Bayesian Nonparametrics: a Soft Introduction

Vanda Inácio de Carvalho

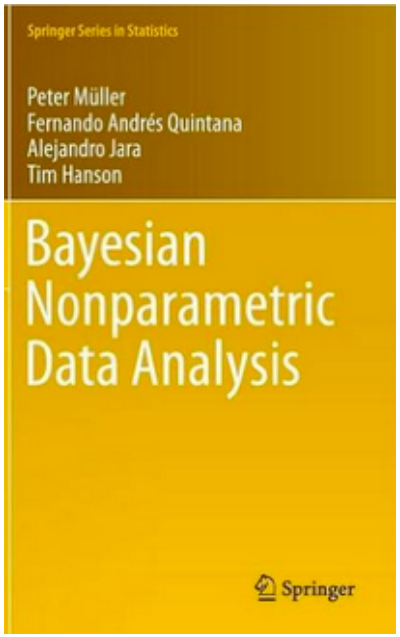
Pontificia Universidad Católica de Chile, Chile

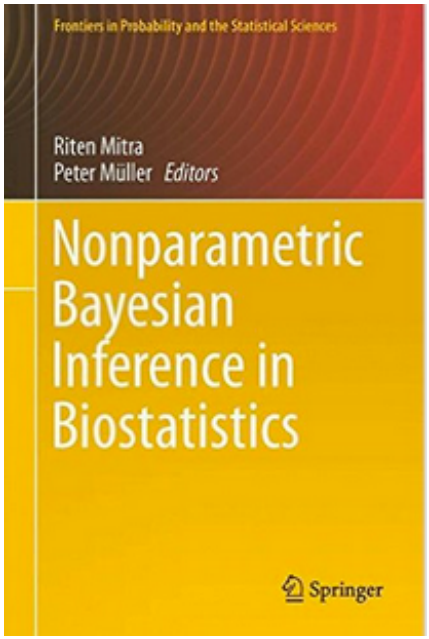
CLATSE 2016, October 26, 2016

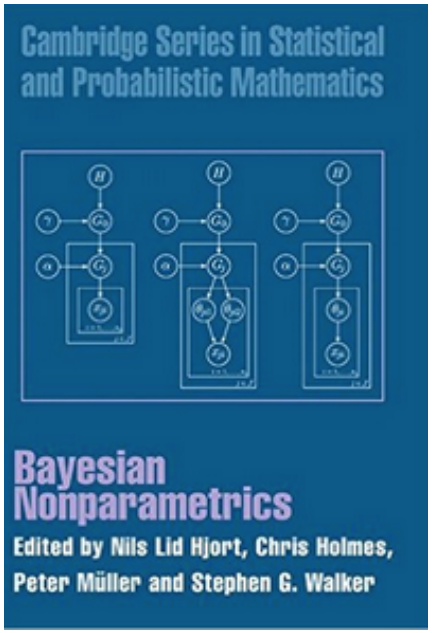
Outline

1. Introduction/motivation
2. Finite mixture models
3. Dirichlet process mixture models
4. Mixtures of finite Polya trees

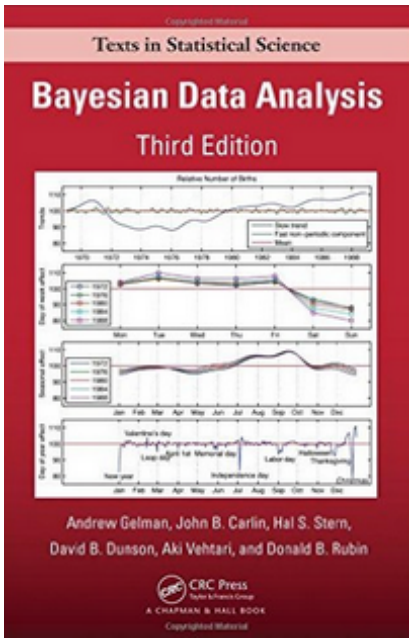
Some references (covers from Amazon.com)

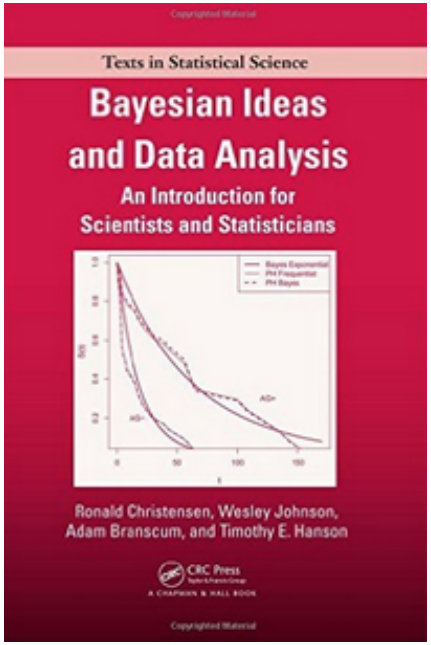






Some references (covers from Amazon.com)





Some references

- Hanson, Branscum and Johnson (2005). Bayesian nonparametric modeling and data analysis: an introduction. In *Handbook of Statistics*, Elsevier.

- Hanson, and Jara, A (2013). Surviving fully Bayesian nonparametric regression models. In *Bayesian Theory and Applications*, Oxford University Press, UK.

- Muller and Rodriguez (2013). *Nonparametric Bayesian Inference*. NSF-CBMS Regional Conference Series in Probability and Statistics, Volume 9.

  `http://projecteuclid.org/euclid.cbms/1362163742`

- Kottas and Rodriguez (2014). Unpublished lecture notes.

  `https://users.soe.ucsc.edu/ thanos/notes-1.pdf`

Some references

- Muller and Quintana (2004). Nonparametric Bayesian data analysis. *Statistical Science* **19**, 95–110.

- Muller and Mitra (2013). Bayesian nonparametric inference – why and how. *Bayesian Analysis* **8**, 269–302.

Available software

- DPpackage: Bayesian Semi- and Nonparametric Modeling in R

  `https://cran.r-project.org/web/packages/DPpackage/`

- Bayesian Regression: Nonparametric and Parametric Models

  `http://georgek.people.uic.edu/BayesSoftware.html`

Why Bayesian nonparametrics?

Motivation

- The motivation is twofold:
  1. Allowing for model flexibility.
  2. Safeguarding against model misspecification.
- Bayesian nonparametrics rely on parametric baseline models while allowing for data-driven deviations.
- So that we can see if parametric models might actually fit by embedding them in nonparametric families (sensitivity analysis).
- Because Bayesian nonparametric modeling is feasible due to modern MCMC methods.

Why Bayesian nonparametrics?

Motivation

- The goal here is to provide a rich class of statistical models for data analysis.
- Data distributions can easily be multimodal or have severe skewness.
- The normal distribution is still widely used in practice.
- But the normal family is symmetric and has only two parameters, one corresponding to location and the other corresponding to dispersion.
- Common practice is to log transform the data. But if after the log, the data still have multimodality and/or skewness, the normal distribution would not be adequate.
- Other parametric distributions are similarly constrained.

Why Bayesian nonparametrics?

Motivation

- We thus proceed to discuss broader families that allow for flexibility and robustness beyond what is achievable using parametric models.
- This goal is accomplished by setting a particular parametric class and expanding it so that there are many more possibilities included.
- In the parametric Bayesian approach, given a particular dataset, a family of models is selected.
- In the normal case, this involves location and scale parameters.
- We then select prior distributions for these two parameters. This induces a prior on the family of distributions.

Why Bayesian nonparametrics?

Motivation

- In the nonparametric case, we do the same, except that, instead of having a small number of parameters that characterize the family of distributions for the data, conceptually there is an infinite number of parameters.
- But since we live in a finite world, practically dictates that these models must be truncated to have a possibly large but finite number of parameters.
- We call them richly parametric as a result.
- Thus, nonparametric Bayesian modeling requires a prior distribution on the (potentially) infinite dimensional parameters.
- Technically, this amounts to placing a prior distribution on the space of all distribution functions.

Why Bayesian nonparametrics?

Motivation

- We consider two approaches:
  1. The first involves the specification of a Dirichlet process mixture model for the data distribution.
  2. The second approach involves the specification of a mixture of finite Polya trees prior on the space of all distribution functions.

Finite mixture models

Motivation

- We start with finite mixture models since they, to a certain extent, provide the background for Dirichlet process mixture models.
- The natural question is: why mixture models?
- In a diversity of situations, the complexity of the observed data may render the use of a single parametric distribution insufficient for data modeling.
- For example, in the context of medical tests, biomarker outcomes for some specific disease may consist of, at least, two subgroups, corresponding to mild diseased and severe diseased individuals.

Finite mixture models

Motivation

- A mixture model assumes that data can be represented by a weighted sum of distributions, with each distribution representing a proportion of the data.
- It is worth nothing that multimodality is not the sole motivation for the use of mixture models.
- For instance, skew data can also be handled by mixtures.
- Of course, for this latter case, one could use a skew distribution, but this would imply that we expect skewness in advance, while with the more general mixture model, we can handle this and other nonstandard features of the data without the need to know them in advance.

Finite mixture models

Formulation

- Throughout, we consider the particular case of mixtures of normal distributions.
- General mixtures of normal densities can approximate any continuous density (e.g., Lo 1984).
- We assume thus that the data $\mathbf{y} = (y_1, \ldots, y_n)$ follows a mixture of K normal distributions, whose density is given by

$$f(y \mid \boldsymbol{\omega}, \boldsymbol{\theta}) = \sum_{k=1}^{K} \omega_k \phi(y \mid \mu_k, \sigma_k^2), \tag{1}$$

  with $\boldsymbol{\omega} = (\omega_1, \ldots, \omega_K)$ and $\boldsymbol{\theta} = (\mu_1, \ldots, \mu_K, \sigma_1^2, \ldots, \sigma_K^2)$.

- Hence, each data point would arise from one of K mixture components, with each component having its own mean and variance.

Finite mixture models

Formulation

- Model (1) is called a location-scale mixture of normals, since both the mean (location) and variance (scale) vary across components.
- An alternative model would treat the variances across components as constants.
- Generally, location-scale mixture of normals produce more accurate results and also correspond to more realistic representations of the data.

Finite mixture models

Formulation

- Model (1) can be equivalently written as

$$f(y \mid \boldsymbol{\omega}, \boldsymbol{\theta}) = \int \phi(y \mid \mu, \sigma^2) \mathrm{d}G(\mu, \sigma^2).$$

- Here G is a discrete finite mixing distribution given by

$$G(\cdot) = \sum_{k=1}^{K} \omega_k \delta_{(\mu_k, \sigma_k^2)}(\cdot),$$

where δ_a denotes a point mass at a.

- The likelihood of the data is then

$$L(\boldsymbol{\omega}, \boldsymbol{\theta}; \mathbf{y}) = \prod_{i=1}^{n} \sum_{k=1}^{K} \omega_k \phi(y_i \mid \mu_k, \sigma_k^2),$$

which is not analytically tractable.

Finite mixture models

Data augmentation

- This issue is overcomed by data augmentation.
- To this end, consider the latent variable $z_i \in \{1, \ldots, K\}$ with $z_i = k$ indicating that observation y_i comes from component k, i.e., from the normal component with mean μ_k and variance σ_k^2.
- The mixture model can then be viewed hierarchically; the observed data y_i is modeled conditionally on z_i and z_i is also given a probabilistic specification.
- It can then be written

$$y_i \mid z_i, \boldsymbol{\theta} \overset{\text{ind.}}{\sim} \phi(y_i \mid \mu_{z_i}, \sigma_{z_i}^2),$$
$$z_i \mid \boldsymbol{\omega} \overset{\text{iid}}{\sim} \text{Mult}(\boldsymbol{\omega}).$$

- If we marginalize over z_i we recover the original mixture formulation.

Finite mixture models

Likelihood

- By introducing, $\mathbf{z} = (z_1, \ldots, z_n)$ we obtain the complete/augmented data likelihood, which is given by

$$
\begin{aligned}
L(\boldsymbol{\omega}, \boldsymbol{\theta}; \mathbf{y}, \mathbf{z}) &= \prod_{i=1}^{n} \omega_{z_i} \phi(y_i \mid \mu_{z_i}, \sigma^2_{z_i}) \\
&= \prod_{k=1}^{K} \prod_{i:z_i=k} \omega_k \phi(y_i \mid \mu_k, \sigma^2_k) \\
&= \prod_{k=1}^{K} \omega_k^{n_k} \prod_{i:z_i=k} \phi(y_i \mid \mu_k, \sigma^2_k). \qquad (2)
\end{aligned}
$$

- Here $n_k = \sum_{i=1}^{n} I(z_i = k)$ counts the number of observations allocated to component k.

Finite mixture models

Prior distributions

- The model specification is completed by specifying prior distributions for the weights, means and variances
$$(\boldsymbol{\omega}, \boldsymbol{\mu}, \boldsymbol{\sigma}^2) \sim p(\boldsymbol{\omega}, \boldsymbol{\mu}, \boldsymbol{\sigma}^2).$$
- We consider prior independence and thus
$$p(\boldsymbol{\omega}, \boldsymbol{\mu}, \boldsymbol{\sigma}^2) = p(\boldsymbol{\omega})p(\boldsymbol{\mu})p(\boldsymbol{\sigma}^2).$$
- We will make use of conjugate priors. Specifically, the weights are assigned a Dirichlet distribution
$$(\omega_1, \ldots, \omega_K) \sim \text{Dir}(\alpha_1, \ldots, \alpha_K).$$
- In turn, the means and variances are assigned normal and inverse-gamma prior distributions, respectively. That is,
$$\mu_k \sim \text{N}(a_\mu, b_\mu^2), \quad \sigma_k^2 \sim \text{IG}(a_{\sigma^2}, b_{\sigma^2}), \quad k = 1, \ldots, K.$$
- Note that placing a prior on the collection $(\{\omega_k\}, \{\mu_k, \sigma_k^2\})$ is equivalent to placing a prior on the finite mixing distribution G

Finite mixture models

Posterior inference

- By combining the likelihood in (2) with this prior specification, the joint posterior distribution is

$$p(\boldsymbol{\omega}, \boldsymbol{\theta} \mid \mathbf{y}, \mathbf{z}) \propto L(\boldsymbol{\omega}, \boldsymbol{\theta}; \mathbf{y}, \mathbf{z}) p(\boldsymbol{\omega}) \prod_{k=1}^{K} p(\mu_k) p(\sigma_k^2) \tag{3}$$

- Although the posterior distribution in (3) does not have a recognizable form, all full conditional distributions have simple conjugate forms.

