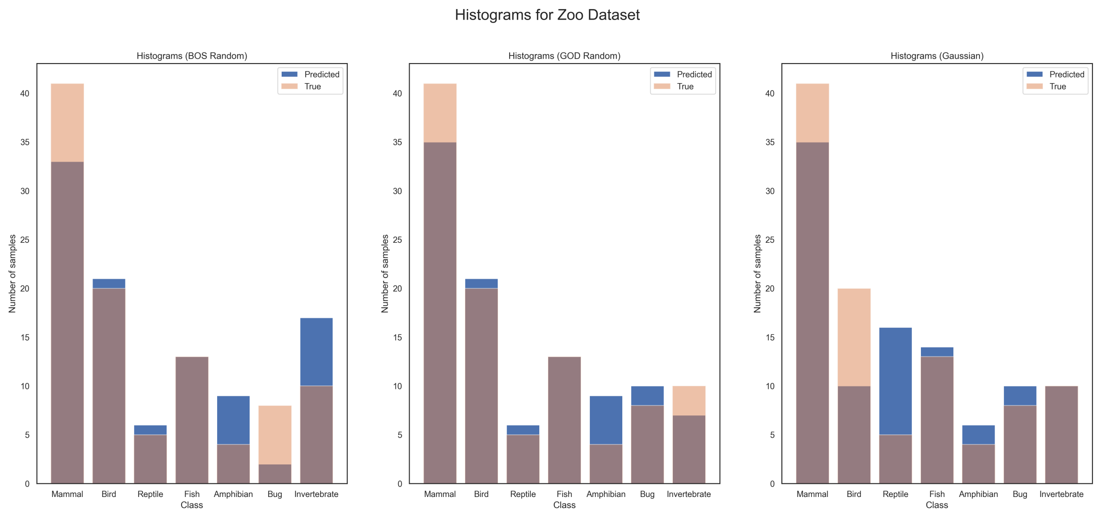

Figure 9: Histograms for the Zoo dataset with different methods.

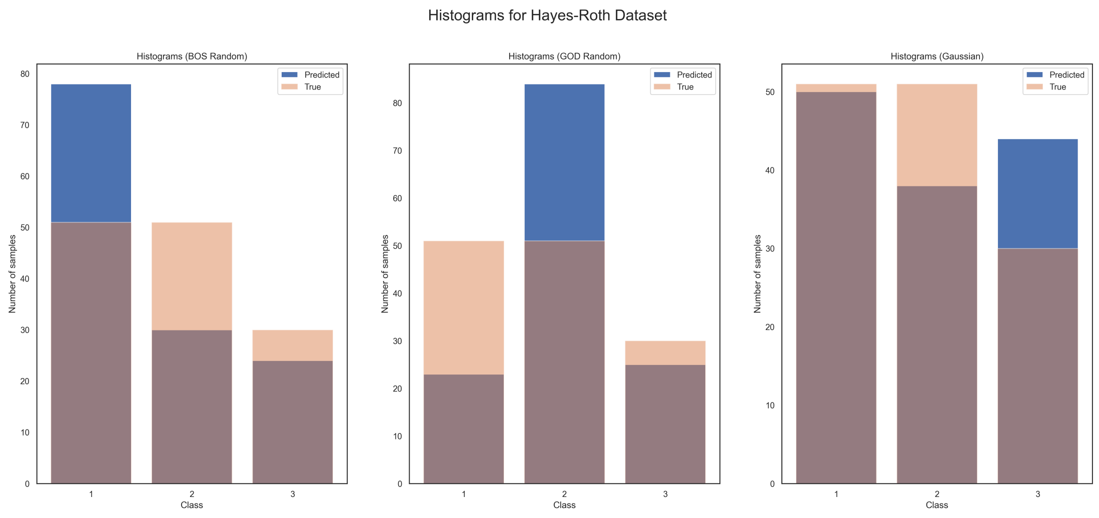

Figure 10: Histograms for the Hayes-Roth dataset with different methods.

F GOD Model proofs

Theorem 7. *If we suppose that the prior distribution of μ is uniform over $[\![1,m]\!]$ and $\pi > \frac{1}{2}$, then* $\forall c \in \{0,1\}^{m-1}$,

$$\underset{k\in[\![1,m]\!]}{\operatorname{argmax}} \Pr(\mu = k | C = c) = \underset{k\in[\![1,m]\!]}{\operatorname{argmin}} ||c - E_k||_1$$

Proof.

