Clustering Milky Way's Globulars: a Bayesian Nonparametric Approach

Julyan Arbel

Clustering Milky Way's Globulars: a Bayesian Nonparametric Approach

Julyan Arbel[1,⋆]

[1] *Univ. Grenoble Alpes, Inria, CNRS, Grenoble INP, LJK, 38000 Grenoble, France.*

Abstract. This chapter presents a Bayesian nonparametric approach to clustering, which is particularly relevant when the number of components in the clustering is unknown. The approach is illustrated with the Milky Way's globulars, that are clouds of stars orbiting in our galaxy. Clustering globulars is key for better understanding the Milky Way's history. We define the Dirichlet process and illustrate some alternative definitions such as the Chinese restaurant process, the Pólya Urn, the Ewens sampling formula, the stick-breaking representation through some simple R code. The Dirichlet process mixture model is presented, as well as the R package BNPmix implementing Markov chain Monte Carlo sampling. Inference for the clustering is done with the variation of information loss function.

1 R requirements

The code used during the presentation of the Stat4Astro summer school is available at the url: https://github.com/jarbel/Stat4Astro-Autrans. Additionally, the code used to generate the plots of this chapter is displayed in the text. This requires the following R packages: `ggplot2`, `hexbin`, `viridis`, `gridExtra`, `ggpubr`, `rgl` for graphical tools, `reshape2` for operations on similarity matrices, `mcclust` and `mcclust.ext` for clustering estimation.

```
needed_packages <- c("ggplot2", "hexbin", "viridis", "gridExtra",
                     "ggpubr", "rgl", "reshape2", "dplyr", "mcclust")
new_packages <- needed_packages[
  !(needed_packages %in% installed.packages()[, "Package"])]
if (length(new_packages))
  install.packages(new_packages)
lapply(needed_packages, require, character.only = TRUE)
```

```
download.file(
  url = "http://wrap.warwick.ac.uk/71934/1/mcclust.ext_1.0.tar.gz",
  destfile = "mcclust.ext_1.0.tar.gz")
```

⋆e-mail: julyan.arbel@inria.fr

```
install.packages("mcclust.ext_1.0.tar.gz", repos = NULL, type = "source")
file.remove("mcclust.ext_1.0.tar.gz")
library("mcclust.ext")
```

2 Introduction and motivation

Globulars[1] are sets of stars orbiting some galactic center. The globular data we are considering here was studied in the 2015 Edition of the Stat4Astro school by [10] who used phylogenetic classification. The data are available on GitHub and can be downloaded as follows:

```
spectra <- read.csv(
  "https://github.com/jarbel/Stat4Astro-Autrans/blob/
  master/Talk_Arbel/bnp_code/data/GC4c_groups.dat",
  sep="")
```

It lists a total of `dim(spectra)[1]`=54 globulars for which `dim(spectra)[2]`=7 variables are available[2]: `GC` stands for the globular identifier; `logTe` is the logarithm of the maximum effective temperature on the horizontal branch; `FeH` denotes the metallicity; `MV` is the absolute V magnitude, which relates to both the brightness and the mass of the globular; `Age` of the globular; `Grp4c` and `Grp3c` are the phylogenetic classifications of [10] obtained by using respectively the four variables `logTe, FeH, MV, Age` and the three variables `logTe, FeH, MV`.

```
## [1] 54  7

## [1] "GC"    "logTe" "FeH"   "MV"    "Age"   "Grp4c" "Grp3c"
names(spectra)

## [1] "GC"    "logTe" "FeH"   "MV"    "Age"   "Grp4c" "Grp3c"
```

By using the additional spatial coordinates available on the Wikipedia list[2], we can obtain a spatial scatterplot of the globulars. In Figure 1, the globulars present in the study (that is for which we have measurements for the above mentionned variables) are depicted in purple, the others in green

The two clusterings `Grp4c` and `Grp3c` are of respective size 3 and 4, and the cluster sizes are obtained with the `table` command as follows:

```
table(spectra$Grp3c)

##
##  1  2  3  4
## 17  7 21  9
```

[1] Globulars are more commonly called globular clusters in the literature, though we shall prefer the phrasing 'globulars' to 'globular clusters' in order to avoid ambiguous terms like 'globular clusters clusters'.

[2] A more comprehensive list of globulars can be accessed at https://en.wikipedia.org/wiki/List_of_globular_clusters, though it contains information about magnitude and diameter only.

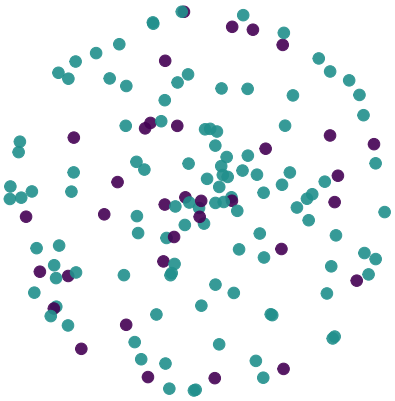
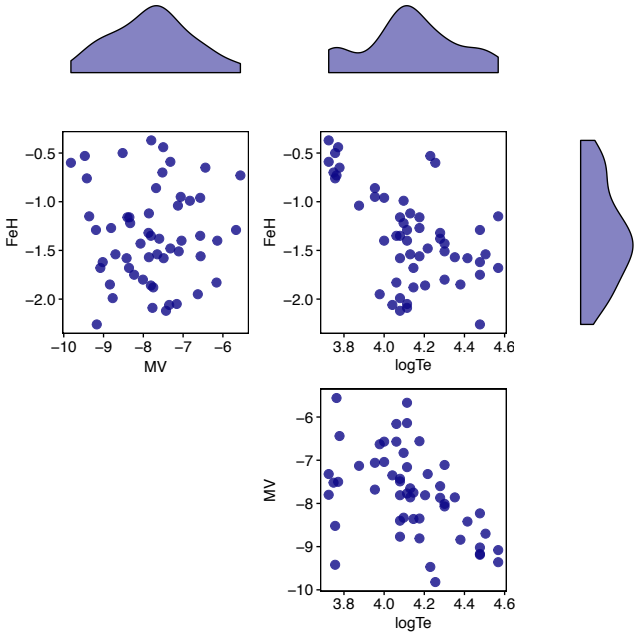


Figure 1: Globular coordinates. Gobulars included in the present study are depicted in purple, the others in green

```
table(spectra$Grp4c)

##
##  1  2  3
## 25 11 18
```

From now on, we will focus on the three variables `logTe, FeH, MV` only. A 2-D representation for each pair of variables, as well as a side density plot for each variable can be obtained by:

```
col <- plasma(1)
alpha <- .8
spVF <- ggscatter(spectra, x = "MV", y = "FeH", color = col,
                  size = 3, alpha = alpha)+ border()
spLF <- ggscatter(spectra, x = "logTe", y = "FeH", color = col,
                  size = 3, alpha = alpha)+ border()
spLV <- ggscatter(spectra, x = "logTe", y = "MV", color = col,
                  size = 3, alpha = alpha) + border()
Vplot <- ggdensity(spectra, "MV", fill = col)
Lplot <- ggdensity(spectra, "logTe", fill = col)
Fplot <- ggdensity(spectra, "FeH", fill = col) + rotate()
Vplot <- Vplot + clean_theme()
Lplot <- Lplot + clean_theme()
Fplot <- Fplot + clean_theme()
ggarrange(Vplot, Lplot, NULL,
          spVF, spLF, Fplot,
          NULL, spLV, NULL,
          ncol = 3, nrow = 3,  align = "hv",
          widths = c(2, 2, 1), heights = c(1, 2, 2))
```



These scatter plots hardly identify any clustering structure amongst the globulars. However, astrophysicists expect several populations of globulars identified by similar time of formation, and similar chemical and physical conditions. The granularity of the populations of globulars may depend on the size of the considered sample: a small number of observations would likely lead to little discriminating power, while a large sample size would provide more evidence for identifying more distinct clusters. This setting where the number of clusters might grow with the sample size is well suited to a Bayesian nonparametric approach to clustering, which this chapter aims at introducing.