- Specifically, for the mean and variance of the components we have

$$\mu_k \mid \text{else} \sim \text{N}\left(\frac{a_\mu / b_\mu^2 + \sum_{i:z_i=k} y_i / \sigma_k^2}{1/b_\mu^2 + n_k/\sigma_k^2}, \frac{1}{1/b_\mu^2 + n_k/\sigma_k^2} \right), \tag{4}$$

$$\sigma_k^2 \mid \text{else} \sim \text{IG}\left(a_{\sigma^2} + n_k/2, b_{\sigma^2} + \sum_{i:z_i=k} (y_i - \mu_k)^2/2 \right), \tag{5}$$

for $k = 1, \ldots, K$.

Finite mixture models

Posterior inference

- For $i = 1, \ldots, n$, the full conditional distribution for z_i is

$$z_i \mid \text{else} \sim \text{Mult}(\mathbf{p}_i).$$

- Here, $\mathbf{p}_i = (p_{i1}, \ldots, p_{iK})$, with

$$p_{ik} = \Pr(z_i = k \mid \boldsymbol{\omega}, \boldsymbol{\theta}, y_i) = \frac{\omega_k \phi(y_i \mid \mu_k, \sigma_k^2)}{\sum_{l=1}^{K} \omega_l \phi(y_i \mid \mu_l, \sigma_l^2)}. \tag{6}$$

- Finally, the full conditional of the weights is given by

$$\boldsymbol{\omega} \mid \text{else} \sim \text{Dir}(\alpha_1 + n_1, \ldots, \alpha_K + n_K).$$

Finite mixture models

Gibbs sampler algorithm

Algorithm

Set an initial value for $\boldsymbol{\omega}$, *and* $\boldsymbol{\theta}$, *say,* $\boldsymbol{\omega}^{(0)}$, *and* $\boldsymbol{\theta}^{(0)}$.

`for` $t = 1, \ldots, T$, `do`:

1. `for` $i = 1, \ldots, n$, *and* $k = 1, \ldots, K$ *compute posterior probabilities of membership using equation* (6).
2. `for` $i = 1, \ldots, n$, *sample the latent component indicator,*

$$z_i^{(t)} \sim Multi(\mathbf{p}_i^{(t)}).$$

3. *Conditional on* $\mathbf{z}^{(t)}$, *update* $\boldsymbol{\omega}$,

$$\boldsymbol{\omega}^{(t)} \sim Dirichlet(\alpha_1 + n_1^{(t)}, \ldots, \alpha_K + n_K^{(t)}), \quad n_1^{(t)} = \sum_{i=1}^{n} z_i^{(t)}.$$

4. *Conditional on* $\mathbf{z}^{(t)}$, *update* $\mu_k^{(t)}$ *and* $(\sigma_k^{(t)})^2$, `for` $k = 1, \ldots, K$, *from* (4) *and* (5), *respectively.*

Finite mixture models

Identifiability issues

- Due to identifiability issues, such as the so-called label switching problem (Jasra et al. 2005), it makes difference whether there is interest in making inferences about the mixture component-specific parameters and clustering.
- The label switching problem (also known as label ambiguity) refers to the fact that there is nothing in the likelihood to distinguish mixture component k from mixture component k'.
- Permuting the K labels in any of $K!$ ways results in the same model for the data.

Finite mixture models

Identifiability issues

- As a concrete example, in the $K = 2$ case, consider

$$p_1 = 0.3, \quad \mu_1 = 1, \quad p_2 = 0.7, \quad \mu_2 = 1.5, \quad \sigma_1^2 = \sigma_2^2 = 1, \qquad \text{(Scenario A)}.$$

- Then, the model is equivalent to one with

$$p_1 = 0.7, \quad \mu_1 = 1.5, \quad p_2 = 0.3, \quad \mu_2 = 1, \quad \sigma_1^2 = \sigma_2^2 = 1, \qquad \text{(Scenario B)}.$$

- If one is only interested in density estimation, then everything is fine, because as illustrated in the example

$$\begin{aligned} f_{\text{A}}(y) &= 0.3\phi(y \mid 1, 1) + 0.7\phi(y \mid 1.5, 1) \\ &= 0.7\phi(y \mid 1.5, 1) + 0.3\phi(y \mid 1, 1) = f_{\text{B}}(y). \end{aligned}$$

- Usually, imposing $\mu_1 < \mu_2 < \cdots < \mu_K$ and using informative priors helps to mitigate the identifiability issues.

Finite mixture models

How to choose K?

- A drawback of finite mixture models is that we must choose the number of components K, which is a non-trivial task in general.
- One possibility is placing a prior on K, which leads to a model that changes dimension (number of parameters) with K.
- One approach to fit such model is to use reversible jump type of algorithms, but these type of algorithms tend to be difficult to implement efficiently in practice.

Finite mixture models

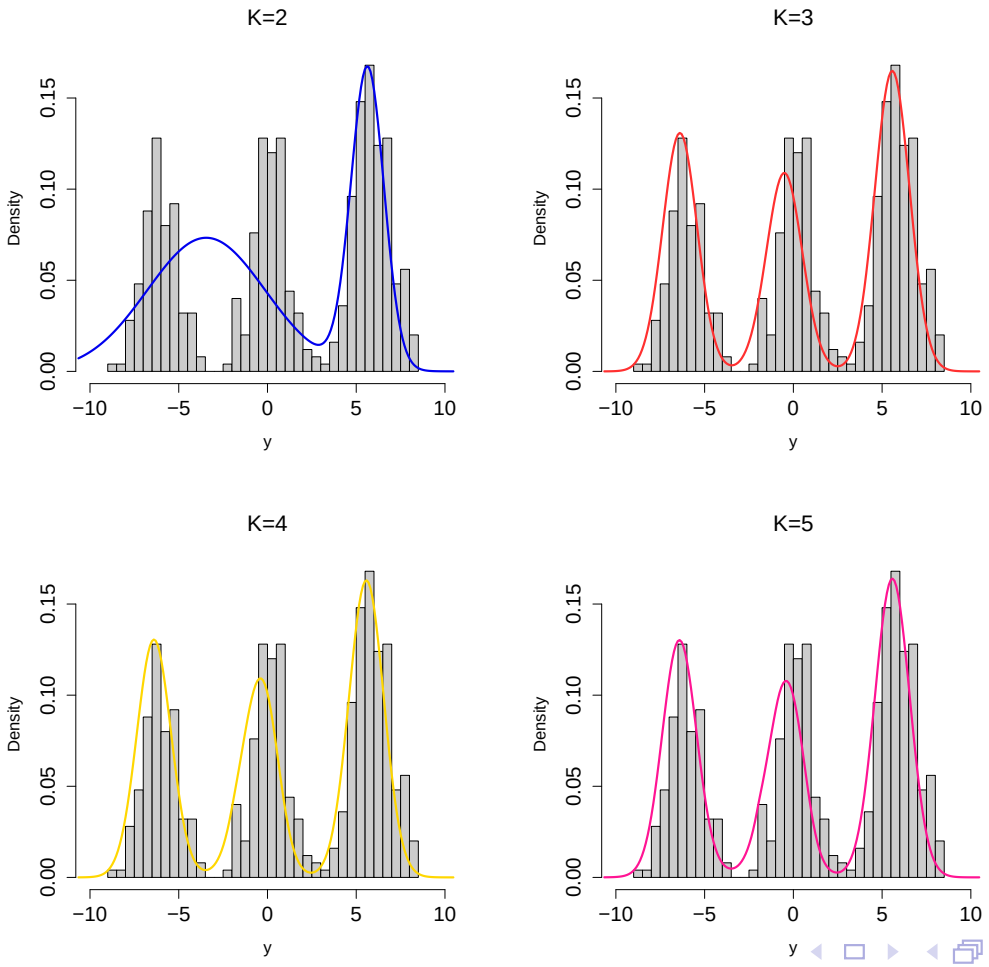
How to choose K?

- Another possibility is to fit the model for different values of K and assess the adequacy of the fit, through the LPML criterion, for instance.
- Placing a K too small is much worst than placing a K too large.
- For instance, one cannot get a trimodal density out of a mixture of two components but one can essentially ignore extra mixture terms and get bimodal densities from mixtures of three or more components.
- Marin and Robert (2013) contains a good discussion of Bayesian finite mixture models.

Finite mixture models

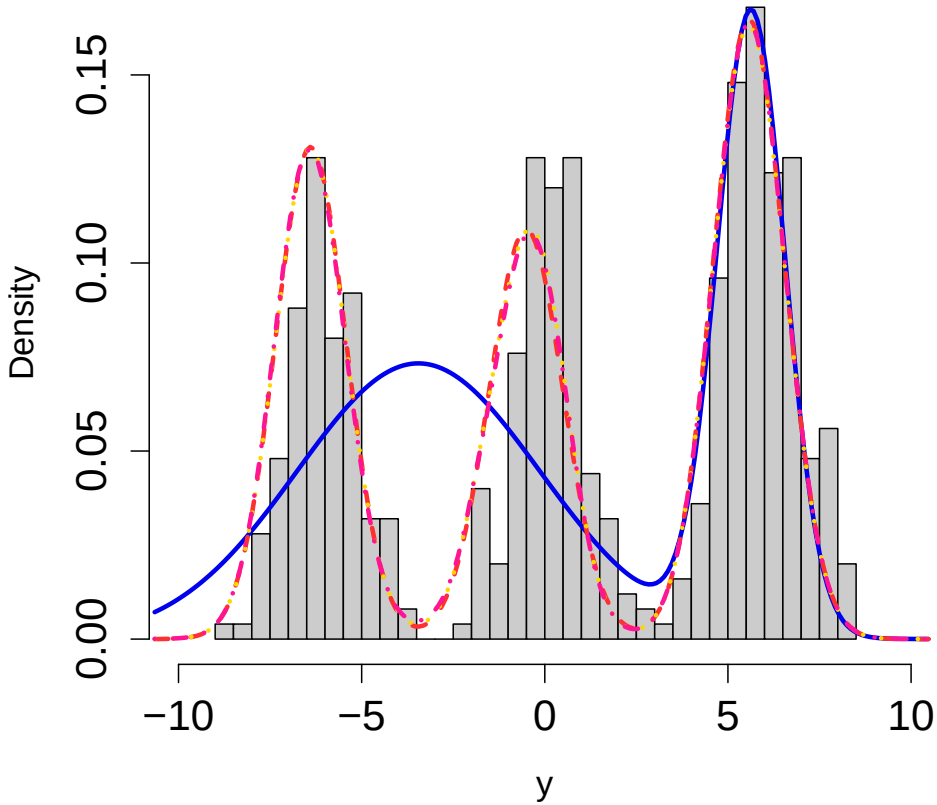
Example

True data generating distribution: $0.3\phi(y \mid -6, 1) + 0.3\phi(y \mid 0, 1) + 0.4\phi(y \mid 6, 1)$, $n = 500$.



Finite mixture models

Example



Dirichlet process mixtures

DP prior

- Another alternative is to use a DP prior (Ferguson 1973, 1974) for the mixing distribution G, resulting in a Dirichlet process mixture (DPM) model.
- The DP prior for G, on one hand, offers the theoretical advantage of having full weak support on all mixing distributions and, on the other hand, the practical advantage of automatically determining the number of components that best fits a given dataset.
- The resulting model is

$$f(y) \equiv f(y \mid G) = \int k(y \mid \boldsymbol{\theta}) \mathrm{d}G(\boldsymbol{\theta}), \quad G \sim \mathrm{DP}(\alpha, G_0).$$

- In the context of DPM of normals, $k(y \mid \boldsymbol{\theta}) = \phi(y \mid \mu, \sigma^2)$.
- The full weak support means that, under mild conditions, any distribution with the same support as G_0 can be well approximated weakly by a DP.

Dirichlet process mixtures

DP prior

- But what does it mean $G \sim \text{DP}(\alpha, G_0)$?
- The DP was originally defined by Ferguson (1973, 1974).
- The DP generates random probability measures (random distributions), G, defined on some sigma-algebra (collection of subsets) of a sample space Ω, such that the distribution of any finite partition of Ω is a Dirichlet distribution.
- That is, $G \sim \text{DP}(\alpha, G_0)$ if for any k and any partition $B_1, \ldots, B_k$ of Ω, then

$$[G(B_1), \ldots, G(B_k)] \sim \text{Dir}[\alpha G_0(B_1), \ldots, \alpha G_0(B_k)].$$

Dirichlet process mixtures

DP prior

- The definition of the Dirichlet process and the properties of the Dirichlet distribution imply that for any subset B of Ω

$$G(B) \sim \text{Beta}(\alpha G_0(B), \alpha(1 - G_0(B))).$$

- Thus,

$$\mathbb{E}\{G(B)\} = G_0(B), \quad \text{Var}\{G(B)\} = \frac{G_0(B)(1 - G_0(B))}{\alpha + 1}.$$

- Hence, from the above, we have
  - G is centered on G_0 (G_0 is also referred as baseline distribution).
  - $\alpha > 0$ is a precision parameter controlling the variance of the process. If α is large, G is highly concentrated around G_0.
- Therefore, the DP prior is centered on a parametric model through the specification of G_0, while allowing α to control uncertainty in this choice.