Lemma 6.

$$\Pr(C[i] = c[i] | \mu < i) = c[i]\pi + (1 - c[i])(1-\pi)$$

$$\Pr(C[i] = c[i] | \mu \not< i) = (1 - c[i])\pi + c[i](1-\pi)$$

Proof.

$$\begin{aligned}
&\Pr(C[i] = c[i] | \mu < i) && (36)\\
&= \Pr(C[i] = c[i] | Z[i] = 1, \mu < i) \Pr(Z[i] = 1) \\
&+ \Pr(C[i] = c[i] | Z[i] = 0, \mu < i) \Pr(Z[i] = 0) && (37)\\
&= c[i] \Pr(Z[i] = 1) + (1 - c[i]) \Pr(Z[i] = 0) && (38)\\
&= c[i]\pi + (1 - c[i])(1 - \pi) && (39)\\
&\Pr(C[i] = c[i] | \mu \not< i) && (40)\\
&= \Pr(C[i] = c[i] | Z[i] = 1, \mu \not< i) \Pr(Z[i] = 1) \\
&+ \Pr(C[i] = c[i] | Z[i] = 0, \mu \not< i) \Pr(Z[i] = 0) && (41)\\
&= (1 - c[i]) \Pr(Z[i] = 1) + c[i] \Pr(Z[i] = 0) && (42)\\
&= (1 - c[i])\pi + c[i](1 - \pi) && (43)
\end{aligned}$$

□

Lemma 7. $\forall c \in \{0,1\}^m, \forall k \in [\![1, m]\!]$,

$$\Pr(C = c | \mu = k) = \pi^{m-1-||c-E_k||_1}(1 - \pi)^{||c-E_k||_1}$$

Proof. Let us compute for $i \in [\![1, m]\!]$, $\Pr(C = c|\mu = i)$ as the $C[i]|\mu$ are independent and using the previous lemma:

$$\begin{aligned}
\Pr(C = c | \mu = k) &= \prod_{i=1}^{m-1} \Pr(C[i] = c[i] | \mu = k) && (44)\\
&= \prod_{i=1}^{k-1} \Pr(C[i] = c[i] | \mu < i) \prod_{i=k}^{m-1} \Pr(C[i] = c[i] | \mu \not< i) && (45)\\
&= \prod_{i=1}^{k-1} [c[i]\pi + (1 - c[i])(1 - \pi)] \prod_{i=k}^{m-1} [(1 - c[i])\pi + c[i](1 - \pi)] && (46)\\
&= \pi^{\sum_{i=1}^{k-1} c[i]} (1 - \pi)^{\sum_{i=1}^{k-1}(1-c[i])} \pi^{\sum_{i=k}^{m-1}(1-c[i])} (1 - \pi)^{\sum_{i=k}^{m-1} c[i]} && (47)\\
&= \pi^{\sum_{i=1}^{k-1} c[i] + \sum_{i=k}^{m-1}(1-c[i])} (1 - \pi)^{\sum_{i=1}^{k-1}(1-c[i]) + \sum_{i=k}^{m-1} c[i]} && (48)\\
&= \pi^{m-1-\left[\sum_{i=1}^{k-1}(1-c[i]) + \sum_{i=k}^{m-1} c[i]\right]} (1 - \pi)^{\sum_{i=1}^{k-1}(1-c[i]) + \sum_{i=k}^{m-1} c[i]} && (49)\\
&= \pi^{m-1-||E_k-c||_1} (1 - \pi)^{||E_k-c||_1} && (50)
\end{aligned}$$

□

$$\begin{aligned}
\Pr(\mu = k | C = c) &= \frac{\Pr(C = c | \mu = k) \Pr(\mu = k)}{\Pr(C = c)} && (51)\\
&= \frac{\Pr(C = c | \mu = k) \Pr(\mu = k)}{\sum_{i=1}^{m} \Pr(C = c | \mu = i) \Pr(\mu = i)} && (52)
\end{aligned}$$

As μ is uniformly distributed over $[\![1, m]\!]$, $\Pr(\mu = k) = \frac{1}{m}$

$$\Pr(\mu = k|C = c) = \frac{\Pr(C = c|\mu = k)}{\sum_{i=1}^{m} \Pr(C|\mu = i)} \tag{53}$$

using Lemma 7:

$$\Pr(\mu = k|C = c) = \frac{\pi^{m-1-||c-E_k||_1}(1-\pi)^{||c-E_k||_1}}{\sum_{i=1}^{m} \pi^{m-1-||c-E_i||_1}(1-\pi)^{||c-E_i||_1}} \tag{54}$$

$$\tag{55}$$

As $\pi > \frac{1}{2}$, we conclude that:

$$\operatorname*{argmax}_{k \in [\![1,m]\!]} \Pr(\mu = k|C = c) = \operatorname*{argmin}_{k \in [\![1,m]\!]} ||c - E_k||_1$$

□

Lemma 8.

$$\Pr(x, c|\pi, \mu) = \mathbb{1}_{\mathcal{C}_x}(c)\pi^{m-1} \frac{\left(\frac{1-\pi}{\pi}\right)^{||c-E_\mu||_1}}{\left|\operatorname{argmin}_{k \in [\![1,m]\!]} ||c - E_k||_1\right|}$$