The rest of the chapter is organised as follows. Model-based clustering and the Dirichlet process are introduced in Section 3 and Section 4. We conclude with an illustration to the Globular dataset.

3 Model-based clustering

Mixture models are creating flexible models starting from simple ones. For instance, combining two unimodal densities p_1 and p_2 into $\pi p_1 + (1-\pi)p_2$, for $\pi \in (0,1)$ can create a bimodal distribution. The aim of mixtures is to increase the modelling capacities by combining simple distributions into flexible distributions which might display multimodality, skewness, etc. A mathematical definition of a mixture density is a convex combination of densities. Each density can be interpreted as a sub-population. Observations are associated to sub-populations through latent (un-observed) variables called allocation variables, which play a major role in devising sampling algorithms.

Consider a parametric family of distributions with densities $p(\cdot|\phi)$, ϕ element of some parameter space Θ, with respect to a common measure. Broadly speaking, mixtures operate in a discrete way (finite or infinite countable number of classes) or in a continuous way (infinite uncountable number of classes). As an example of continuous mixtures, a Student t-distribution can be written as a mixture of Gaussian kernels with fixed mean and variance

averaged over an inverse gamma distribution (playing the role of a mixing distribution). We will instead focus here on discrete mixtures.

Going back to the simple mixture of two densities, the mixture density takes on the form $\pi p_1(\cdot|\phi_1) + (1-\pi)p_2(\cdot|\phi_2)$, where the unknown parameters are (π, ϕ_1, ϕ_2). Modelling data $\mathbf{x} = (x_1, \ldots, x_N)$ with such a density can be thought as the following two-step procedure, for $i \in \{1, \ldots, N\}$:

1. toss a coin, which samples from one of the two classes $\theta_i \in \{\phi_1, \phi_2\}$. In other words, the class identified with parameter ϕ_1 has probability π, while the ϕ_2 class has the complement probability $1 - \pi$;

2. sample the observation from the corresponding density $x_i \sim p(x|\theta_i)$.

The coin-tossing step can be equally thought of as a throw of dice whose number of facets equals the number of components in the mixture, and each facet has a probability given by the component weight. This mixing distribution takes the mathematical form of a convex combination $\pi\delta_{\phi_1} + (1-\pi)\delta_{\phi_2}$ of Dirac masses at ϕ_1 and ϕ_2, each of which identifies with a class of the mixture.

We speak about nonparametric (discrete) mixtures when the number of classes is (countable) infinite. In which case the dice has an infinite number of facets. We index the classes by the positive integers, and denote the parameters by ϕ_k and weights by π_k. Then those weights must sum up to one, and the mixing measure can be written as

$$G = \sum_{k=1}^{\infty} \pi_k \delta_{\phi_k},$$

which is a probability measure. Sampling from the mixture distribution

$$p_G(\cdot) = \sum_{k=1}^{\infty} \pi_k p(\cdot|\phi_k)$$

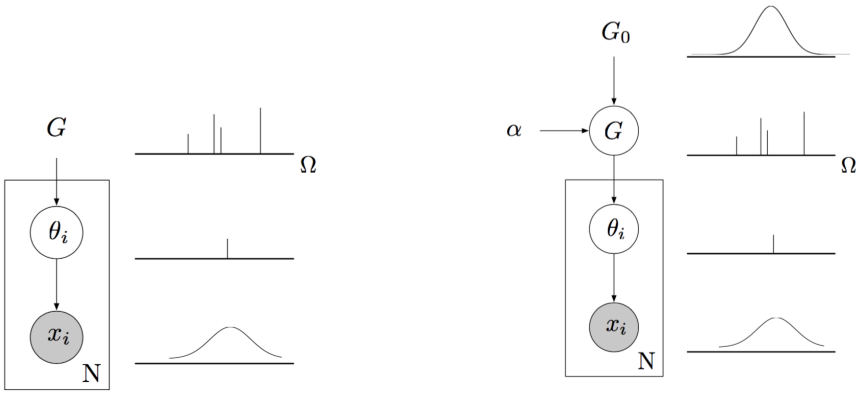
can again be done in the sequential way, for $i \in \{1, \ldots, N\}$:

1. throw the infinite dice G: $\theta_i \sim G$;

2. sample the observation from the corresponding density $x_i \sim p(x|\theta_i)$.

The mixture density p_G has an infinite number of parameters: the vector of weights $(\pi_1, \pi_2, \ldots)$ which is an element of the infinite simplex, and the vector of class parameters $(\phi_1, \phi_2, \ldots)$. A Bayesian approach to the problem requires endowing the parameters with some prior distribution. The Dirichlet process is the most prominent example of such a prior distribution. It is parametrised by the base measure, denoted by G_0, and the precision parameter, a positive scalar denoted by α. Figure 2 displays plate representations of a Bayesian and a non Bayesian (or frequentist) approach to the mixture model. Such a plate representation makes it clear that a Bayesian approach add a hierarchical layer (that of the prior) to the non Bayesian set-up.

Choosing the number of components to include in a finite mixture is often a delicate question, as one rarely knows it a priori. There are mainly three strategies: i) to fit several finite mixture models with a range of plausible values for the number of components; then choose the model which maximises some information criterion; ii) consider the number of



(a) Frequentist setting: non random G. (b) Bayesian approach: $G \sim \mathrm{DP}(\alpha G_0)$.

Figure 2: Plate representations of a nonparametric mixture with mixing measure G.

components as Bayesian parameter, which is to say endow it with a prior distribution on positive integers, such as a Poisson distribution. Such a model is termed a mixture of finite mixtures (MFM) by [15]; or iii) let this number be a priori infinite (or as large as needed) and select it or estimate it a posteriori. We focus on this last option in the next section.

4 Bayesian nonparametrics around the Dirichlet process

Most of the models described in the previous chapters of this book are parametric: they can be described by a finite and fixed number of parameters. This number of parameters is independent of the dataset. Very convenient models in a Bayesian setting include conjugate models, where the posterior distribution has the same form as the prior distribution. For example, normal prior and normal likelihood, or beta prior and binomial likelihood, or gamma prior and Poisson likelihood. In contrast, we are dealing in this chapter with *nonparametric* models. Let us put it straightaway, the *nonparametric* saying is not the most fortunate, as nonparametric models *do* have parameters, many of them. [6] define a *Bayesian nonparametric model* as a probability model with infinitely many parameters, also referred to by [16] as a model with massively many parameters. I think there are three ways this large number of parameters can be thought of: the number of parameters is *infinite*, or it is *random*, or it is *growing* with the data sample size.

It goes without saying that parametric models are easier to handle than their nonparametric counterpart, that are computationally and analytically more challenging. It is also true that interpreting a small and fixed number of parameters is likely to be easier than for, say, an infinite number of them. However, the computational and analytical extra burden is the price to pay for flexibility. Nonparametric models are less prone to misspecification than parametric models, which require a strong belief in the particular structure they imply. Without such a belief in the parametric assumption, a model might not be reliable. By nature, nonparametric models are well suited to study curves in general, so typical applications include density estimation and regression estimation. Clustering is also an example which has historically played a central role within Bayesian nonparametrics.

In this section we provide basic facts about the Dirichlet process, precisely different possible representations: the definition via the finite dimensional marginal distribution in Definition 4.2, the Chinese restaurant process in Proposition 4.3, the posterior distribution in Theorem 4.1, the Pólya Urn in Proposition 4.2 and the stick-breaking representation in Theorem 4.2.

4.1 Definition

The Dirichlet process [9] plays a central role in Bayesian nonparametrics. A Dirichlet process can be viewed as a random variable where the variable is a probability measure. It has two parameters: the base measure, denoted by G_0, and the precision parameter, a positive scalar denoted by α.