Dirichlet process mixtures

Stick-breaking representation

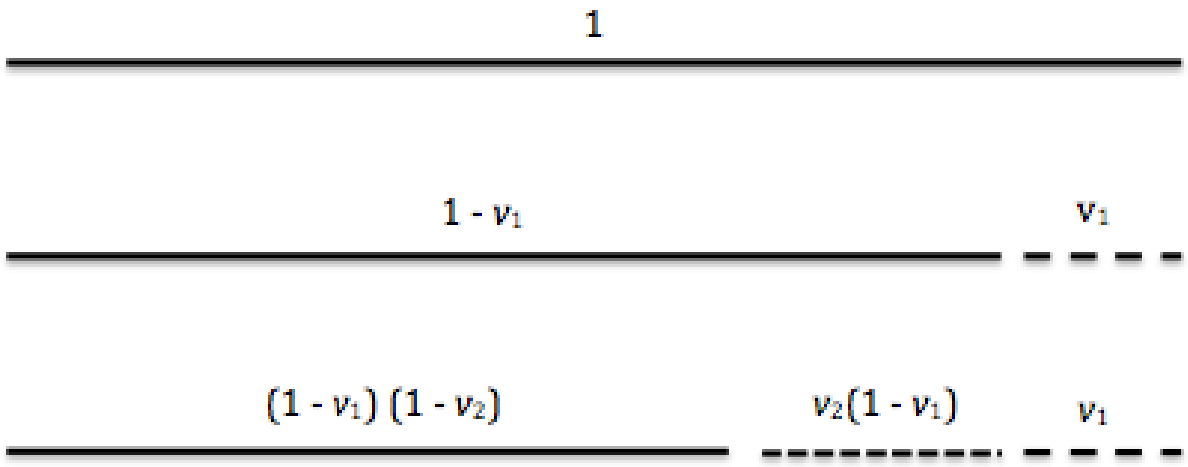
- Although useful, the definition of the DP does not provide an intuition for what realizations of a DP actually look like.
- Undoubtedly, the most useful definition of the DP is its constructive definition (Sethuraman, 1994), according to which G has an almost sure representation of the form

$$G(\cdot) = \sum_{k=1}^{\infty} \omega_k \delta_{\boldsymbol{\theta}_k}(\cdot). \tag{7}$$

- In (7), we have
  - $\boldsymbol{\theta}_k \overset{\text{iid}}{\sim} G_0$, $k = 1, 2, \ldots$
  - $\omega_1 = v_1$, and for $k \geq 2$, $\omega_k = v_k \prod_{l<k}(1 - v_l)$, with $v_k \overset{\text{iid}}{\sim} \text{Beta}(1, \alpha)$ (independently of the θ_k) $\Rightarrow$ Stick-breaking construction.

Dirichlet process mixtures

Stick-breaking representation



Dirichlet process mixtures

Stick-breaking representation

- As the stick-breaking construction proceeds, the stick gets shorter and shorter and the lengths allocated to higher index atoms decrease stochastically, with the rate of decrease depending on α.

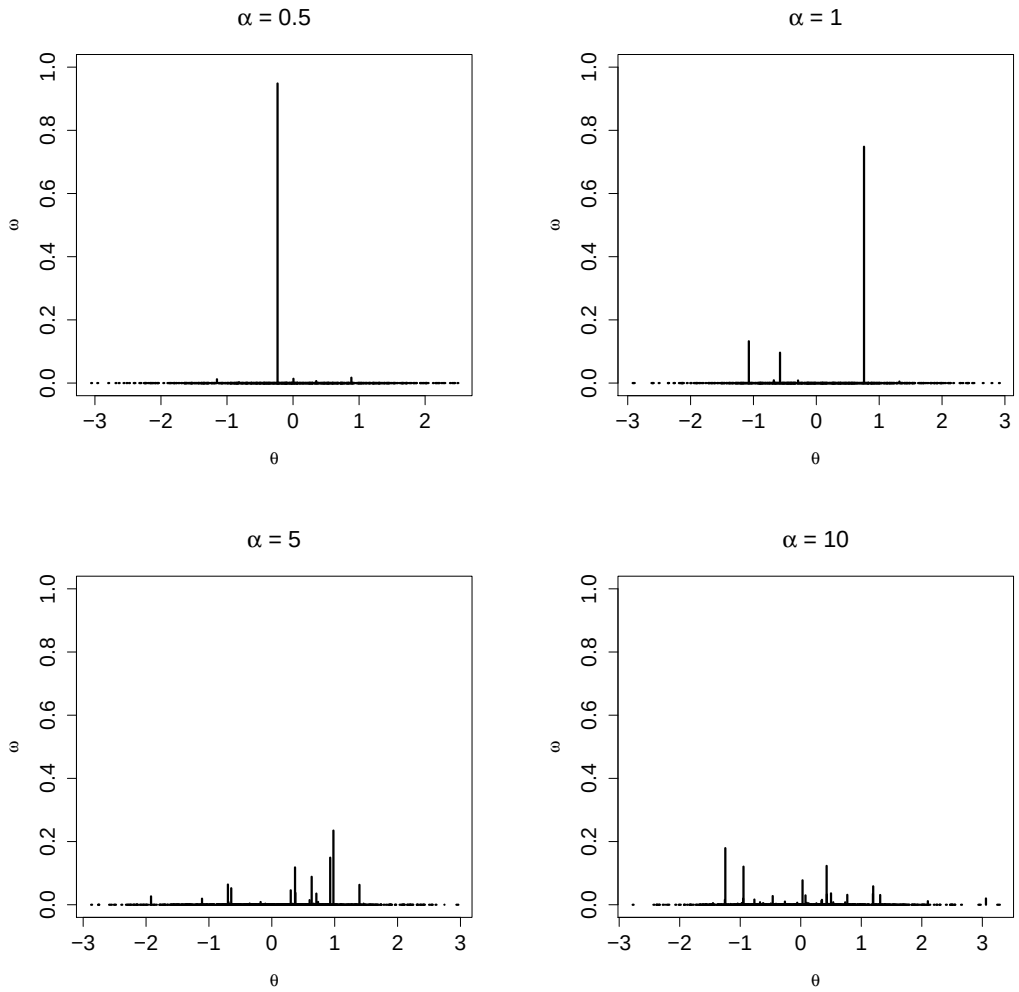
- Since $v_k \sim \text{Beta}(1, \alpha)$, then

$$E(v_k) = \frac{1}{1+\alpha},$$

so that values of α close to zero lead to high weight on the first few atoms, with the remaining atoms being assigned small probabilities.

Dirichlet process mixtures

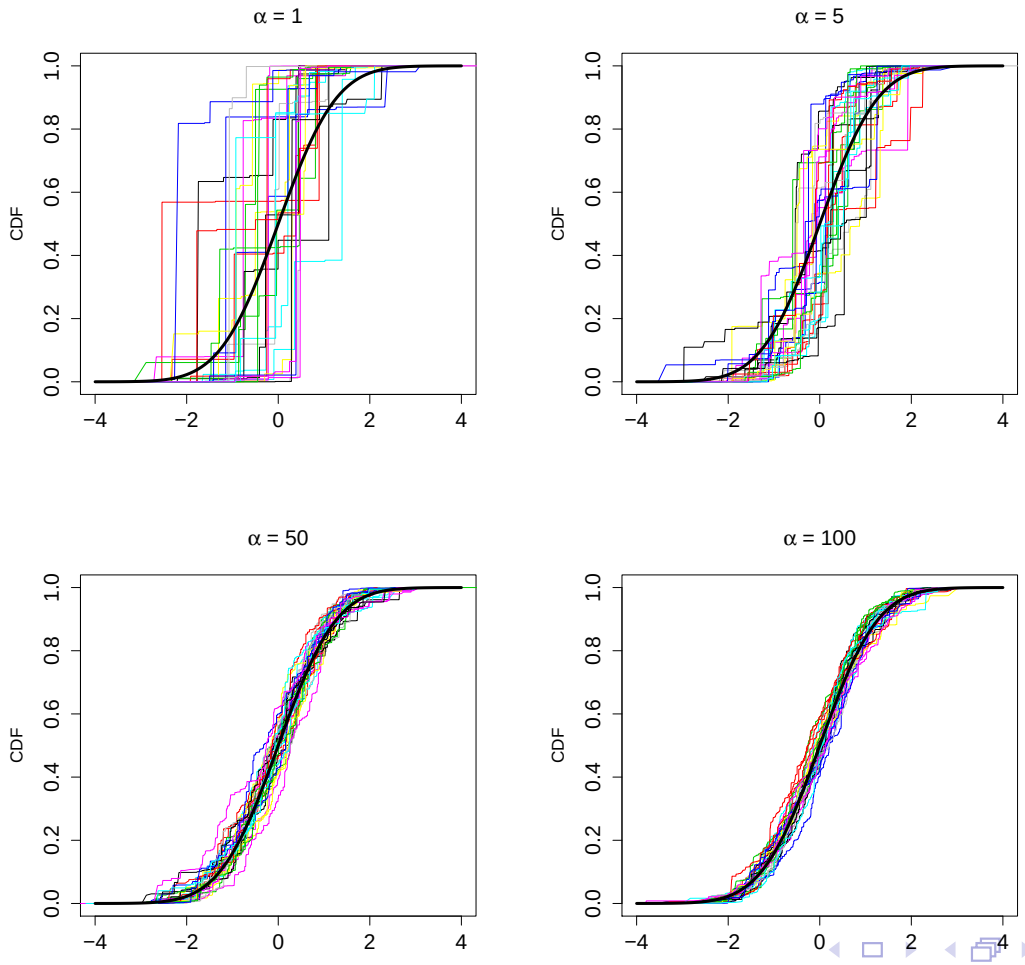
Stick-breaking representation



Dirichlet process mixtures

DP prior trajectories

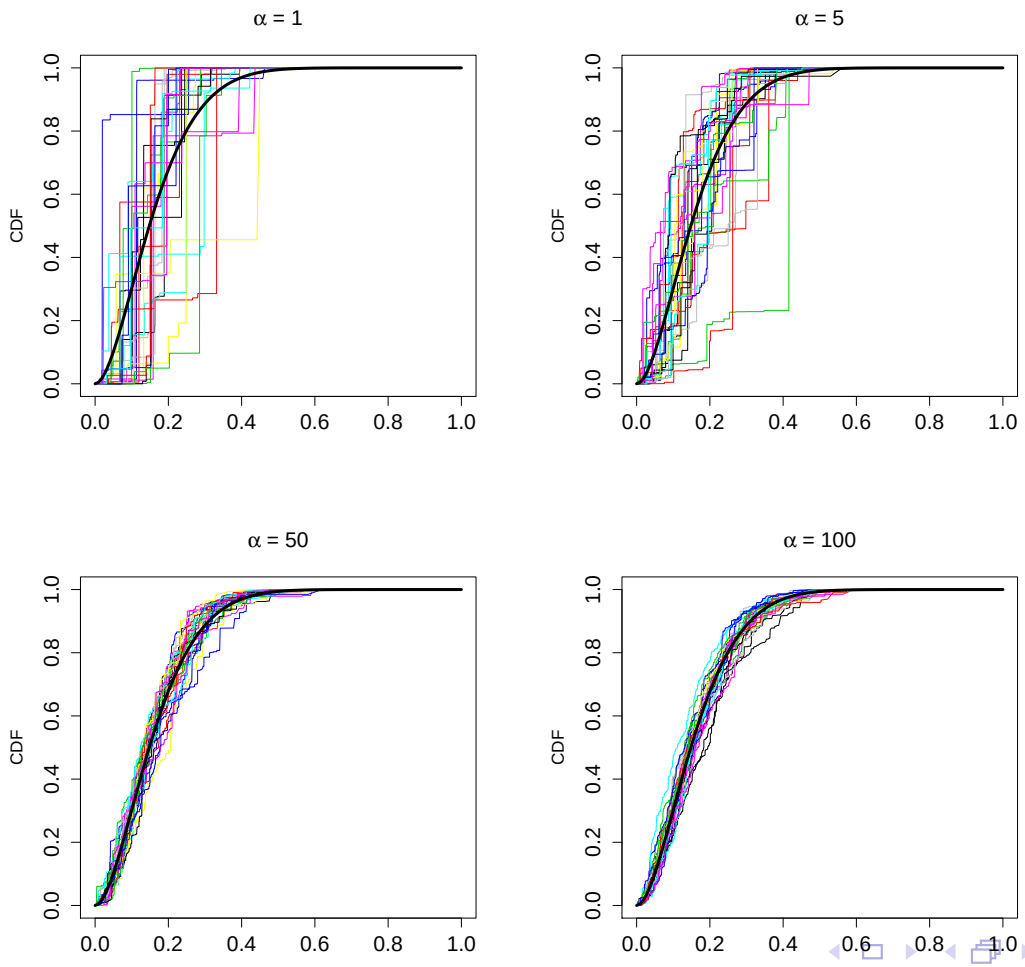
30 trajectories from a DP with $G_0 = \mathrm{N}(0, 1)$ (based on 1000 draws).