Proof. Using Bayes' theorem, then Lemma 7 and the fact that μ is uniformly distributed over the set defined by the argmin, we have:

$$\begin{aligned}
\Pr(x, C = c|\pi, \mu) &= \Pr(x|c, \pi, \mu) \Pr(C = c|\pi, \mu) && (56)\\
&= \mathbb{1}_{\mathcal{C}_x}(c) \Pr(x|c \in \mathcal{C}_x, \pi, \mu) \Pr(c|\pi, \mu) && (57)\\
&= \mathbb{1}_{\mathcal{C}_x}(c) \frac{\pi^{m-1-||c-E_\mu||_1}(1-\pi)^{||c-E_\mu||_1}}{\left|\operatorname{argmin}_{k \in [\![1,m]\!]} ||c - E_k||_1\right|} && (58)\\
&= \mathbb{1}_{\mathcal{C}_x}(c)\pi^{m-1} \frac{\left(\frac{1-\pi}{\pi}\right)^{||c-E_\mu||_1}}{\left|\operatorname{argmin}_{k \in [\![1,m]\!]} ||c - E_k||_1\right|} && (59)
\end{aligned}$$

□

Theorem 8 (Observation likelihood)**.**

$$\Pr(x|\pi, \mu) = \pi^{m-1} \sum_{d=0}^{m-1} \left(\frac{1-\pi}{\pi}\right)^d u(x, \mu, d)$$

Proof. By marginalizing over c and then using the previous lemma, we have:

$$\begin{aligned}
\Pr(x|\pi, \mu) &= \sum_{c \in \{0,1\}^{m-1}} \Pr(x, c|\pi, \mu) && (60)\\
&= \pi^{m-1} \sum_{c \in \mathcal{C}_x} \left(\frac{1-\pi}{\pi}\right)^{||c-E_\mu||_1} \left|\operatorname*{argmin}_{k \in [\![1,m]\!]} ||c - E_k||_1\right|^{-1} && (61)\\
&= \pi^{m-1} \sum_{d=0}^{m-1} \left(\frac{1-\pi}{\pi}\right)^d \sum_{c \in \mathcal{C}_x / ||c-E_\mu||_1 = d} \left|\operatorname*{argmin}_{k \in [\![1,m]\!]} ||c - E_k||_1\right|^{-1} && (62)
\end{aligned}$$

□

Lemma 9 (Concavity of log composed functions)**.** *For $f : I \to \mathbb{R}_+^*$ be a twice-differentiable function, we have that:*

$$\ln \circ f \text{ is concave} \iff f'^2 - ff'' \geqslant 0$$
$$\ln \circ f \text{ is strictly} \iff f'^2 - ff'' > 0$$

Proof. We have that:

$$(\ln \circ f)'' = \frac{f''f - f'^2}{f^2}$$

Therefore, $\ln \circ f$ is concave if and only if $f'^2 - ff'' \geqslant 0$ and strictly concave if and only if $f'^2 - ff'' > 0$. □

Lemma 10. *We define for $d \in \mathbb{N}$,*

$$c_d : \begin{cases} [\frac{1}{2}, 1] & \to \mathbb{R}_+^* \\ x & \mapsto \left(\frac{1-x}{x}\right)^d \end{cases}$$

We have that $\forall d \in \mathbb{N}, \forall x \in]\frac{1}{2}, 1[$

$$c_d'(x)^2 - c_d(x)c_d''(x) > 0$$

and $\forall x \in [\frac{1}{2}, 1]$:

$$c_d'(x)^2 - c_d(x)c_d''(x) \geqslant 0$$

Proof. We have that:

$$c_d'(x) = -dx^{-2}\left(\frac{1-x}{x}\right)^{d-1}$$

$$\begin{aligned} c_d''(x) &= 2dx^{-3}\left(\frac{1-x}{x}\right)^{d-1} + d(d-1)x^{-4}\left(\frac{1-x}{x}\right)^{d-2} && (63)\\ &= dx^{-4}\left(\frac{1-x}{x}\right)^{d-2}\left(2x\left(\frac{1-x}{x}\right) + (d-1)\right) && (64)\\ &= dx^{-4}\left(\frac{1-x}{x}\right)^{d-2}(1-2x+d) && (65) \end{aligned}$$

Therefore, we have that:

$$\begin{aligned} c_d'(x)^2 - c_d(x)c_d''(x) &= d^2x^{-4}\left(\frac{1-x}{x}\right)^{2d-2} - dx^{-4}\left(\frac{1-x}{x}\right)^{2d-2}(1-2x+d) && (66)\\ &= dx^{-4}\left(\frac{1-x}{x}\right)^{2d-2}(d-1+2x-d) && (67)\\ &= dx^{-4}\left(\frac{1-x}{x}\right)^{2d-2}(2x-1) && (68) \end{aligned}$$