[9] defines the Dirichlet process by its finite marginal distributions: ie, how does the random measure spread its mass in the sample space? The answer is: as a Dirichlet distribution (which, in passing, explains the name). Recall that the Dirichlet distribution generalises the Beta distribution to any dimension $k \geq 2$. The Dirichlet distribution in $\mathbb{R}^k$ is restricted to the unit simplex, with density parametrised by k positive scalars $\alpha_1, \ldots, \alpha_k$ and proportional to

$$x_1^{\alpha_1 - 1} \cdots x_k^{\alpha_k - 1}.$$

Definition 4.1 (Dirichlet distribution)**.** *A Dirichlet distribution on a simplex Δ_K is a probability distribution with parameters $\alpha_i > 0$ and a density function*

$$f(x_1, \ldots, x_K; \alpha_1, \ldots, \alpha_K) = \frac{1}{B(\alpha)} \prod_{i=1}^{K} x_i^{\alpha_i - 1}.$$

It is common to refer to Dirichlet distribution as $\mathrm{Dir}(x_1, \ldots, x_k)$. Let us remark that

Remark 4.1. *The Dirichlet distribution is conjugate prior for the multinomial distribution.*

Consider a finite partition of the space, denoted by $(A_1, \ldots, A_k)$. The mass allocated by a Dirichlet process G to each region A_j is a random variable $G(A_j)$, thus giving rise to a random vector $G(A_1), \ldots, G(A_k)$ for the whole partition. Then G follows a Dirichlet process with parameters α and G_0 if the random vector $G(A_1), \ldots, G(A_k)$ is distributed as a Dirichlet distribution with parameters $\alpha G_0(A_1), \ldots, \alpha G_0(A_k)$.

Definition 4.2 (Dirichlet process, [9])**.** *A random probability measure G follows a Dirichlet process with parameters α and G_0 on some space if for any finite partition $(A_1, \ldots, A_k)$ of the space,*

$$(G(A_1), \ldots, G(A_k)) \sim \mathrm{Dir}(\alpha G_0(A_1), \ldots, \alpha G_0(A_k)).$$

Note that this definition entails the strong result that such finite dimensional distributions consistently define a stochastic process. Figure 3 shows different realisations of a Dirichlet process in $\mathbb{R}^2$, with a standard Gaussian base measure G_0 and precision parameter equal to one, and with different partitions. Of course, not all of the sample space $\mathbb{R}^2$ can be represented, but most of the mass is captured in the represented part of the space.

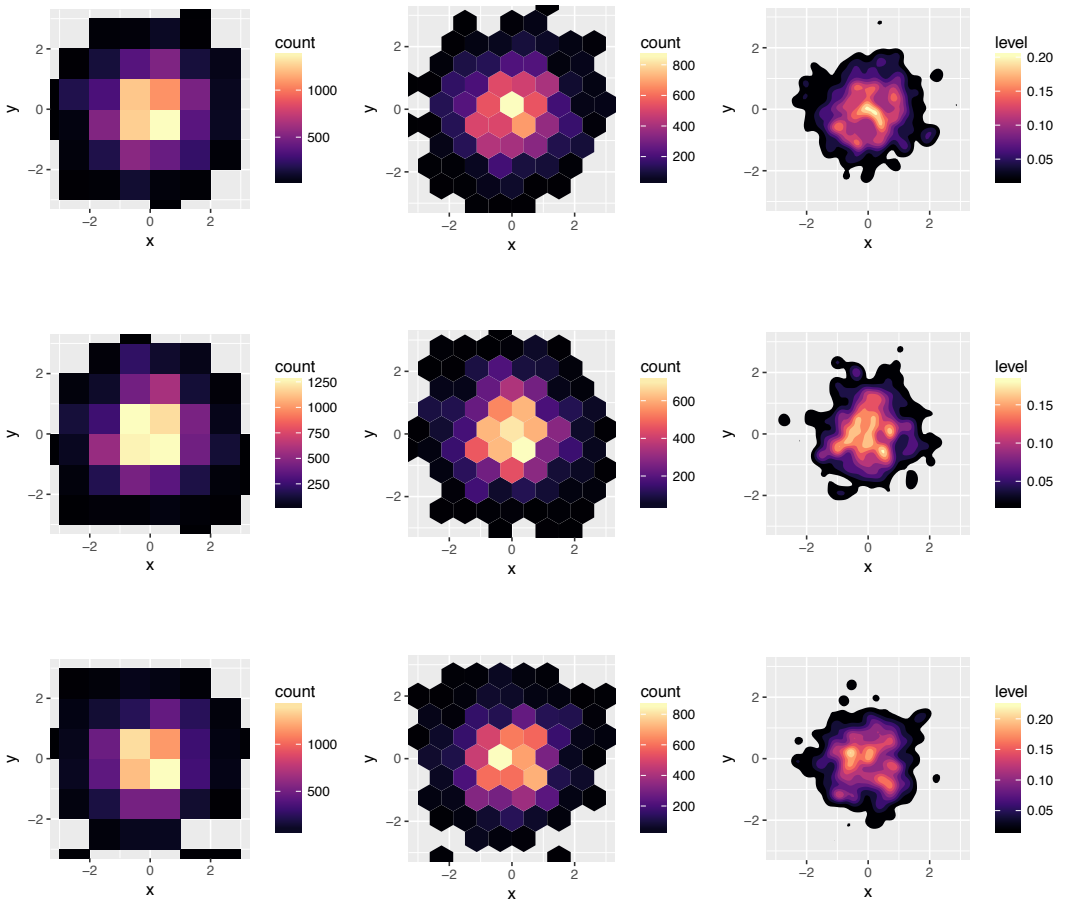


Figure 3: Dirichlet process realisations in $\mathbb{R}^2$, varying in rows. Different partitions of $\mathbb{R}^2$ are used in columns. The base measure G_0 is a standard Gaussian and the precision parameter is equal to one.

4.2 Properties

From Definition 4.2, we see that any measure set A of the space receives mass according to the following Beta distribution

$$P(A) \sim \text{Beta}(\alpha G_0(A), \alpha(1 - G_0(A)). \tag{1}$$

We have the following moments.

Proposition 4.1 (Dirichlet process moments)**.** *If $G \sim \mathrm{DP}(\alpha G_0)$, then for any measurable sets A and B*

$$\mathrm{E}[G(A)] = G_0(A)$$
$$\mathrm{Var}[G(A)] = \frac{G_0(A)(1 - G_0(A))}{1 + \alpha}$$
$$\mathrm{Cov}[G(A), G(B)] = \frac{G_0(A \cap B) - G_0(A)G_0(B)}{1 + \alpha}$$

Proof. We will make use of Equation (1). From this we obtain

$$\mathrm{E}(P(A)) = \frac{\alpha P_0(A)}{\alpha(P_0(A) + 1 - P_0(A))} = P_0(A)$$

and

$$\mathrm{Var}(P(A)) = \frac{\alpha^2 P_0(A)(1 - P_0(A))}{\alpha^2(\alpha + 1)}.$$

We derive the covariance term in two cases, firstly taking into consideration the one with $A \cap B = \emptyset$. In that case the sample space is partitioned into A, B and $(A \cup B)^c$, the complementary set of $A \cup B$, which is equal to $(A \cup B)^c = A^c \cap B^c$. Therefore we may write a joint probability vector

$$\big(P(A), P(B), P(A^c \cap B^c)\big) \sim \mathrm{Dir}\big(\alpha P_0(A), \alpha P_0(B), \alpha P_0(A^c \cap B^c)\big),$$

and hence $\mathrm{Cov}(P(A), P(B)) = -P_0(A)P_0(B)/(1 + \alpha)$. In the more general case one may decompose

$$A = (A \cap B) \cup (A \cap B^c)$$
$$B = (B \cap A) \cup (B \cap A^c),$$

so that

$$\mathrm{Cov}(P(A), P(B)) = \mathrm{Cov}(P(A \cap B) + P(A \cap B^c), P(B \cap A) + P(B \cap A^c))$$

and so forth using the linearity of covariance. □

In passing, this shows that any two measurable parts of the space which are non intersecting receive mass from the Dirichlet process with negative correlation. Which makes sense since the total measure is constrained to be a probability measure, so more mass in some part of the space means less in the rest of the space.