Dirichlet process mixtures

DP prior trajectories

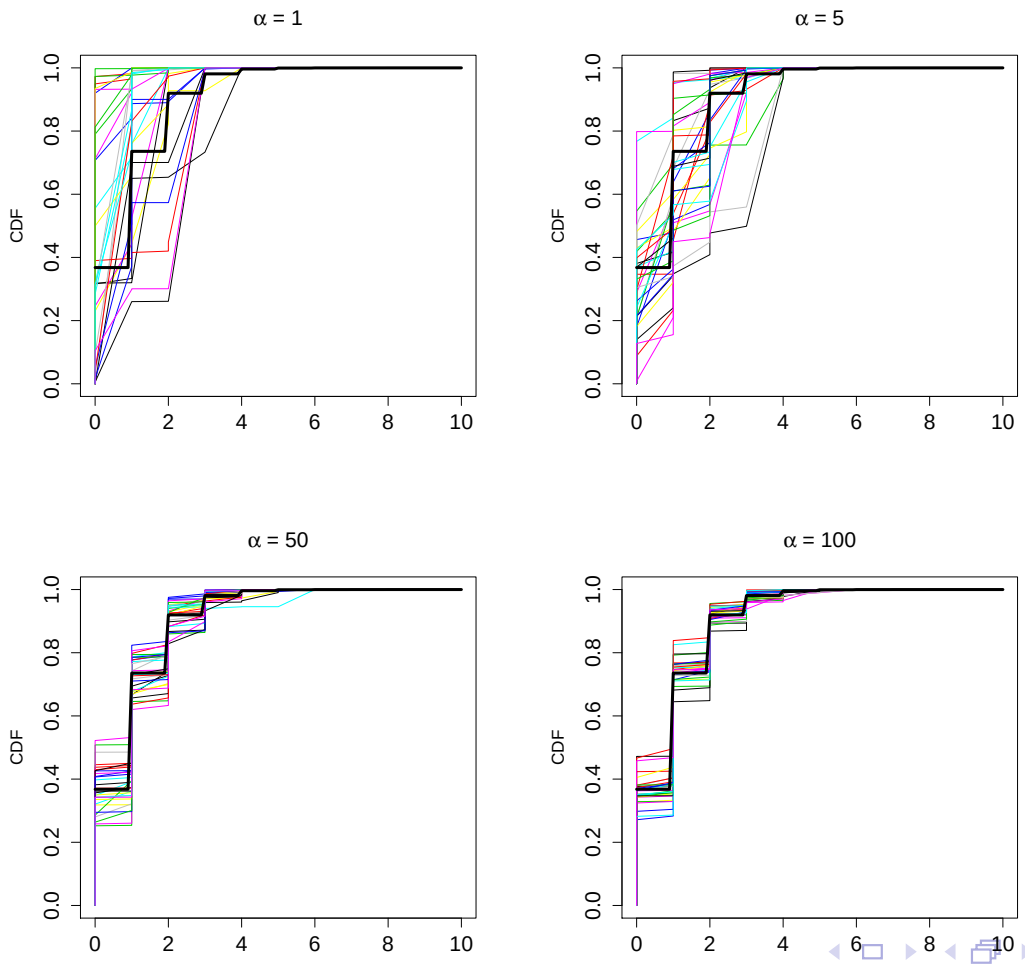
30 trajectories from a DP with $G_0 = \text{Beta}(2, 10)$ (based on 1000 draws).



Dirichlet process mixtures

DP prior trajectories

30 trajectories from a DP with $G_0 = \text{Pois}(1)$ (based on 1000 draws).



Dirichlet process mixtures

Conjugacy of the DP prior

- The DP prior is closed under sampling. That is, the posterior distribution is also a DP.
- Let $y_1, \ldots, y_n \mid G \overset{\text{iid}}{\sim} G$ and $G \sim \text{DP}(\alpha, G_0)$. Then

$$G \mid y_1, \ldots, y_n \sim \text{DP}\left(\alpha + n, \frac{\alpha}{\alpha + n} G_0 + \frac{1}{\alpha + n} \sum_{i=1}^{n} \delta_{y_i}\right)$$

- The posterior mean is then

$$E(G \mid y_1, \ldots, y_n) = \frac{\alpha}{\alpha + n} G_0 + \frac{n}{\alpha + n} \sum_{i=1}^{n} \frac{1}{n} \delta_{y_i}$$

- Thus, the posterior mean is a weighted average between the centering distribution and the empirical cdf.

Dirichlet process mixtures

Posterior inference using a DP prior/ Bayesian bootstrap

- When $\alpha \to 0$, the limiting posterior distribution is

$$G \mid y_1, \ldots, y_n \sim \text{DP}\left(n, \frac{1}{n}\sum_{i=1}^{n}\delta_{y_i}\right),$$

and thus

$$E(G \mid y_1, \ldots, y_n) = \frac{1}{n}\sum_{i=1}^{n}\delta_{y_i}$$

- This limiting posterior distribution is known as the Bayesian bootstrap (Rubin 1981, Gasparini 1995).
- A Bayesian bootstrap estimator for G can be computed as

$$G = \sum_{i=1}^{n} p_i \delta_{y_i}, \quad (p_1, \ldots, p_n) \sim \text{Dir}(n; 1, \ldots, 1). \tag{8}$$

- Again, from (8) it follows that

$$E(G \mid y_1, \ldots, y_n) = E\left(\sum_{i=1}^{n} p_i \delta_{y_i}\right) = \sum_{i=1}^{n} E(p_i)\delta_{y_i} = \frac{1}{n}\sum_{i=1}^{n}\delta_{y_i}$$

Dirichlet process mixtures

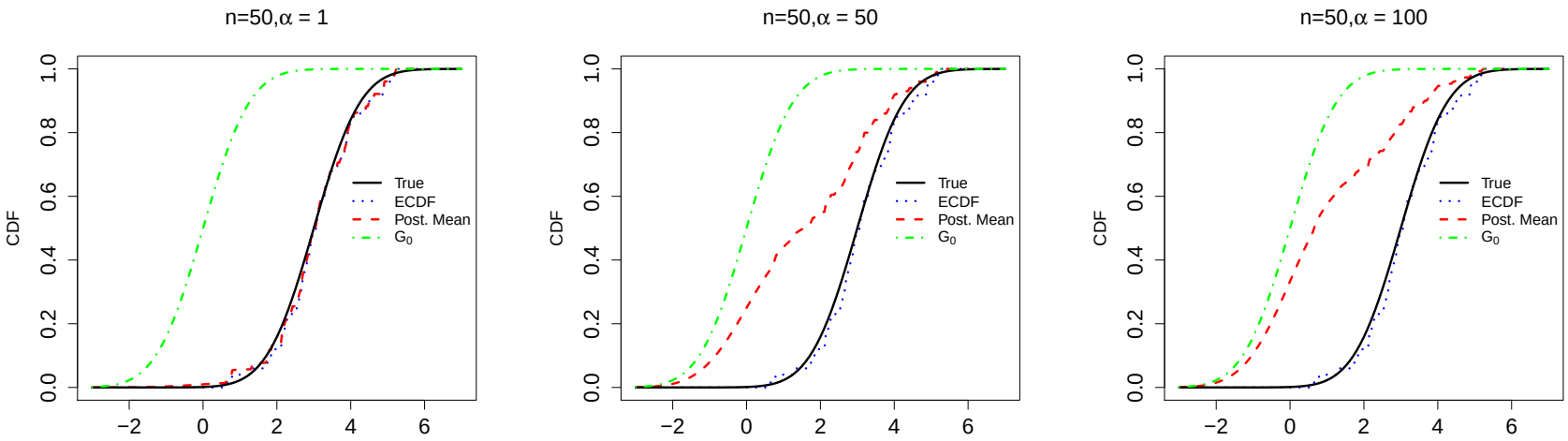
Posterior inference using a DP prior/ Bayesian bootstrap

- Recall that in the frequentist bootstrap (Efron, 1979) $p_i \in \{0, 1/n, \ldots, n/n\}$ corresponding to the number of times y_i appears in a bootstrap sample.
- Thus, the weights in the Bayesian bootstrap are smoother than those from Efron's frequentist bootstrap.
- This is justified by the fact that in the BB the weights arise from a Dirichlet (continuous distribution) while the weights in the classical bootstrap have a discrete distribution.
- Note that in the BB the data should be regarded as fixed, so that we do not resample from it.

Dirichlet process mixtures

Posterior inference using a DP prior

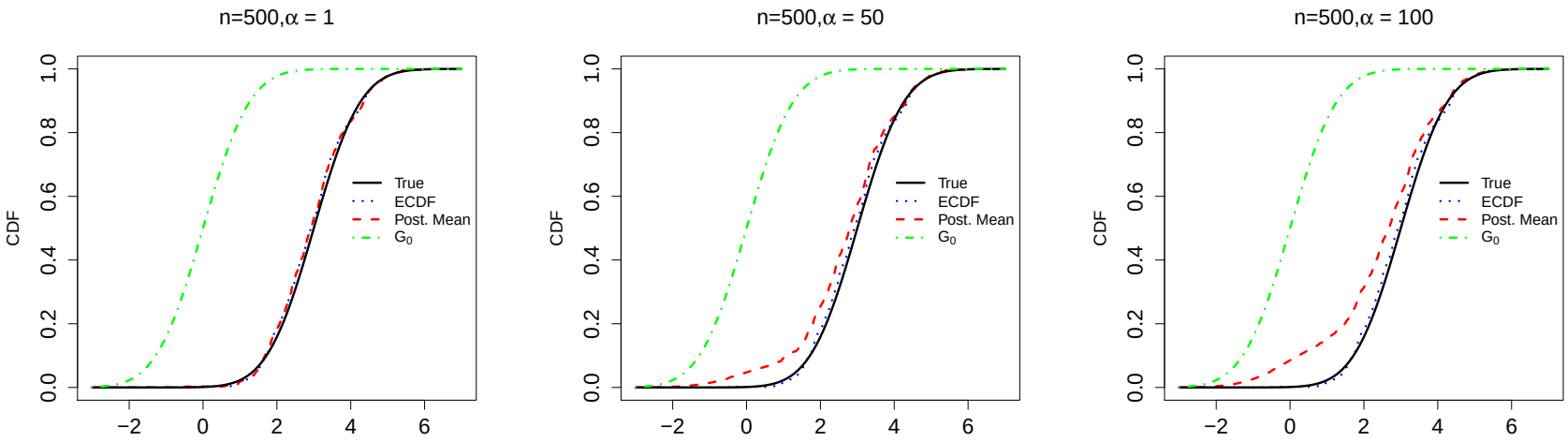
Estimating the posterior mean under a DP prior using simulated data. The true distribution generating the data is $N(3, 1)$, while the centering is a $N(0, 1)$ distribution.



Dirichlet process mixtures

Posterior inference using a DP prior

Estimating the posterior mean under a DP prior using simulated data. The true distribution generating the data is $N(3, 1)$, while the centering is a $N(0, 1)$ distribution.



Dirichlet process mixtures

DPM model

- The DP generates flexible albeit discrete distributions.
- Due to this fact, and as we have already anticipated, it is commonly used as a prior in mixture models.
- The Dirichlet process mixture model can be specified as

$$f(y) = \int k(y \mid \theta) \mathrm{d}G(\theta), \quad G \sim \mathrm{DP}(\alpha, G_0)$$

- Because G is random, the mixture density f is also random.
- Further, the mixture density can be discrete or continuous, univariate or multivariate, depending on the nature of $k(y \mid \theta)$.

Dirichlet process mixtures

DPM model

- Hereby, we focus on Dirichlet process mixtures of normals

$$f(y) = \int \phi(y \mid \mu, \sigma^2) \mathrm{d}G(\mu, \sigma^2), \quad G \sim \mathrm{DP}(\alpha, G_0). \tag{9}$$

- The centering distribution G_0 is defined on $\mathbb{R} \times \mathbb{R}^+$.
- Due to conjugacy reasons, G_0 is usually taken to be the normal-inverse-gamma distribution, that is

$$G_0 \equiv \mathrm{N}(a_\mu, b_\mu^2)\mathrm{IG}(a_{\sigma^2}, b_{\sigma^2}).$$

- To allow for extra flexibility, hyperpriors can be placed on a_μ, b_μ^2, a_{σ^2}, b_{σ^2}.
- The stick-breaking representation of the DP allows us to write (9) as the following countably infinite mixture of normals

$$f(y) = \sum_{k=1}^{\infty} \omega_k \phi(y \mid \mu_k, \sigma_k^2), \tag{10}$$

- Note that equation (10) resembles the finite mixture model considered earlier but with the important difference that the number of mixture components is set to infinite and the weights now follow a stick-breaking construction.

Dirichlet process mixtures

MCMC

- Posterior inference can be conducted using (at least) two different kinds of MCMC strategies:
  1. Employing a truncation of the stick-breaking representation (Ishwaran and James, 2001, 2002).
  2. Using a marginal/collapsed Gibbs sampling where the mixing distribution is integrated out from the model (MacEachern 1998, Neal 2000).
- Throughout approach 1 will be detailed.

Dirichlet process mixtures

Blocked Gibbs sampler

- The blocked Gibbs sampler relies on truncating the stick-breaking representation to a finite number of components.
- Hence

$$G_N(\cdot) = \sum_{k=1}^{N} \omega_k \delta_{(\mu_k, \sigma_k^2)}(\cdot).$$

- The atoms (μ_k, σ_k^2) are iid G_0, i.e., $(\mu_k, \sigma_k^2) \overset{\text{iid}}{\sim} \text{N}(a_\mu, b_\mu^2)\text{IG}(a_{\sigma^2}, b_{\sigma^2})$, $k = 1, \ldots, N$.
- The weights arise through a truncated stick-breaking construction

$$\omega_1 = v_1, \quad \text{for } k \geq 2 \quad \omega_k = v_k \prod_{l<k}(1 - v_l), \quad v_k \overset{\text{iid}}{\sim} \text{Beta}(1, \alpha), \ k = 1, \ldots, N-1$$

$$v_N = 1, \quad \omega_N = \prod_{l=1}^{N-1}(1 - v_l)$$

Dirichlet process mixtures

Blocked Gibbs sampler

- Ishwaran and James (2001) showed the following bound for the truncation error

$$4\left[1 - E\left\{\left(\sum_{k=1}^{N-1}\omega_k\right)^n\right\}\right] \approx 4n\exp\left(\frac{1-N}{\alpha}\right)$$

- For instance, if $n = 500$ and if we use a truncate value of $N = 20$, then for $\alpha = 1$, we get a bound of $\approx 1.1 \times 10^{-5}$.

- In practice, $N = 20$ or $N = 50$ are commonly chosen as a default.

Dirichlet process mixtures

Blocked Gibbs sampler

- Using the truncated version G_N of G, the normal mixture density can be expressed as

$$f(y) = \sum_{k=1}^{N} \omega_k \phi(y \mid \mu_k, \sigma_k^2),$$

with ω_k generated from the truncated stick-breaking representation, whereas $\mu_k \overset{\text{iid}}{\sim} \text{N}(a_\mu, b_\mu^2)$, and $\sigma_k^2 \overset{\text{iid}}{\sim} \text{IG}(a_{\sigma^2}, b_{\sigma^2})$.