We get the desired result as $2x - 1 > 0$ on $]\frac{1}{2}, 1[$. □

Theorem 9. $\forall \mu \in [\![1, m]\!], \forall x \in [\![1, m]\!]$

$$\pi \mapsto \Pr(x|\pi, \mu)$$

is stricly log*-concave on $]\frac{1}{2}, 1[$ and* log*-concave on $[\frac{1}{2}, 1]$.*

Proof. We use the following expression:

$$\log \Pr(x|\pi,\mu) = (m-1)\log \pi + \log\left[\sum_{d=0}^{m-1}\left(\frac{1-\pi}{\pi}\right)^{d} u(x,\mu,d)\right]$$

As $\pi \mapsto (m-1)\log \pi$ is concave and the sum of positive weighted $(m-1 \geqslant 0)$ concave functions is concave, we only need to prove that $\ln g$ is concave where:

$$g : t \mapsto \sum_{d=0}^{m-1} c_d(t) u_d$$

As we will only use the fact that $u(x,\mu,d) \geqslant 0$ we replace the $u(x,\mu,d)$ by a generic u_d.

Using the lemma 9 we just have to check that $\forall t \in]\frac{1}{2},1[, g'(t)^2 - g(t)g''(t) > 0$.

Let $t \in\in]\frac{1}{2},1[$. As g is a positively weighted sum of c_d and as each c_d verify that $c_d'(t)^2 - c_d(t)c_d''(t) > 0$ we have $g'(t)^2 - g(t)g''(t) \geqslant 0$.

We can conclued that $\Pr(x|\bullet,\mu)$ is stricly log-concave on $]\frac{1}{2},1[$ and using the same argument that it is log-concave on $[\frac{1}{2},1]$. ☐

Open combinatorial problem

Definition 7 (Heaviside vector). *For $k \in [\![1,m]\!]$, we define:*

$$E_k := (1)^{k-1}(0)^{m-k} = (\underbrace{1,\dots,1}_{k-1},\underbrace{0,\dots,0}_{m-k}).$$

We define for $x \in [\![1,m]\!]$,

$$\mathcal{C}_x := \left\{ c \in \{0,1\}^{m-1} \,|x \in \underset{k\in[\![1,m]\!]}{\operatorname{argmin}} ||c - E_k||_1 \right\}$$

The goal is to find an algorithm that compute for every $x \in [\![1,m]\!]$, $\mu \in [\![1,m]\!]$ and $d \in [\![0,m-1]\!]$ $u(d,\mu,x)$ where:

$$u(\mu,x,d) := \sum_{c\in\mathcal{C}_x/||c-E_\mu||_1=d} \left| \underset{k\in[\![1,m]\!]}{\operatorname{argmin}} ||c-E_k||_1 \right|^{-1}$$

The algorithm could be efficient at computing only all this values at the same time. We have an algorithm that compute $u(d,\mu,x)$ in $O(m2^m)$ for all d,x,μ. Is there a better algorithm? (Or alternatively is the problem NP-hard?)

Lemma 11.

$$\sum_{c\in\{0,1\}^{m-1}} \left| \underset{k\in[\![1,m]\!]}{\operatorname{argmin}} ||c-E_k||_1 \right| = 2^m - \binom{m}{\lfloor \frac{m}{2} \rfloor}$$

Proof. May include the fact that:

$$2^m - \binom{m}{\lfloor \frac{m}{2} \rfloor} = \sum_{k=0}^{m-1} \binom{m}{\lfloor \frac{k}{2} \rfloor}$$

□

Lemma 12. $\forall m \geqslant 3$*:*

$$\sum_{c\in\{0,1\}^{m-1}} \mathbb{1}(\left| \underset{k\in[\![1,m]\!]}{\operatorname{argmin}} \right| = 2) = 2^{m-3}$$

$\forall m \geqslant 4$*:*

$$\sum_{c\in\{0,1\}^{m-1}} \mathbb{1}(\left| \underset{k\in[\![1,m]\!]}{\operatorname{argmin}} \right| = 3) = 2^{m-4} - \binom{m-4}{\lfloor \frac{m-4}{2} \rfloor}$$

$\forall m \geqslant 5$*:*

$$\sum_{c\in\{0,1\}^{m-1}} \mathbb{1}(\left| \underset{k\in[\![1,m]\!]}{\operatorname{argmin}} \right| = 4) = 2^{m-5} - \binom{m-4}{\lfloor \frac{m-4}{2} \rfloor}$$

$\forall m \geqslant 4$*:*

$$\sum_{c\in\{0,1\}^{m-1}} \mathbb{1}(\left| \underset{k\in[\![1,m]\!]}{\operatorname{argmin}} \right| = 5) = A191389[m-4]$$