Another central property is the conjugacy of the Dirichlet process: if some data are sampled from a Dirichlet process G, then the posterior distribution of G conditional on the data is still a Dirichlet process.

Theorem 4.1 (Dirichlet process posterior distribution, [9])**.** *Let data $\boldsymbol{X} = (X_1, \ldots, X_n)$ be distributed according to the model*

$$P \sim \mathrm{DP}(\alpha G_0)$$
$$X_1, \ldots, X_n | G \overset{iid}{\sim} G.$$

Then the posterior distribution of G is given by

$$G | X_1, \ldots, X_n \sim \mathrm{DP}\left(\alpha P_0 + \sum_{i=1}^{n} \delta_i\right).$$

Proof. This posterior can be obtained by remarking that for any finite measurable partition $(A_1, \dots, A_k)$, the posterior distribution of $P(A_1), \dots, P(A_k)$ depends on the observations only via their cell counts (this comes from the *tail-free* property of the DP). Denote $N_j = \#\{1 \leq i \leq n : x_i \in A_j\}$, i.e. the number of observations in each cell of the partition of $(A_1, \dots, A_k)$. Then we have

$$\big(P(A_1), \dots, P(A_k)\big)\big|X_{1:n} \overset{\mathrm{d}}{=} \big(P(A_1), \dots, P(A_k)\big)\big|N_{1:k}.$$

Let us use the shorthand notation: $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_k) = (P(A_1), \dots, P(A_k))$ and $\boldsymbol{N} = (N_1, \dots, N_k)$. Then

$$\begin{cases} \boldsymbol{\alpha} \sim \mathrm{Dir}_k(\alpha P_0(A_1), \dots, \alpha P_0(A_k)), \\ \quad \boldsymbol{N}|P \sim \mathrm{Multinom}_k(n, \boldsymbol{\alpha}), \end{cases}$$

and hence we obtain the prior of the form

$$p(\boldsymbol{\alpha}) \propto \alpha_1^{\alpha P_0(A_1)-1} \cdots \alpha_k^{\alpha P_0(A_k)-1},$$

while sampling model is

$$p(\boldsymbol{N}|\boldsymbol{\alpha}) \propto \alpha_1^{N_1} \cdots \alpha_k^{N_k}.$$

This results in the posterior of form

$$p(\boldsymbol{\alpha}|\boldsymbol{N}) \propto \alpha_1^{\alpha P_0(A_1)+N_1-1} \cdots \alpha_k^{\alpha P_0(A_k)+N_k-1} = \mathrm{Dir}_k\big(\alpha P_0(A_1) + N_1, \dots, \alpha P_0(A_k) + N_k\big).$$

□

Note that the notational convention adopted for the parameters of the Dirichlet process is interesting here: it provides the simple 'product' parameter $\alpha P_0 + \sum_{i=1}^n \delta_i$, which can also be decoupled into a posterior precision parameter α_n and a posterior base measure G_n given by

$$\alpha_n = \alpha + n,$$
$$G_n = \frac{\alpha}{\alpha + n} P_0 + \frac{1}{\alpha + n} \sum_{i=1}^n \delta_{X_i}.$$

This posterior conjugacy, in turn, provides a simple form of the predictive distribution, that is the distribution of a new observation conditional on the data. This predictive is referred to as the Pólya urn, or Blackwell-MacQueen urn.

Proposition 4.2 (Predictive distribution, Pólya urn, [7])**.** *In the model of Proposition 4.1, the predictive distribution for a new observation X_{n+1} is given by*

$$X_{n+1}|X_1, \dots, X_n \sim \frac{\alpha}{\alpha + n} P_0 + \frac{1}{\alpha + n} \sum_{i=1}^n \delta_i.$$

Proof. This property is a result of taking the expected value of the posterior given in Proposition 4.1. □

Such a predictive distribution induces ties in the observations with positive probability. More precisely, n observations sampled from a DP induce a partition of the integers $1, \dots, n$. The distribution of this random partition is called the Chinese restaurant process[3]. This

[3] According to [1], the restaurant analogy is due to Jim Pitman and Lester Dubins.

culinary metaphor describes the random partition induced by the DP as follows. Customers join a populated table with probability $n_j/(\alpha+n)$, where n_j denotes the number of clients already sitting around the table or sit at new table with probability $\alpha/(\alpha+n)$.

Proposition 4.3 (Chinese restaurant process, [2]). *A random sample $X_{1:n}$ from a DP with precision parameter α induces a partition of $\{1,\dots,n\}$ into k sets of sizes $n_1,\dots,n_k$ with probability*

$$p(n_1,\dots,n_k) = p(\{n_1,\dots,n_k\}) = \alpha^k \frac{\Gamma(\alpha)}{\Gamma(\alpha+n)} \prod_{j=1}^{k} \Gamma(n_j).$$

Proof. We will use the Pólya urn schema slightly changed by using $n_1,\dots,n_k$

$$\mathrm{P}(X_{n+1}|X_{1:n}) = \frac{\alpha}{\alpha+n} P_0 + \frac{1}{\alpha+n} \sum_{j=1}^{k} n_j \delta_{X_j^*}.$$

By exchangeability, the distribution of $\{n_1,\dots,n_k\}$ does not depend on the order of the observations. Let's compute $p(n_1,\dots,p_k)$ as the probability of one draw where the first table consists of first n_1 observations etc.

To proceed, let us use Pólya urn scheme: we denote $\bar{n}_j = \sum_{i=1}^{j} n_i$ and hence $\bar{n}_k = n$, the total number of observations. We can observe the following pattern: first ball opens new table, following $n_j - 1$ ones fill in that table and so forth. That quantity can be rewritten as

$$\frac{\alpha^k}{\alpha(\alpha+1)\dots(\alpha+n-1)} \prod_{j=1}^{k} (n_j-1)!,$$

where one can rewrite both terms using Gamma function $\Gamma(x) = \int_0^\infty u^{x-1}e^{-u}du$:

$$\alpha(\alpha+1)\dots(\alpha+n-1) = \frac{\Gamma(\alpha+n)}{\Gamma(\alpha)},$$

and $(n_j-1)! = \Gamma(n_j)$.

Note that for ordered partitions we have

$$\bar{p}(n_1,\dots,n_k) = \frac{p(n_1,\dots,n_k)}{k!}.$$

□

The following lines of code sample observations from a Dirichlet process with a base measure in argument, which is also interpreted as the color distribution in the Pólya urn scheme.

```
polya_urn_model <- function(base_measure, N_ball, alpha) {
  balls <- c()
  for (i in 1:N_ball) {
    if (runif(1) < alpha / (alpha + length(balls))) {
      # Add a new ball color.
      new_color <- base_measure()
```

```
      balls <- c(balls, new_color)
    } else {
      # Pick out a ball from the urn, and add back a
      # ball of the same color.
      ball <- balls[sample(1:length(balls), 1)]
      balls <- c(balls, ball)
    }
  }
  balls
}
```