- As it was the case for the finite mixture model, derivation of the full conditionals for Gibbs sampling involves the data-augmented likelihood.

- The means, variances, and latent component indicators are sampled in an identical manner to the finite mixture model.

- The main difference is that, unlike in the finite mixture model, uncertainty in the component weights is shifted to **v**, the inputs into the construction of the stick-breaking weights.

Dirichlet process mixtures

Blocked Gibbs sampler

- The full conditional distributions are

$$\mu_k \mid \text{else} \sim \text{N}\left(\frac{a_\mu/b_\mu^2 + \sum_{i:z_i=k} y_i/\sigma_k^2}{1/b_\mu^2 + n_k/\sigma_k^2}, \frac{1}{1/b_\mu^2 + n_k/\sigma_k^2}\right), \tag{11}$$

$$\sigma_k^2 \mid \text{else} \sim \text{IG}\left(a_{\sigma^2} + n_k/2, b_{\sigma^2} + \sum_{i:z_i=k} (y_i - \mu_k)^2/2\right), \tag{12}$$

  for $k = 1, \ldots, N$.

- For $i = 1, \ldots, n$, the full conditional distribution for z_i is

$$z_i \mid \text{else} \sim \text{Mult}(\mathbf{p}_i),$$

  with $\mathbf{p}_i = (p_{i1}, \ldots, p_{iN})$ and $p_{ik} = \frac{\omega_k \phi(y_i|\mu_k, \sigma_k^2)}{\sum_{l=1}^K \omega_l \phi(y_i|\mu_l, \sigma_l^2)}$, $k = 1, \ldots, N$.

Dirichlet process mixtures

Blocked Gibbs sampler

- For $k = 1, \ldots, N-1$, update the inputs of the stick-breaking weights from

$$v_k \mid \text{else} \sim \text{Beta}\left(n_k + 1, \alpha + \sum_{l=k+1}^{N} n_l\right). \tag{13}$$

- Regarding the precision parameter α, it can be fixed at a small value, for instance, $\alpha = 1$ is widely used in applications.

- Alternatively, one can place a prior on α and allow the data to inform about the appropriate value of α.

- Letting $\alpha \sim \text{Gamma}(a_\alpha, b_\alpha)$, the resulting full conditional for α is

$$\alpha \mid \text{else} \sim \text{Gamma}\left(a_\alpha + N - 1, b_\alpha - \sum_{k=1}^{N-1} \log(1 - v_k)\right). \tag{14}$$

Algorithm

Set an initial value for α, ω, *and* $\boldsymbol{\theta}$, *say,* $\alpha^{(0)}$, $\omega^{(0)}$, *and* $\boldsymbol{\theta}^{(0)}$.

`for` $t = 1, \ldots, T$, `do`:

1. `for` $i = 1, \ldots, n$, *and* $k = 1, \ldots, K$, *compute posterior probabilities of membership using equation* (6).
2. `for` $i = 1, \ldots, n$, *sample the latent component indicator,*

$$z_i^{(t)} \sim Multi(\mathbf{p}_i^{(t)}).$$

3. *Simulate stick-breaking inputs* $\mathbf{v}^{(t)}$ *from equation* (13).
4. *Given* $\mathbf{v}^{(t)}$, *compute* $\boldsymbol{\omega}^{(t)}$ *using the stick-breaking construction.*
5. *Conditional on* $\mathbf{z}^{(t)}$, *update* $\mu_k^{(t)}$ *and* $(\sigma_k^{(t)})^2$, `for` $k = 1, \ldots, K$, *from* (11) *and* (12), *respectively.*
6. *Update the precision parameter* α *from* (14).

Dirichlet process mixtures

Blocked Gibbs sampler

- A valid concern with the blocked Gibbs sampler is that by truncating the stick-breaking representation we are effectively fitting a finite (and hence parametric) mixture model.

- Quoting Dunson (2011):

  "*For example, if we let $N = 25$ as a truncation level, a natural question is how this is better or intrinsically different than fitting a finite mixture model with 25 components. One answer is that N is not the number of components occupied by the subjects in your sample but is instead an upper bound on the number of subjects.*

Dirichlet process mixtures

Collapsed Gibbs sampler/ Polya urn scheme

- The function `DPdensity` from the `R` package `DPpackage` can also to be used to fit a DPM of normals.
- The MCMC scheme behind this function is a marginalized/collapsed Gibbs sampler.
- The collapsed Gibbs sampler avoids specifying a truncation level by marginalizing out G and relying on the Polya urn scheme of Blackwell and MacQueen (1973).
- Letting

$$y_i \sim \phi(\boldsymbol{\theta}_i), \quad \boldsymbol{\theta}_i = (\mu_i, \sigma_i^2) \sim G, \quad G \sim \text{DP}(\alpha, G_0),$$

and marginalizing out G, we obtain the Polya urn predictive rule

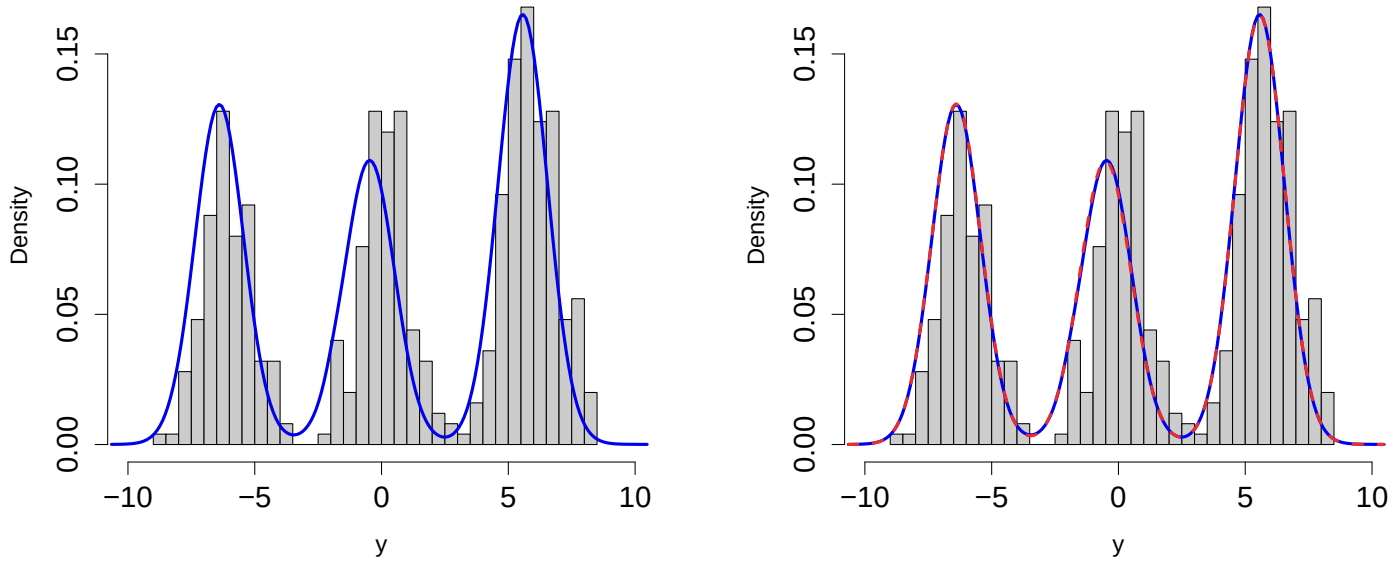
$$p(\boldsymbol{\theta}_i \mid \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_{i-1}) \propto \frac{\alpha}{\alpha + i - 1} G_0(\theta_i) + \frac{1}{\alpha + i - 1} \sum_{j=1}^{i-1} \delta_{\boldsymbol{\theta}_j}(\boldsymbol{\theta}_i)$$

- The Polya urn rule form the basis of the collapsed Gibbs sampler. For those interested in further details, see (Escobar and West 1995, MacEachern 1998, Neal 2000).

Dirichlet process mixtures

Example

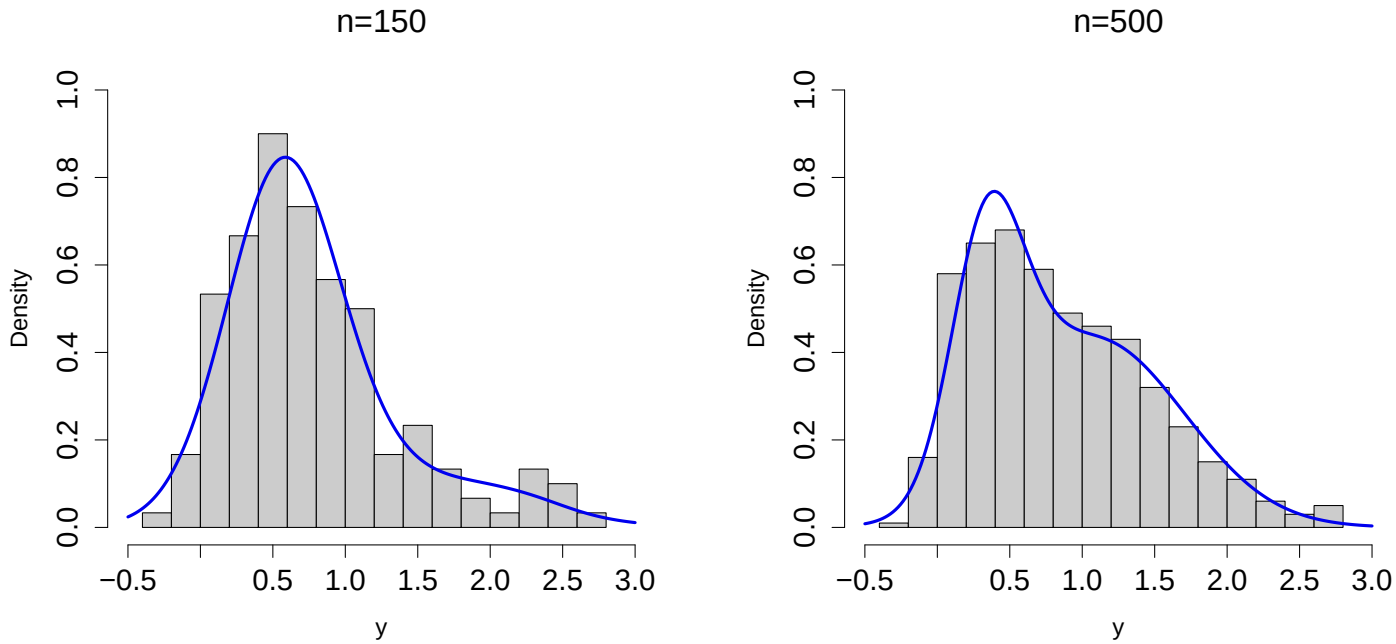
Same data from the finite mixture example. DPM fit (left) and comparison against the fit of a 3 component mixture model, which corresponds to the true data generating mechanism (right).



Dirichlet process mixtures

Example

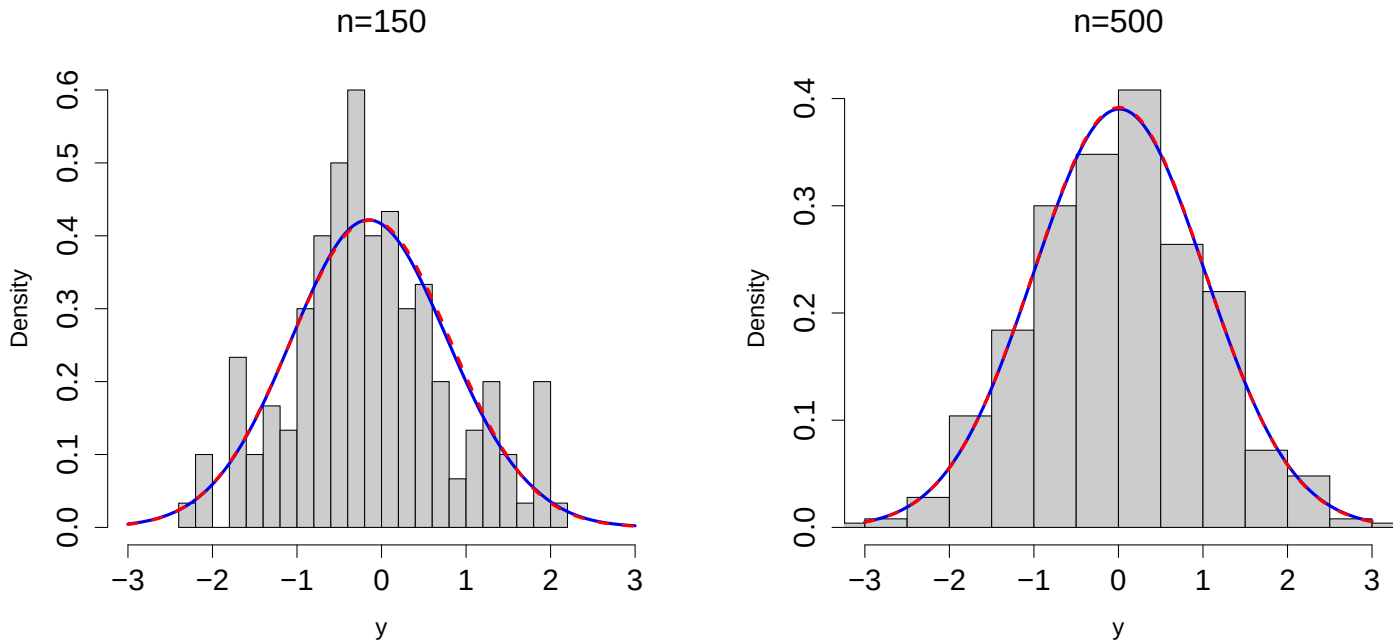
True data generating mechanism: $\phi_{\text{SN}}(y \mid \mu = 0, \sigma^2 = 1, \lambda = 8)$



Dirichlet process mixtures

Example

True data generating mechanism: $\phi(y \mid 0, 1)$. DPM fit (blue) against normal fit (red).



Dirichlet process mixtures

Censored responses

- The DPM can easily handle censored responses (left,right, or interval).
- Remember that under the hierarchical formulation

$$y_i \mid \boldsymbol{\mu}, \boldsymbol{\sigma}^2, \mathbf{z} \sim \phi(y_i \mid \mu_{z_i}, \sigma^2_{z_i})$$

$$\vdots$$

- For instance, if y_i is right censored, we know that its true value, say y_i^*, is greater than y_i, that is, $y_i^* > y_i$.
- We can take care of these censored observations by simply adding an extra step in the blocked Gibbs algorithm.
- In fact, we can simulate those observations from

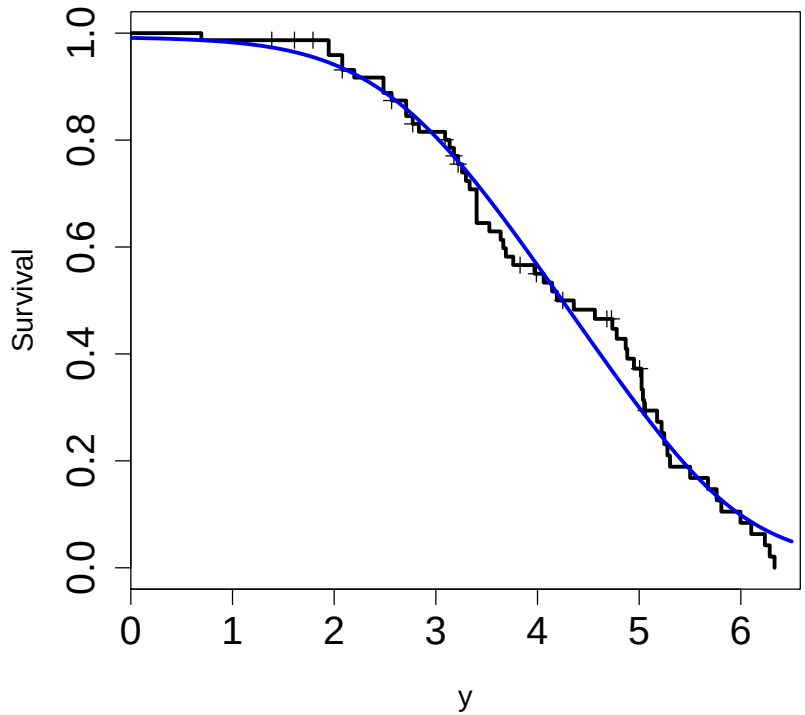
$$y_i^* \mid y_i, z_i, \boldsymbol{\mu}, \boldsymbol{\sigma}^2 \sim \phi(y_i^* \mid \mu_{z_i}, \sigma^2_{z_i}) I(y_i^* > y_i).$$

- This can be accomplished by simulating y_i^* from a truncated normal distribution with lower limit equal to y_i.

Dirichlet process mixtures

Censored responses

DPM fit (blue line) against Kaplan–Meier fit (black line). The censored observations are represented by crosses.



Dependent Dirichlet process mixtures

Density regression

- So far we have focused on the problem of density estimation.
- We will now move to the problem of density regression.
- Traditional regression models allow just one or two characteristics (e.g., mean and/or variance) to change as a function of covariates.
- Here we will explore tools that allow the entire density/distribution to change as a function of covariates.

Dependent Dirichlet process mixtures

Density regression

- Let $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$ be regression data, where $\mathbf{x}_i \in \mathcal{X} \subseteq \mathbb{R}^p$.
- It is assumed that
$$y_i \mid \mathbf{x}_i \overset{\text{ind.}}{\sim} f(\cdot \mid \mathbf{x}_i), \quad i = 1, \ldots, n.$$
- We specify a probability model for the entire collection of densities $\mathcal{F} = \{f(\cdot \mid \mathbf{x}) : \mathbf{x} \in \mathbb{X}\}$.
- Further, one possibility is to model the conditional density using covariate-dependent mixture of normal models
$$f(y \mid \mathbf{x}) = \int \phi(y \mid \mu, \sigma^2) \mathrm{d}G_{\mathbf{x}}(\mu, \sigma^2).$$
- The probability model for the conditional densities is induced by specifying a prior for the collection of mixing distributions
$$G_{\mathcal{X}} = \{G_{\mathbf{x}} : \mathbf{x} \in \mathcal{X}\} \sim \mathcal{G}.$$
- $G_{\mathbf{x}}$ is the random mixing distribution at covariate $\mathbf{x}$ and $\mathcal{G}$ is the prior for the collection $G_{\mathcal{X}}$.

Dependent Dirichlet process mixtures

DDP prior

- One possibility for $\mathcal{G}$ is the dependent DP (DDP) proposed by MacEachern (1999,2000), which is built upon the constructive definition of the DP.
- In its full generality, the DDP is specified as follows:

$$G_{\mathbf{x}}(\cdot) = \sum_{k=1}^{\infty} \omega_{k,\mathbf{x}} \delta_{\theta_{k,\mathbf{x}}}(\cdot), \quad \omega_{1,\mathbf{x}} = v_{1,\mathbf{x}}, \quad \omega_{k,\mathbf{x}} = v_{k,\mathbf{x}} \prod_{l<k} (1 - v_{k,\mathbf{x}}).$$

- Here
  - $\theta_{1,\mathbf{x}}, \theta_{2,\mathbf{x}}, \ldots$ are realizations of a stochastic process (e.g., a Gaussian process) over $\mathcal{X}$.
  - $v_{1,\mathbf{x}}, v_{2,\mathbf{x}}, \ldots$ are realizations from a stochastic process on $\mathcal{X}$ such that $v_{k,\mathbf{x}} \sim \text{Beta}(1, \alpha_{\mathbf{x}})$

Dependent Dirichlet process mixtures

DDP prior

- Because of complications involved in allowing the weights to depend on covariates, the 'single weights' DDP, which assumes fixed weights, is commonly used.
- Following De Iorio et al. (2009), a possibility for $G_{\mathbf{x}}$ is

$$G_{\mathbf{x}}(\cdot) = \sum_{k=1}^{\infty} \omega_k \delta_{\theta_{k,\mathbf{x}}}(\cdot),$$

  where the weights match those from a standard DP and $\theta_{k,\mathbf{x}} = (\mu_{k,\mathbf{x}}, \sigma_k^2)$, with $\mu_{k,\mathbf{x}} = \mathbf{x}^T \beta_k$.
- Thus, under this formulation, the base stochastic processes are replaced with a base distribution G_0 that generates the component-specific regression coefficients and variances.

Dependent Dirichlet process mixtures

DPM of Gaussian regression models

- Thus, the conditional density can therefore be represented as a DP mixture of Gaussian regression models

$$f(y \mid \mathbf{x}) = \int \phi(y \mid \mathbf{x}^T\beta, \sigma^2)\mathrm{d}G(\beta, \sigma^2), \quad G \sim \mathrm{DP}(\alpha, G_0). \tag{15}$$

- For example, G_0 could be $N_p(\mu_\beta, \mathbf{S}_\beta)\mathrm{IG}(a_{\sigma^2}, b_{\sigma^2})$.
- This model is known as the linear dependent Dirichlet process (LDDP) (De Iorio 2004, 2009).
- Note that Eq. (15) can be equivalently written as

$$f(y \mid \mathbf{x}) = \sum_{k=1}^{\infty} \omega_k \phi(y \mid \mathbf{x}^T\beta_k, \sigma_k^2).$$

Dependent Dirichlet process mixtures

DPM of Gaussian regression models - spline based version

- The LDDP although flexible, does not allow for nonlinear effects of the covariates.
- An alternative is instead of considering $\mu_{k,\mathbf{x}} = \mathbf{x}^T \boldsymbol{\beta}_k$ to consider an additive formulation based on B-splines, namely

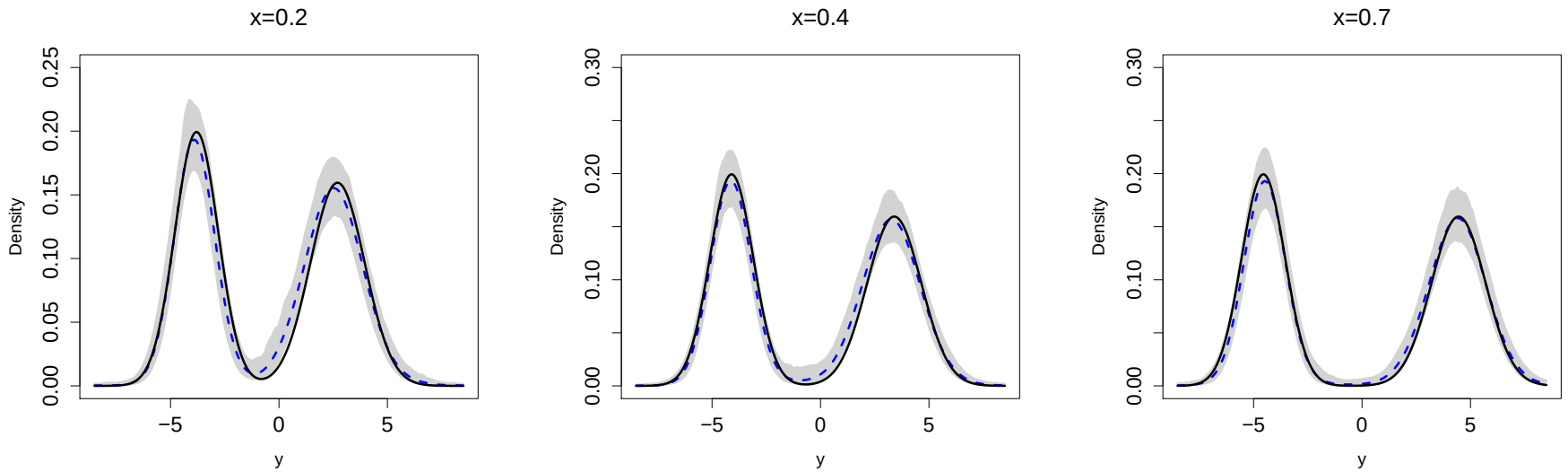
$$\mu_{k,\mathbf{x}} = \beta_{k0} + \sum_{r=1}^{p} \left(\sum_{s=1}^{L_r} \beta_{krs} \Psi_s(x_r, d_r) \right),$$

  where $\Psi(x, d)$ corresponds to the sth B spline basis function of degree d evaluated at x.
- As in the density estimation setup, posterior inference can be conducted using a blocked Gibbs sampler or a collapsed Gibbs sampler.

Dependent Dirichlet process mixtures

Example

Data generating mechanism: $y_i \mid x_i \sim 0.5\phi(y_i \mid -3.5 - 1.5x_i, 1^2) + 0.5\phi(y_i \mid 2 + 3.5x_i, 1.25^2)$, $x_i \sim \text{U}(0, 1)$, for $i = 1, \ldots, 500$. Results from the fit of a LDDP.



Mixtures of finite Polya trees

PT prior

- We now focus on another popular nonparametric prior for density/distribution estimation. The discussion closely follows Branscum, Johnson, and Baron (2013).
- Polya tree priors have been discussed as early as Freedman (1963), Fabius (1964), and Ferguson (1974).
- However, the natural starting point for understanding their potential use in modeling data is Lavine (1992, 1994), while Hanson (2006) considered some computational details.
- Polya trees are way less popular than DPs but they are a very powerful tool as well.

Mixtures of finite Polya trees

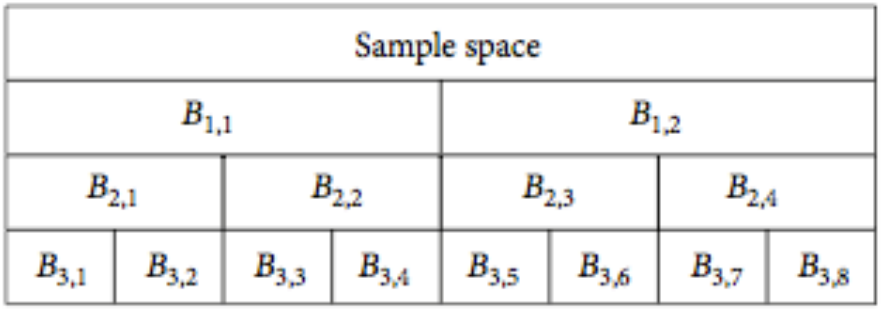
PT prior

- Suppose that a random sample $y_1, \ldots, y_n$ is obtained from an unknown/random distribution F.
- Polya tree priors, as Dirichlet process priors, place a distribution on a collection of distributions.
- A Polya tree for a distribution F is constructed by dividing the sample space into finer-and-fines disjoint sets using successive binary partitioning.
- For instance, the first partition splits the sample space into two non overlapping intervals.
- In the second partition, those two intervals are each split, yielding a finer partition that contain four intervals.
- Then, these four intervals are each split to give an eight interval third level partition of the sample space.
- At level j of the tree, the sample space is partitioned into 2^j intervals, $j = 1, \ldots$

Mixtures of finite Polya trees

PT prior

- Let $B_{j,k}$ denotes the kth interval at level j of the tree, for $j = 1, \ldots,$ and $k = 1, \ldots, 2^j$.



- Observe that the partitions are nested within one another, starting at the top of the tree and working up.
- For example, by definition, $B_{1,1} = B_{2,1} \cup B_{2,2}$ and $B_{j-1,1} = B_{j,1} \cup B_{j,2}$, etc.
- These intervals can be thought as the bins in the histogram.

Mixtures of finite Polya trees

Finite PT prior

- For a full tree the splitting continues ad infinitum.
- However, in practice, we truncate to a fixed J, hence the term, finite Polya tree.
- Generally, setting J equal to 4, 5, or 6 often works well in practice.
- Another option is to select J so that roughly $J = \log_2 n$ (Hanson, 2006).