This is applied to sample 10 observations from a Dirichlet process with a Gaussian $N(0, 1)$ base measure, which is also interpreted as the color distribution in the Pólya urn scheme, and precision parameter varying from 1, 10 to 100.

```
N_ball <- 10
# with alpha = 1
polya_sample <- polya_urn_model(function() rnorm(1), N_ball, 1)
rev(sort(table(polya_sample)))

## polya_sample
## -1.04779556205753 0.353418450483626
##                 8                 2

# with alpha = 10
polya_sample <- polya_urn_model(function() rnorm(1), N_ball, 10)
rev(sort(table(polya_sample)))

## polya_sample
##  -0.262911838454736   -0.85109415351288    2.01291826160114
##                   2                   2                   1
##    1.73642130785405   0.546212584065414   0.264154437076562
##                   1                   1                   1
##   0.191324198334465 0.00291702961916057
##                   1                   1

# with alpha = 100
polya_sample <- polya_urn_model(function() rnorm(1), N_ball, 100)
rev(sort(table(polya_sample)))

## polya_sample
##  0.977248376259983  0.766722814230755  0.597919221042819
##                  1                  1                  1
##  0.466119538300814  0.460876928395281  0.304868236499072
##                  1                  1                  1
##  0.157307784983287 -0.269715738859888 -0.652901580743433
##                  1                  1                  1
##  -1.41731999466404
##                  1
```

These experiments illustrate that large values of the mass parameter α tend to produce a larger number of distinct values in a sample of a given size. For instance, the probability that all $n = 10$ observations are distinct is equal to

$$\mathbb{P}(X_1, \ldots, X_n \text{ are pairwise distinct}) = \alpha^n \frac{\Gamma(\alpha)}{\Gamma(\alpha + n)},$$

which is approaching 1 as α grows, for n fixed.

In our case, this probability is respectively equal to 3e-07, 3e-02 and 0.6 for α equal to 1, 10 and 100, for $n = 10$.

4.3 Stick-breaking representation

The Dirichlet process is a discrete random probability measure which can be represented as a convex combination of infinitely many Dirac masses,

$$G = \sum_{k=1}^{\infty} \pi_k \delta_{\phi_k}.$$

The stick-breaking representation, due to [18], provides a constructive way of building the weights $(\pi_k)_k$ of the Dirichlet process. This is done by sequentially breaking a stick of initial unit length, into pieces whose lengths correspond to the $(\pi_k)_k$. More specifically, we require independent and identically distributed (iid) random variables $V_k \sim \text{Beta}(1, \alpha)$. The first weight π_1 corresponds to V_1. This leaves a piece of length $1 - V_1$, which is broken at V_2 in order to define $\pi_2 = V_2(1 - V_1)$. And sequentially, the same procedure is applied to the remaining part, which equals $(1 - V_1)(1 - V_2)$ at this second step. It is easy to see that after k steps, one defines $\pi_k = V_k(1 - V_1) \cdots (1 - V_{k-1})$, and the remaining piece has length $(1 - V_1) \cdots (1 - V_k)$. The representation is completed by assuming iid draws from the base measure G_0 for the locations ϕ_k, independent from the V_k.

Theorem 4.2 (Stick-breaking representation, [18]). *Let* $V_1, V_2, \ldots \overset{iid}{\sim} \text{Beta}(1, \alpha)$ *and* $\phi_1, \phi_2, \ldots \overset{iid}{\sim} G_0$ *be independent random variables. Define*

$$\begin{aligned} \pi_1 &= V_1, \\ \pi_k &= V_k(1 - V_1) \cdots (1 - V_{k-1}), \text{ for any } k \geq 2. \end{aligned}$$

Then $G = \sum_{k=1}^{\infty} \pi_k \delta_{\phi_k} \sim \text{DP}(\alpha G_0)$.

Proof. We provide a sketch of proof of this result in two steps. First, to show that the remaining stick length at step k, $(1 - V_1) \cdots (1 - V_k)$, converges to zero as $k \to \infty$. This ensures that the weights vector lives in the unit simplex, and in turn that the measure $\sum_{k=1}^{\infty} \pi_k \delta_{\phi_k}$ is a probability measure. Second, to use the stick-breaking construction to show that the defined G satisfies the distributional equation

$$G \overset{d}{=} V \delta_{\phi} + (1 - V)G, \tag{2}$$

where $V \sim \text{Beta}(1, \alpha)$ and $\phi \sim G$, independently, whose only solution turns out to be the Dirichlet process, by properties of the Dirichlet distribution. □

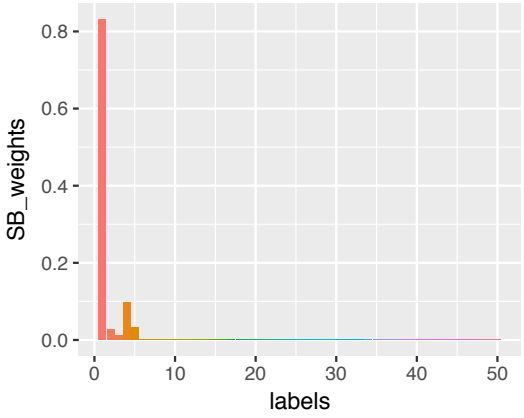
The following code implements the sampling of the first `num_weights`=50 weights of a DP. These are then plotted in the order they appear in the stick-breaking construction (left) as well as indexed by their corresponding location (right). Colors indicate the stick-breaking order of appearance. Note that the weights are not necessarily strictly decreasing, but only stochastically decreasing. This means they are decreasing in expectation, as can be easily checked:

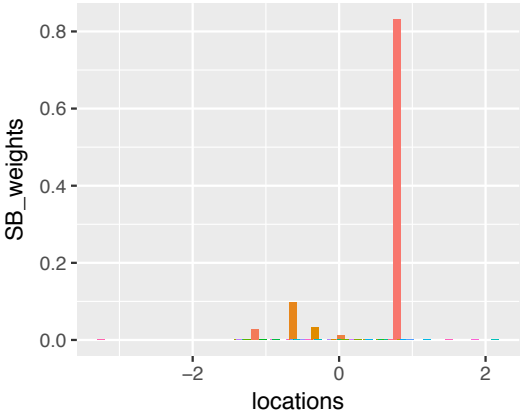
$$\mathrm{E}(\pi_k) = \mathrm{E}(V_k(1-V_1)\cdots(1-V_{k-1})) = \mathrm{E}V_k\mathrm{E}(1-V_1)\cdots\mathrm{E}(1-V_{k-1}) = \frac{1}{\alpha+1}\left(\frac{\alpha}{\alpha+1}\right)^{k-1}.$$

```
stick_breaking_process = function(num_weights, alpha) {
  betas = rbeta(num_weights, 1, alpha)
  remaining_stick_lengths = c(1, cumprod(1 - betas))[1:num_weights]
  weights = remaining_stick_lengths * betas
  weights
}
```

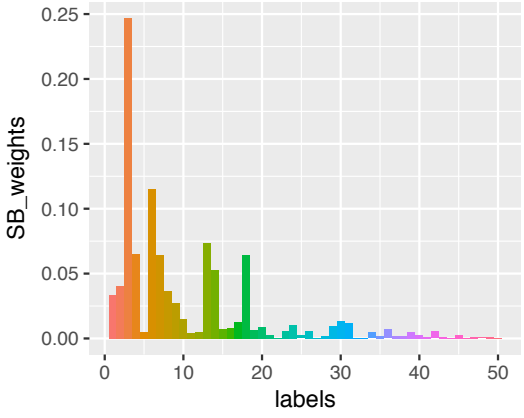
```
num_weights <- 50
draw_stick_breaking <- function(alpha) {
  labels <- 1:num_weights
  locations <- rnorm(num_weights)
  SB_weights <- stick_breaking_process(num_weights, alpha)
  df <- data.frame(labels, locations, SB_weights)
  order_plot <-
        ggplot(df, aes(labels, SB_weights, fill = as.factor(labels))) +
    geom_bar(stat = "identity") +
    theme(legend.position="none")
  location_plot <-
        ggplot(df, aes(locations, SB_weights, fill = as.factor(labels))) +
    geom_bar(stat = "identity", width = .1) +
    theme(legend.position="none")
  grid.arrange(order_plot, location_plot, ncol = 2)
}
```