Mixtures of finite Polya trees

Finite PT prior

- Informally speaking, the unknown distribution F assigns the data points y_is to the intervals at level J of the tree and the task is to use the observed distribution of the data into the intervals to estimate F.
- Although all levels of the tree are important for the purpose of estimating F, of primary importance is level J.
- The goal here is to produce a data-driven estimate of F that assigns high probability to intervals that contain lots of data, assigns low probability to empty intervals, and assigns midrange probability to intervals that contain some (but not a lot) of the data.

Mixtures of finite Polya trees

Finite PT prior

- Let us consider first the simplest case $J = 1$.
- Then data are assigned to either $B_{1,1}$ or $B_{1,2}$.
- The probability of a data point y_i being assigned to $B_{1,1}$ is $F(B_{1,1}) = \Pr(y_i \in B_{1,1})$ which since F is a cdf and $B_{1,1}$ is an interval of the form (L, U) is to be interpreted as $F(B_{1,1}) = F(U) - F(L)$.
- Denote this unknown probability by $\pi_{1,1}$.
- Then, by the complement rule, $\pi_{1,2} = 1 - \pi_{1,1}$ is the probability assigned to set $B_{1,2}$.
- In notation, $\Pr(y_i \in B_{1,1}) = F(B_{1,1}) = \pi_{1,1}$ and $\Pr(y_i \in B_{1,2}) = F(B_{1,2}) = \pi_{1,2}$.
- Since F is unknown, $\pi_{1,1}$ and $\pi_{1,2}$ are also unknown.

Mixtures of finite Polya trees

Finite PT prior

- To help interpret these parameters, suppose the data arise from a right-skewed distribution (lots of more data in $B_{1,1}$ than in $B_{1,2}$), then $\pi_{1,1}$ would be large and hence $\pi_{1,2}$ would be small, and vice versa for left-skewed data.

- Obviously, in practice, $J = 1$ is never used because it would lead to a crude estimate of the density function f, much like estimating a density using a relative frequency histogram that contains only two bins.

Mixtures of finite Polya trees

Finite PT prior

- Let us now consider, again for simplicity, the case of $J = 2$. We have now four sets.
- Data assignment is based on whether the data point was in $B_{1,1}$ or $B_{1,2}$ at the previous level $j = 1$.
- If $y_i \in B_{1,1}$ then y_i is assigned to interval $B_{2,1}$ with unknown probability $\pi_{2,1}$ or to interval $B_{2,2}$ with probability $\pi_{2,2} = 1 - \pi_{2,1}$.
- Similarly, define $\pi_{2,3}$ and $\pi_{2,4}(= 1 - \pi_{2,3})$ to be the probability of y_i being assigned to set $B_{2,3}$ or $B_{2,4}$, respectively, given that y_i was on $B_{1,2}$.
- The $\pi_{j,k}$s are conditional parameters, since $\pi_{2,1} = \Pr(y_i \in B_{2,1} \mid y_i \in B_{1,1})$ and $\pi_{2,3} = \Pr(y_i \in B_{2,3} \mid y_i \in B_{1,2})$.
- To relate the $\pi_{j,k}$s to F we must determine the marginal probability of assignment to the various intervals at level $J(= 2)$.

Mixtures of finite Polya trees

Finite PT prior

- Observe that interval $B_{2,1}$ is nested on interval $B_{1,1}$, so the marginal probability of interval $B_{2,1}$ is

$$\begin{aligned} F(B_{2,1}) &= \Pr(y_i \in B_{2,1}) \\ &= \Pr(y_i \in B_{2,1} \cap B_{1,1}) \\ &= \Pr(y_i \in B_{2,1} \mid y_i \in B_{1,1}) \Pr(y_i \in B_{1,1}) \\ &= \pi_{2,1}\pi_{1,1}. \end{aligned}$$

- Similar steps lead to $F(B_{2,2}) = (1 - \pi_{2,1})\pi_{1,1}$, $F(B_{2,3}) = \pi_{2,3}\pi_{1,2}$, and $F(B_{2,4}) = (1 - \pi_{2,3})\pi_{1,2}$.

- Suppose again that F is right skewed . Then the data will estimate $\pi_{1,1}$ to be large, and it will estimate $\pi_{2,1}$ to be large since most of the n data points will be assigned to set $B_{2,1}$.

- Therefore, the estimate of $F(B_{2,1})$ will be (relatively) large.

Mixtures of finite Polya trees

Finite PT prior

- Notice that level 1 has only one unique parameter, $\pi_{1,1}$, associated with it because $\pi_{1,2}$ is completely determined by $\pi_{1,1}$.
- Similarly, level 2 has two unique parameters, $\pi_{2,1}$ and $\pi_{2,3}$ associated with it.
- We can continue the partitioning to any level J.
- For $J = 3$, we add eight conditional probabilities parameters, $\pi_{3,1}, \pi_{3,2}, \ldots, \pi_{3,8}$, but only four of these are unique.
- For instance, $\pi_{3,1}$ is the probability that y_i is in interval $B_{3,1}$ given that it is in interval $B_{2,1}$, and

$$\begin{aligned} F(B_{3,1}) &= \Pr(y_i \in B_{3,1}) \\ &= \Pr(y_i \in B_{3,1} \cap B_{2,1}) \\ &= \Pr(y_i \in B_{3,1} \mid y_i \in B_{2,1}) \Pr(y_i \in B_{2,1}) \\ &= \pi_{3,1}\pi_{2,1}\pi_{1,1}. \end{aligned}$$

- In general, we have

$$F(B_{j,k}) = \prod_{l=1}^{j} \pi_{l,\text{Int}\{(k-1)2^{l-j}+1\}}, \quad j = 1, \ldots, J, \quad k = 1, \ldots, 2^j.$$

Mixtures of finite Polya trees

Finite PT prior

- The key point is that if we can estimate all of the $\pi_{j,k}$s, then we can estimate the probability that it is allocated by F to each interval at level J.
- So far, we have modeled the probability of assignment to each set at level J, but we have not modeled how probability mass is distributed within each interval at level J.
- For instance, all the y_is can be clumped together in the center of the interval.
- Alternatively, the data could be uniformly distributed, clumped to the right or left side of the interval or have any other dispersion pattern within each interval.
- To address this issue, we model the data according to how a user-specified parametric distribution F_0 allocated probability mass within the intervals at level J.
- So, as it was in the DP (or DPM), here with finite Polya trees, the user also needs to specify a probability distribution (and we will see in a few slides that F_0 is also a centering distribution).

Mixtures of finite Polya trees

Finite PT prior

- The distribution F_0 is also used to determine the lower and upper endpoints of all intervals in the tree.
- The median of F_0 is used to split the sample space into two intervals at level 1 of the tree.
- The quartiles of F_0 define cut points for intervals at level 2.
- Writing the 25th percentile as $F_0^{-1}(1/4)$, the median as $F_0^{-1}(2/4)$, and the 75th percentile as $F_0^{-1}(3/4)$, we have for a sample space that covers the real line

$$B_{2,1} = \left(-\infty, F_0^{-1}(1/4)\right), \quad B_{2,2} = \left(F_0^{-1}(1/4), F_0^{-1}(2/4)\right)$$
$$B_{2,3} = \left(F_0^{-1}(2/4), F_0^{-1}(3/4)\right), \quad B_{2,4} = \left(F_0^{-1}(3/4), \infty\right)$$

- In general, the (j,k)th interval is

$$B_{j,k} = \left(F_0^{-1}\left(\frac{k-1}{2^j}\right), F_0^{-1}\left(\frac{k}{2^j}\right)\right), \quad j = 1, \ldots, J, \quad k = 1, \ldots, 2^j.$$

Mixtures of finite Polya trees

Finite PT prior

- The collection $\Pi = \{\pi_{j,k} : j = 1, \ldots, J, k = 1, \ldots, 2^j\}$ constitutes the unknown parameters corresponding to F.
- The probabilities in Π are assumed mutually independent. That is, for instance, (π_{21}, π_{22}) and (π_{23}, π_{24}) are independent.
- We thus need to specify a prior distribution over this collection.
- Due to the fact that when k is an even number between 2 and 2^j, $\pi_{j,k} = 1 - \pi_{j,k-1}$, priors are needed only on $\pi_{j,k}$ when k is odd.
- Since the $\pi_{j,k}$s are probabilities, it is standard to use independent beta priors, specifically

$$\pi_{j,2k-1} \sim \text{Beta}(c\rho(j), c\rho(j)), \quad j = 1, \ldots, J, \quad k = 1, \ldots, 2^{j-1}.$$

- In most of the applications $\rho(j) = j^2$ as this guarantees an absolutely continuous F (in an infinite tree) (Ferguson, 1974).

Mixtures of finite Polya trees

Finite PT prior

- Before proceeding on further considerations about the role of c, let us note that under this parameterization
$$E[F(B(j,k))] = \frac{1}{2^j} = F_0(B_{j,k}).$$
- Thus, F_0 is the prior expectation of the unknown distribution function F.
- F_0 is usually selected based on our best prior assessment of the data-generating distribution F.
- The parameter $c > 0$, also referred to as the weight parameter, acts much like as the precision parameter α in the Dirichlet process.
- As in the Dirichlet process, large values of c leads to realizations of F that are close to F_0.
- A very low value of c (e.g., $c = 0.1$) will often lead to an estimate of F that is similar to the empirical cdf. Usually $c = 1$ works well in practice.
- Just like α, c can be regarded as random a prior placed on it.

Mixtures of finite Polya trees

Finite PT prior

- Once we have selected J, F_0, and c we have all the elements needed to specify a finite Polya tree for F.
- The formula for the density function f (our interest here) is given by

$$f(y) = 2^J p(B_{J,k(y)}) f_0(y). \tag{16}$$

- Here $k(y) \in \{1, \ldots, 2^J\}$ identifies the interval at level J containing y and $p(B_{J,k(y)})$ is the probability of that interval (the product of J of the $\pi_{j,k}$s) (Hanson, 2006).
- The interval that contains y at level J can be determined using the formula $k(y) = \text{Int}(2^J F_0(y) + 1)$.
- Note that the density at stage J is just the product of a weighting factor $2^J p(B_{J,k(y)})$ and the original used-specified parametric density f_0.
- Prior or posterior distributions that focus high probability on regions around $\pi_{jk} = 0.5$ for all j and k will behave very much like the f_0 density.

Mixtures of finite Polya trees

Finite PT prior

- Given Π, F is known.
- So, in order to compute posterior estimates of F (or f) we only need to know how to update the $\pi_{j,k}$s.
- Fortunately, just like the DP, Polya trees enjoy also a simple conjugacy result.
- Specifically, if

$$y_1, \ldots, y_n \mid F \overset{\text{iid}}{\sim} F, \quad F \sim \text{FPT}_J(c, F_0),$$

then $F \mid \mathbf{y}$ is updated through

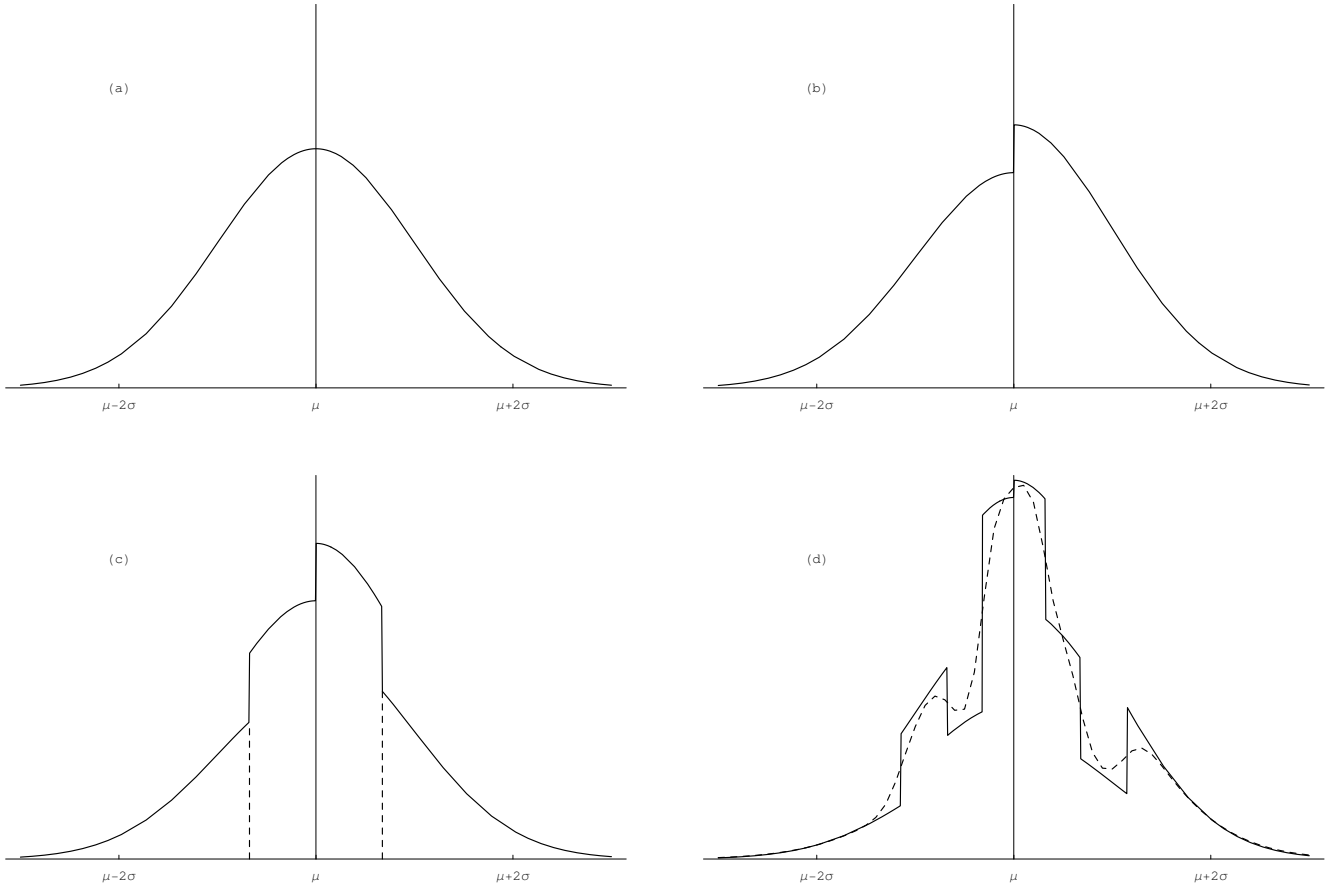
$$\pi_{j,2k-1} \mid \mathbf{y} \overset{\text{ind.}}{\sim} \text{Beta}\left(cj^2 + \sum_{i=1}^{n} I(y_i \in B_{j,k}), cj^2 + \sum_{i=1}^{n} I(y_i \in B_{j,k+1})\right), \tag{17}$$

for $j = 1, \ldots, J$ and $k = 1, \ldots, 2^{j-1}$.