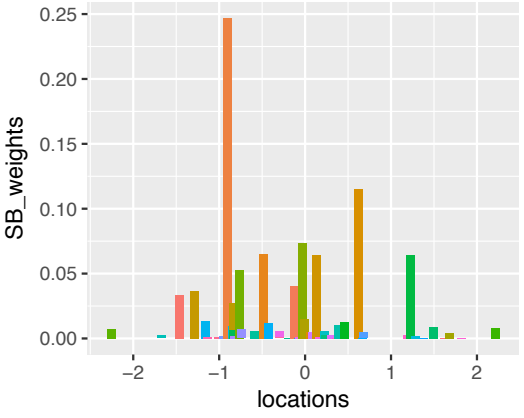
```
draw_stick_breaking(1)
```



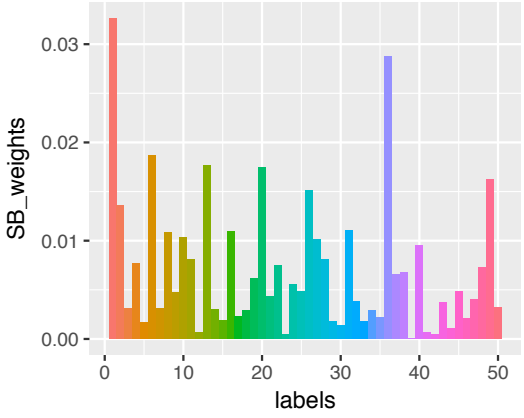


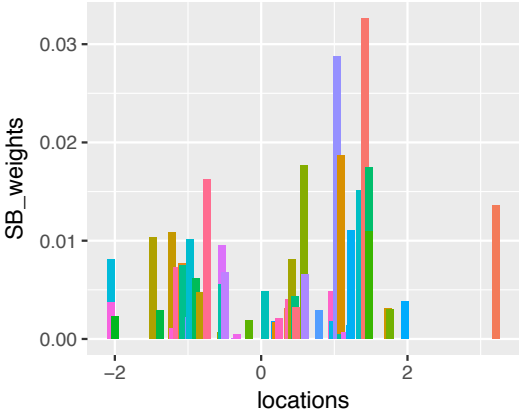

```
draw_stick_breaking(10)
```






```
draw_stick_breaking(100)
```



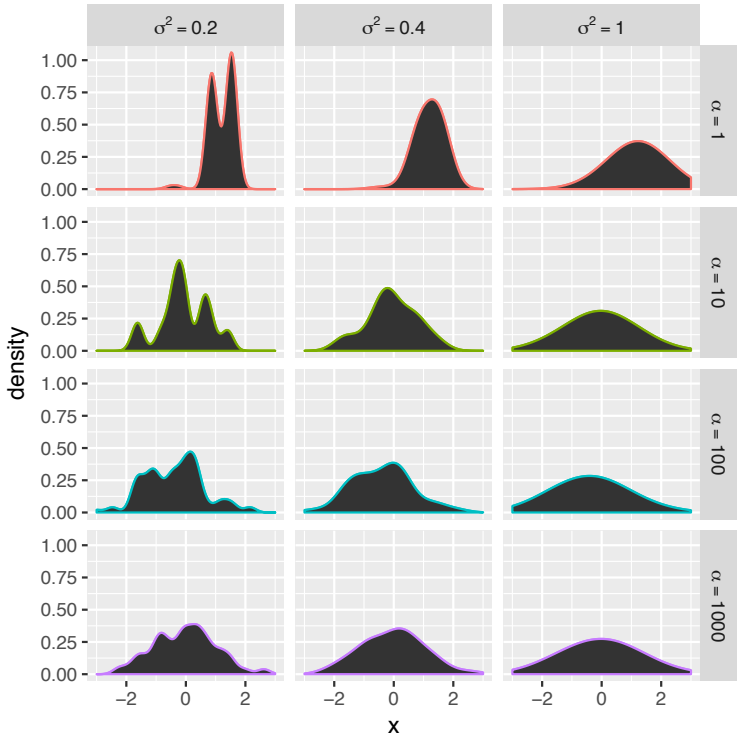


4.4 Dirichlet process mixture models

Combining the model-based clustering approach described in Section 3 with the DP, we are ready to work with nonparametric mixture models, where the Dirichlet process can be used as a prior distribution on the mixing probability measure [12].

The following code shows densities draws from DP mixture densities with $N(0,1)$ base measure, varying mass parameter in $\{1, 10, 100, 1000\}$, and centered Gaussian kernel with varying variance σ^2 in $\{0.2, 0.4, 1\}$. Enlarging α tends to produce flatter densities, while reducing σ^2 tends to produce more irregular ones.

```
alpha_vect <- c(1, 10, 100, 1000)
N_urns <- 3
sigma2 <- c(1,.4,.2)
N_draws <- 100
N_xaxis <- 200
x_axis <- seq(-3, 3, length = N_xaxis)
result <- NULL
  for (alpha in alpha_vect) {
    PU <- polya_urn_model(function() rnorm(1), N_draws, alpha)
    for (u in 1:N_urns) {
    res <- mapply(function(mean) dnorm(x_axis, rep(mean, N_xaxis),
                                       rep(sigma2[u], N_xaxis)), PU)
    res <- apply(res, 1, mean)
    new_draw <- cbind(res, x_axis, alpha, sigma2[u])
    result <- rbind(result, new_draw)
  }
}
result <- as.data.frame(result)
names(result) <- c("density", "x", "alpha", "sigma2")
DP_mixt <- qplot(data = result, y = density, x = x,
                 geom = c("line", "area")) +
  facet_grid(alpha ~ sigma2, labeller =
               label_bquote(rows = alpha == .(alpha),
                            cols = sigma^2 == .(sigma2))) +
  aes(color = as.factor(alpha)) +
  theme(legend.position = "none")
DP_mixt
```



5 Application to clustering of globulars of our galaxy

In this last section, we show how Dirichlet process mixtures can be applied to perform clustering of globulars of our galaxy.

5.1 Markov chain Monte Carlo sampling

Inference in R for DPM is implemented in packages including `DPpackage` [11], `BNPdensity` [5]. Here we present a recent package called `BNPmix` [4]. It uses C++ code with the `Rcpp` package, which makes the implementation efficient. The package is available on GitHub[4] and can be installed with `devtools` as follows:

```
library(devtools)
install_github("rcorradin/BNPmix")
```

This package uses d-dimensional location-scale Dirichlet process mixture of Gaussians. This means that it assumes a Gaussian kernel $p(X|\phi) = \Phi_d(X|\boldsymbol{\mu}, \boldsymbol{\Sigma})$, where the component specific parameter θ is composed of the mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. The base measure G_0 is chosen as an independent product of a Gaussian and an inverse-Wishart distributions

$$G_0(\mathrm{d}\boldsymbol{\mu}, \mathrm{d}\boldsymbol{\Sigma}; \boldsymbol{\pi}) = N_d(\mathrm{d}\boldsymbol{\mu}; \mathbf{m}_0, \mathbf{B}_0) \times IW(\mathrm{d}\boldsymbol{\Sigma}; \nu_0, \mathbf{S}_0), \tag{3}$$

[4] At the url: https://github.com/rcorradin/BNPmix.

The hyperparameters $(\mathbf{m}_0, \mathbf{B}_0, \nu_0, \mathbf{S}_0)$ are typically expressed by an empirical Bayes approach. See [4] for details. Additionally, the precision parameter α is endowed with a gamma distribution with shape t_1 and scale t_2.

Posterior inference is carried out by a Markov chain Monte Carlo (MCMC) algorithm. The function is called `DPmixMulti`, and it takes the following arguments:

- the `data`;
- a `grid` of the sample space, with the same dimension as that of the data, on which the density is to be evaluated;
- `MCMC_param`, a list of parameters for the MCMC including number of simulations `nsim` and burn-in `nburn`;
- `starting_val`, a list containing the initial values for the components;
- `params`, hyperparameters $(\mathbf{m}_0, \mathbf{B}_0, \nu_0, \mathbf{S}_0)$ for the base measure G_0 of the DPM and (t_1, t_2) for the hyperprior on the precision parameter α. When omitted, those parameters are set in an empirical Bayes way by default.

A grid to evaluate the densities is defined as follows, spanning all data points, plus/minus 10% with respect to the extreme data points.

```
data <- spectra[, c(2,3,4)]
grid <- expand.grid(seq(range(data[,1])[1]-.1 * diff(range(data[,1])),
                        range(data[,1])[2]+.1 * diff(range(data[,1])),
                        length.out = 40),
                    seq(range(data[,2])[1]-.1 * diff(range(data[,2])),
                        range(data[,2])[2]+.1 * diff(range(data[,2])),
                        length.out = 40),
                    seq(range(data[,3])[1]-.1 * diff(range(data[,3])),
                        range(data[,3])[2]+.1 * diff(range(data[,3])),
                        length.out = 40))
```

We can now run the MCMC function `DPmixMulti`:

```
MCMC_param <-  list(nsim = 10^4, nburn = 5*10^3)
MCMC_output <- DPmixMulti(data = as.matrix(data),
                          grid = grid,
                          MCMC_param = MCMC_param)
```

The output `MCMC_output` of this function is a list of three objects:

- `distribution` the estimated (posterior mean) distribution evaluated on the grid.
- `result_cluster` a matrix, each row is an iteration, each column an observation, each entry is a latent component, thus this contains clusterings for each iteration of the chain.
- `result_theta` a vector, the value of DP precision parameter α over the iterations.

The posterior clustering information is enclosed in the second object listed above, that we can save in a matrix `MCMC_clustering`:

```
MCMC_clustering <- MCMC_output[[2]]
```

5.2 Clustering estimation

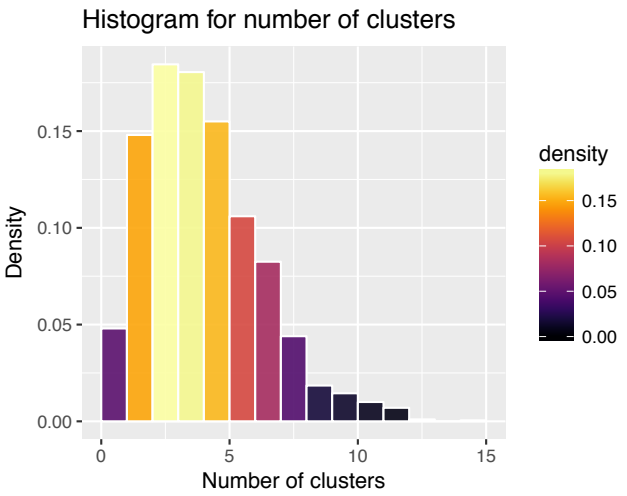
It should be noted that a DPM assumes a priori an infinite number of components in the mixture (see for instance the stick-breaking representation). As such, it is a misspecified model in essence when it comes to estimating a fixed clustering, that is a clustering which is not deemed to grow as the sampled data grows. However, in the spirit of the famous quote by George Box *"All models are wrong but some are useful"*, misspecified models can be used as long as they provide insightful results.

Clustering with DPM can be done in different ways. Indeed, the MCMC output we have obtained so far essentially consists in many different clusterings of the data, among which one needs to choose a 'best' clustering for some decision rule.

A first approach consists in looking at the number of components, an estimator of which can be obtained with the mode a posterior (MAP) for instance. However, the posterior distribution of the number of components in a DPM turns out to be *inconsistent* under some model specifications [14]. Posterior consistency is a theoretical property of a posterior distribution: when more and more data are collected from some fixed data distribution, we say that the posterior distribution of some parameter is consistent if it converges to a point mass at the true fixed value of this parameter. The inconsistency property provided by [14] is as follows. Assume a DPM model with standard Gaussian base measure, precision parameter equal to one, and standard Gaussian kernel. Suppose data are generated from a standard Gaussian. This means that the data are sampled from a very simple mixture which admits a single component. [14] study the posterior distribution of the number of clusters in this situation, and they prove that it does not concentrate to a point mass at one, thus proving posterior inconsistency.

Let us plot the histogram of the number of clusters:

```
MCMC_number_cluster = apply(X = MCMC_clustering, MARGIN = 1, FUN = max)
df_cluster <- data.frame(nb = MCMC_number_cluster)
c <- ggplot(df_cluster, aes(nb,..density..))
c + geom_histogram(breaks=(0:max(df_cluster)),
                   aes(fill=..density..)
                   ,
                   col="white",
                   alpha = .9
                   ) +
  scale_fill_viridis(option = "inferno") +
  labs(title="Histogram for number of clusters") +
  labs(x="Number of clusters", y="Density")
```



The MAP estimator for the number of clusters is three, obtained as:

```
which.max(table(MCMC_number_cluster))

## 3
## 3
```

Note that the above histogram is also quite in agreement with an estimate of four clusters.

Stepping back, recall that a Bayes estimator is obtained from a formal decision theory rule: given a loss function, a Bayes estimator minimizes the posterior expected loss. For instance, with Euclidean parameter spaces,

- the L^2, squared loss provides the posterior mean,
- the L^1, absolute loss provides the posterior median,
- the $0-1$ loss provides the mode a posteriori (MAP).

We focus now on a loss function L on clusterings. The posterior expected loss of clustering c', denoted by $L(c')$, is obtained by averaging the loss with respect to posterior distribution, over the set of all partitions of the integers $1, \ldots, n$ denoted by $\mathcal{A}_n$

$$L(c') = \sum_{c \in \mathcal{A}_n} L(c, c') p(c|\mathbf{x}),$$

and the decision is taken by choosing the best partition

$$\hat{c} = \arg \min_{c' \in \mathcal{A}_n} \sum_{c \in \mathcal{A}_n} L(c, c') p(c|\mathbf{x}).$$

Several losses have been considered in the literature:

- 0-1 loss [17],
- Binder loss [8],
- Variation of information (VI) [19].

The 0-1 loss gives rise to the MAP estimator:

$$L_{0-1}(c') = \sum_{c \in \mathcal{A}_n} L_{0-1}(c, c') p(c|\mathbf{x}) = \sum_{c \in \mathcal{A}_n,\, c \neq c'} p(c|\mathbf{x}) = 1 - p(c'|\mathbf{x})$$

which is to say that the expected loss of c' is all the posterior mass except that of c'. So that it is easily minimized at the value c' which has maximum posterior weight:

$$\hat{c} = \arg \min_{c' \in \mathcal{A}_n} L_{0-1}(c') = \arg \max_{c' \in \mathcal{A}_n} p(c'|\mathbf{x}) := MAP.$$

Negative results by [17] show that the mode a posteriori (MAP) is inconsistent.

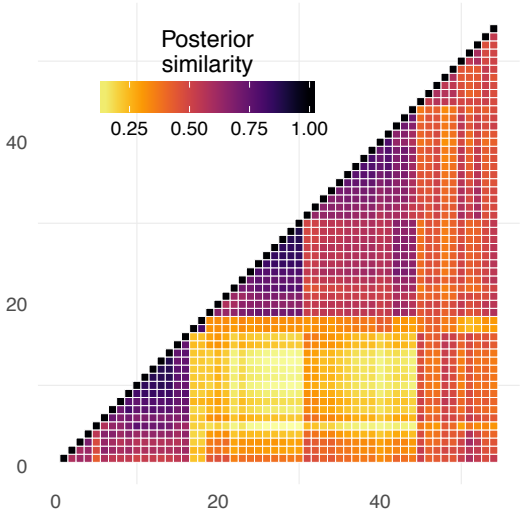
Instead of the 0-1 loss, one can resort to the Variation of information (VI) which was devised by [13] for clustering comparison. It stems from information theory, and compares information (in terms of Shannon entropy H) in two clusterings with information shared between the two clusterings (I), see [13] for details:

$$\mathrm{VI}(c, \widehat{c}) = H(c) + H(\widehat{c}) - 2I(c, \widehat{c}).$$

Estimation under the Variation of information loss function was recently studied by [19], with the interesting finding that this loss function tends to reduce the over-estimation of the number of cluster that is commonly obtained under Binder loss, for instance by [8]. See also [3] for a further comparison between Binder and VI on varying sample sizes. We implement the Variation of information approach by using the mcclust.ext R package developed by [19]. This requires to compute the posterior similarity matrix associated to the MCMC output, that is a matrix whose entries represent the posterior probability that two observations are clustered together (or rather, a Monte Carlo approximation of it).

```
posterior_similarity_matrix = comp.psm(MCMC_clustering)
```

It can be represented as follows, where darker colors depict a higher posterior probability of shared clustering.



The next step consists in resolving the minimization problem

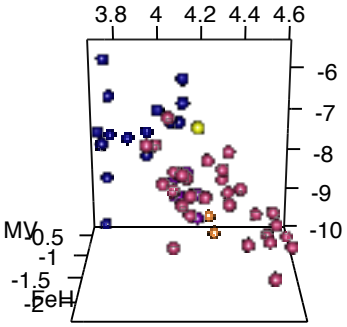
$$\hat{c} = \arg \min_{c' \in \mathcal{A}_n} \sum_{c \in \mathcal{A}_n} \mathrm{VI}(c, c') p(c|\mathbf{x}).$$

Of course, this optimization is only done approximately by the mcclust.ext R package, which scans the MCMC output and a neighbouring region of it in order to choose the best clustering.

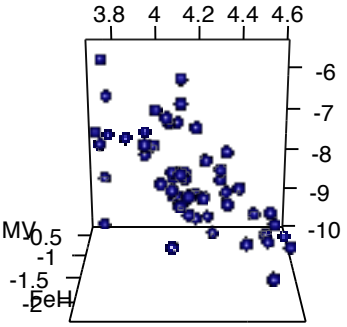
```
clustering_Binder=minbinder.ext(posterior_similarity_matrix, method = "greedy")
table(clustering_Binder$cl)

##
##  1  2  3  4  5
## 16  6 29  2  1

clustering_VI=minVI(posterior_similarity_matrix, method = "greedy")
table(clustering_VI$cl)

##
##  1
## 54
```

Thus, the Binder loss produces an estimated clustering with 5 groups, whereas under the Variation of information, all the observations are gathered into a single cluster of size 54. The interpretation of these results is that under the conditions of the assumed Dirichlet process mixture model, data are not deemed heterogeneous enough to justify multiple clusters under VI loss, whereas they do under Binder loss. The estimated clusterings and the ones obtained by [10] are represented below in a 3D-like fashion by using the `rgl` package function `plot3d`.

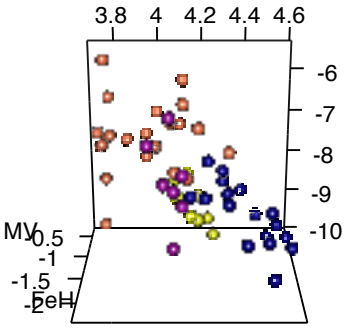
```
subtitle = c("Binder loss, 5 groups", "VI loss, 1 group",
"Fraix-Burnet (2009), 4 groups", "Fraix-Burnet (2009), 3 groups")
clustering = cbind(clustering_Binder$cl,
                   clustering_VI$cl,
                   Grp3c,
                   Grp4c)
for(i in 1:4){
        cl = clustering[, i]
        plot3d(logTe, FeH, MV,
               type="s", size=2, col=plasma(max(cl))[cl],
               box = FALSE, sub = subtitle[i])
}
```



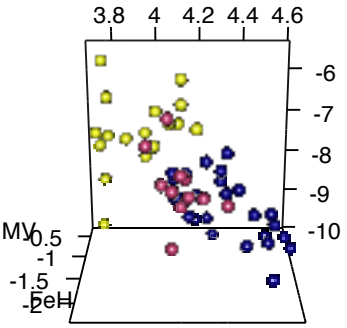
logTe
Binder loss, 5 groups



logTe
VI loss, 1 group



logTe
Fraix-Burnet (2009), 4 groups



logTe
Fraix-Burnet (2009), 3 groups

Acknowledgements

I would like to thank Michał Lewandowski for helping with typing parts of this chapter and with merging the Globular data by [10] with the Globular table that can be found on Wikipedia, see Footnote 2.

References

[1] Aldous, D. J. (1985). Exchangeability and related topics. In *École d'Été de Probabilités de Saint-Flour XIII—1983*, pages 1–198. Springer.

[2] Antoniak, C. E. (1974). Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems. *The Annals of Statistics*, pages 1152–1174.

[3] Arbel, J., Corradin, R., and Łewandowski, M. (2018a). Discussion of "Bayesian Cluster Analysis: Point Estimation and Credible Balls", by Wade and Ghahramani. *Bayesian Analysis*.

[4] Arbel, J., Corradin, R., and Nipoti, B. (2018b). Dirichlet process mixtures under affine transformations of the data. *Submitted*.

[5] Barrios, E., Lijoi, A., Nieto-Barajas, L. E., and Prünster, I. (2013). Modeling with normalized random measure mixture models. *Statistical Science*, 28(3):313–334.

[6] Bernardo, J. M. and Smith, A. F. (2009). *Bayesian theory*, volume 405. Wiley.

[7] Blackwell, D. and MacQueen, J. B. (1973). Ferguson distributions via Pólya urn schemes. *The Annals of Statistics*, pages 353–355.

[8] Dahl, D. B. (2006). Model-based clustering for expression data via a Dirichlet process mixture model. *Bayesian inference for gene expression and proteomics*, pages 201–218.

[9] Ferguson, T. (1973). A Bayesian analysis of some nonparametric problems. *The Annals of Statistics*, 1(2):209–230.

[10] Fraix-Burnet, D., Davoust, E., and Charbonnel, C. (2009). The environment of formation as a second parameter for globular cluster classification. *Monthly Notices of the Royal Astronomical Society*, 398:1706–1714. To appear in MNRAS.

[11] Jara, A., Hanson, T., Quintana, F., Müller, P., and Rosner, G. (2011). DPpackage: Bayesian non-and semi-parametric modelling in R. *Journal of statistical software*, 40(5):1.

[12] Lo, A. (1984). On a class of Bayesian nonparametric estimates: I. Density estimates. *The Annals of Statistics*, 12(1):351–357.

[13] Meilă, M. (2007). Comparing clusterings—an information based distance. *Journal of Multivariate Analysis*, 98(5):873–895.

[14] Miller, J. W. and Harrison, M. T. (2013). A simple example of Dirichlet process mixture inconsistency for the number of components. In *Advances in neural information processing systems*, pages 199–206.

[15] Miller, J. W. and Harrison, M. T. (2017). Mixture models with a prior on the number of components. *Journal of the American Statistical Association*, pages 1–17.

[16] Müller, P. and Mitra, R. (2013). Bayesian nonparametric inference–why and how. *Bayesian Analysis*, 8(2).

[17] Rajkowski, Ł. (2016). Analysis of MAP in CRP Normal-Normal model. *arXiv preprint arXiv:1606.03275*.

[18] Sethuraman, J. (1994). A constructive definition of Dirichlet priors. *Statistica Sinica*, 4:639–650.

[19] Wade, S. and Ghahramani, Z. (2018). Bayesian cluster analysis: Point estimation and credible balls. *Bayesian Analysis*.