- In words, we update the Beta parameters by counting the number of observations that fall in each set of each level of the tree.
- That is, $F \mid \mathbf{y}$ is a PT with Beta parameters updated through (17).

Mixtures of finite Polya trees

Finite PT prior: example



FPT density estimates considering $F_0 = \mathrm{N}(\mu, \sigma^2)$ and $J = 3$. (a) $\mathrm{N}(\mu, \sigma^2)$ density. (b) $j = 1$; $\pi_{1,1} = 0.45$. (c) $j = 2$; $\pi_{2,1} = 0.4$, $\pi_{2,3} = 0.6$. (d) J=3; $\pi_{3,1} = 0.3$, $\pi_{3,3} = 0.3$, $\pi_{3,5} = 0.6$, $\pi_{3,7} = 0.3$.

Mixtures of finite Polya trees

- All densities in the previous figure are too jagged, which turns out to be the result of using a fixed F_0.
- In fact, one of the major criticisms of Polya trees is that, unlike the DP, inferences are somewhat sensitive to the choice of a fixed partition.
- A remedy is to place a prior distribution on the parameters of F_0, say $\boldsymbol{\theta}$, we denote the resulting centering distribution as $F_{0,\boldsymbol{\theta}}$ to emphasize the dependence on $\boldsymbol{\theta}$.
- A prior on $\boldsymbol{\theta}$ implies that the starting and endpoints of the sets of the tree are uncertain/random.
- This has the effect of smoothing out the abrupt jumps at these points that are noticeable in the previous figure.
- In fact, in panel (d) of the previous figure it is also shown the estimate obtained by considering $\boldsymbol{\theta} = (\mu, \sigma^2)$ as random (dashed line).

Mixtures of finite Polya trees

- So, the final model is

$$
\begin{aligned}
y_1, \ldots, y_n \mid F &\overset{\text{iid}}{\sim} F, \\
F \mid c, \boldsymbol{\theta} &\sim \text{FPT}_J(F_{0,\boldsymbol{\theta}}, c), \\
\boldsymbol{\theta} &\sim p(\boldsymbol{\theta}),
\end{aligned}
$$

  and it is known as a mixture of finite Polya trees.

- It can be alternatively written as

$$
F \sim \int \text{FPT}_J(F_{0,\boldsymbol{\theta}}, c) p(\boldsymbol{\theta}) \mathrm{d}\boldsymbol{\theta}.
$$

- The formula for the density function f is identical to that given in (16).
- To conduct posterior inference we will now need to know how to sample $\boldsymbol{\theta} \mid \mathbf{y}, \Pi$ (we already know how to sample $\Pi \mid \mathbf{y}, \boldsymbol{\theta}$).
- We will make this concrete considering the particular case of $F_{0,\boldsymbol{\theta}} = \text{N}(\mu, \sigma^2)$.

Mixtures of finite Polya trees

- We center random F at $F_{0,\boldsymbol{\theta}} = \mathrm{N}(\mu, \sigma^2)$, where $\boldsymbol{\theta} = (\mu, \sigma^2)$.
- Using (16) the likelihood $L(\Pi, \boldsymbol{\theta}; \mathbf{y})$ is

$$\prod_{i=1}^{n} 2^J \phi(y \mid \mu, \sigma^2) p(k_{\boldsymbol{\theta}}(J, y_i)).$$

- We write $p(k_{\boldsymbol{\theta}}(J, y_i))$ instead of $p(B_{J,k(y_i)})$ to alleviate notation and to make clear the dependence on $\boldsymbol{\theta}$.
- As in Branscum et al. (2008), we assume $\mu \sim \mathrm{N}(a_\mu, b_\mu^2)$ and $\sigma \sim \Gamma(a_\sigma, b_\sigma)$.
- Assuming further that $\boldsymbol{\theta}$ and Π are a priori independent, the joint posterior density is proportional to

$$p(\boldsymbol{\theta}, \Pi \mid \mathbf{y}) \propto L(\Pi, \boldsymbol{\theta}; \mathbf{y}) p(\boldsymbol{\theta}) p(\Pi).$$

- The full conditionals for μ and σ are not recognizable as belonging to a parametric family thus these parameters are updated through Metropolis–Hastings steps.

Mixtures of finite Polya trees

MCMC

Algorithm

1. μ *is updated by sampling* $\mu^* \sim N(\mu, s_1)$ *and accepted with probability*

$$\min\left\{1, \frac{\exp\{-0.5b_\mu^{-2}(\mu^* - a_\mu)^2\}}{\exp\{-0.5b_\mu^{-2}(\mu - a_\mu)^2\}} \frac{\prod_{i=1}^n p(k_{\mu^*,\sigma}(J, y_i))}{\prod_{i=1}^n p(k_{\mu,\sigma}(J, y_i))} \times \frac{\exp\{-0.5\sigma^{-2}\sum_{i=1}^n (y_i - \mu^*)^2\}}{\exp\{-0.5\sigma^{-2}\sum_{i=1}^n (y_i - \mu)^2\}}\right\}.$$

Here s_1 *is a tuning parameter that needs to be calibrated to achieve a desirable acceptance rate.*

2. σ *is updated by sampling* $\sigma^* \sim \Gamma(\sigma s_2, s_2)$ *and accepted with probability*

$$\min\left\{1, \frac{f_\Gamma(\sigma^*; a_\sigma, b_\sigma)}{f_\Gamma(\sigma; a_\sigma, b_\sigma)} \frac{\prod_{i=1}^n p(k_{\mu,\sigma^*}(J, y_i))}{\prod_{i=1}^n p(k_{\mu,\sigma}(J, y_i))} \times \frac{\sigma^n \exp\{-0.5(\sigma^*)^{-2}\sum_{i=1}^n (y_i - \mu)^2\}}{\sigma^{*n} \exp\{-0.5\sigma^{-2}\sum_{i=1}^n (y_i - \mu)^2\}} \frac{f_\Gamma(\sigma; \sigma^* s_2, s_2)}{f_\Gamma(\sigma^*; \sigma s_2, s_2)}\right\},$$

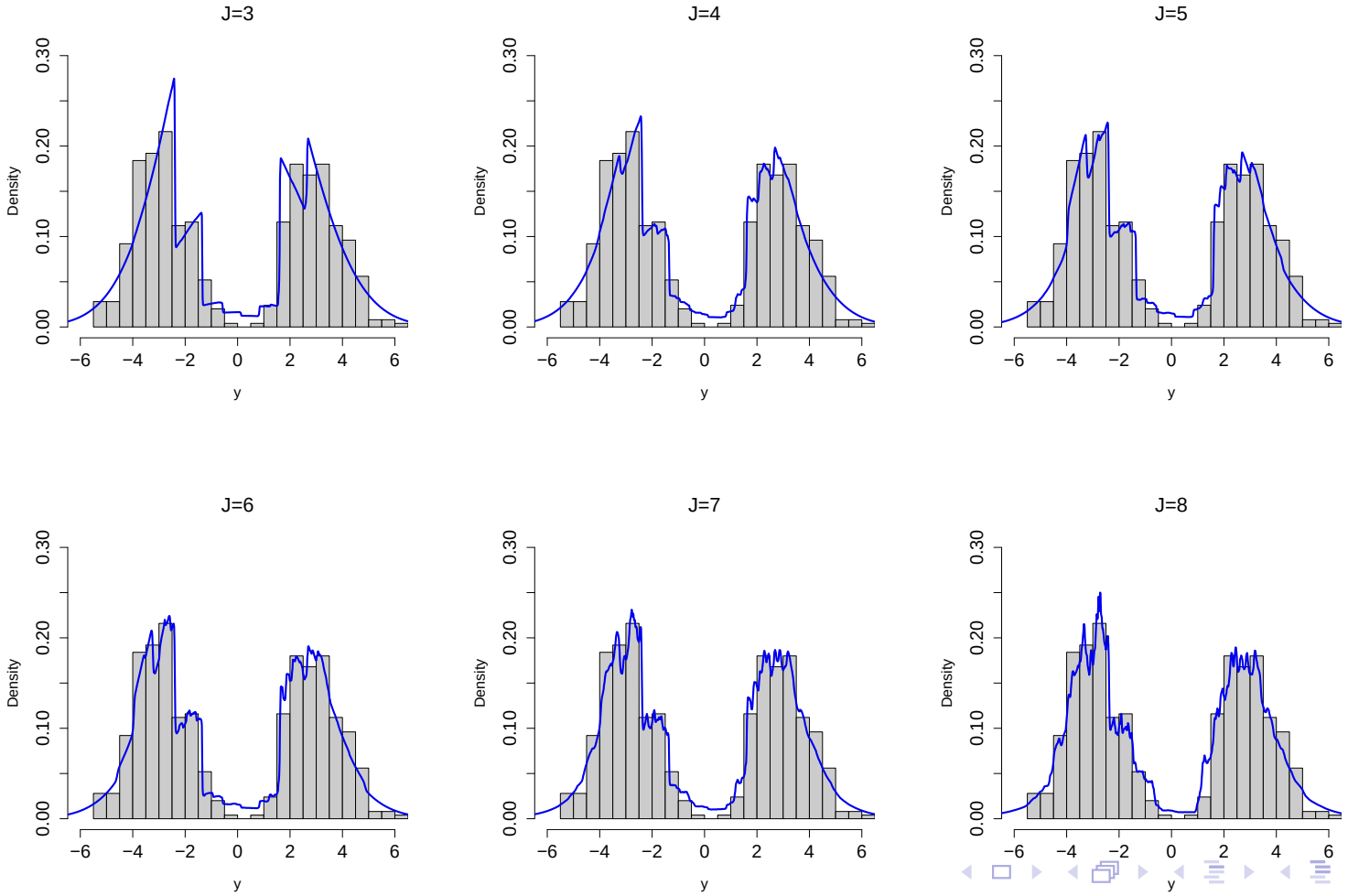
where s_2 *has the same meaning as* s_1*.*

3. *Use* (17) *to update* $\pi_{j,k}$*, for* $k = 1, \ldots, 2^{j-1}$ *and* $j = 1, \ldots, J$.

Mixtures of finite Polya trees

Example

Data generated from $0.5\phi(y \mid -3, 1) + 0.5\phi(y \mid 3, 1)$, $n = 500$.

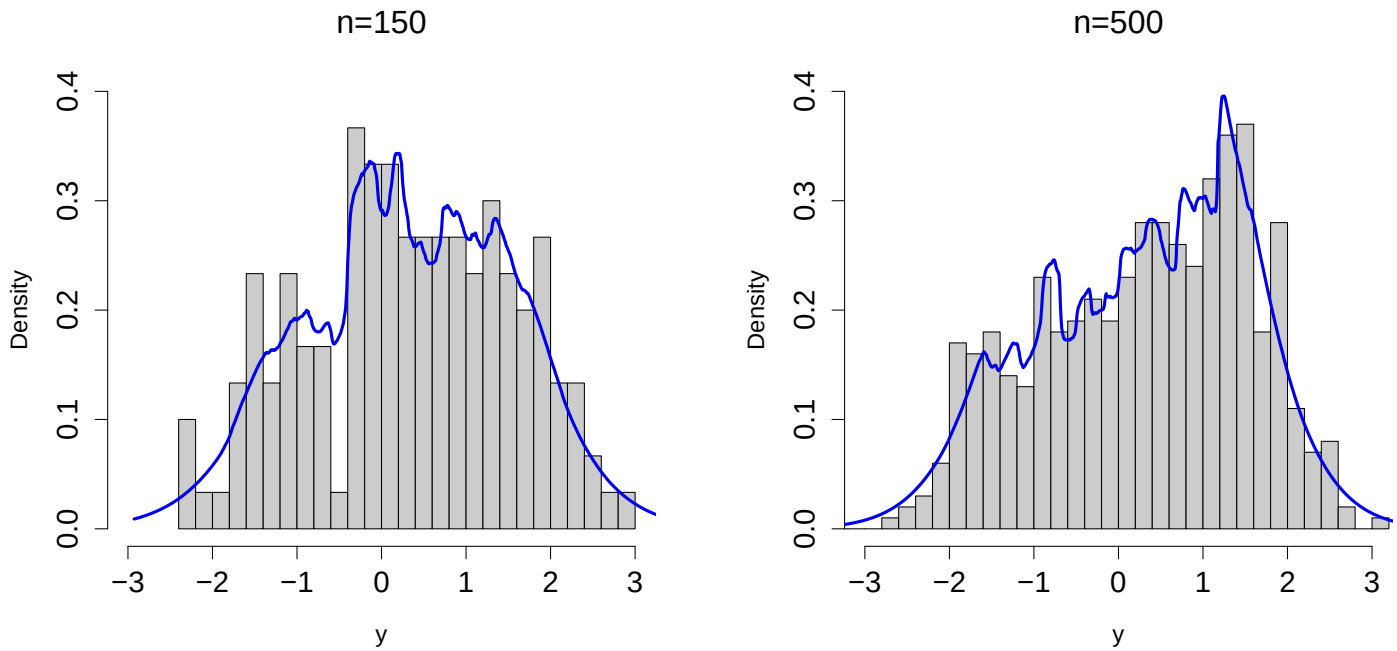


Mixtures of finite Polya trees

Example

True data generating mechanism:

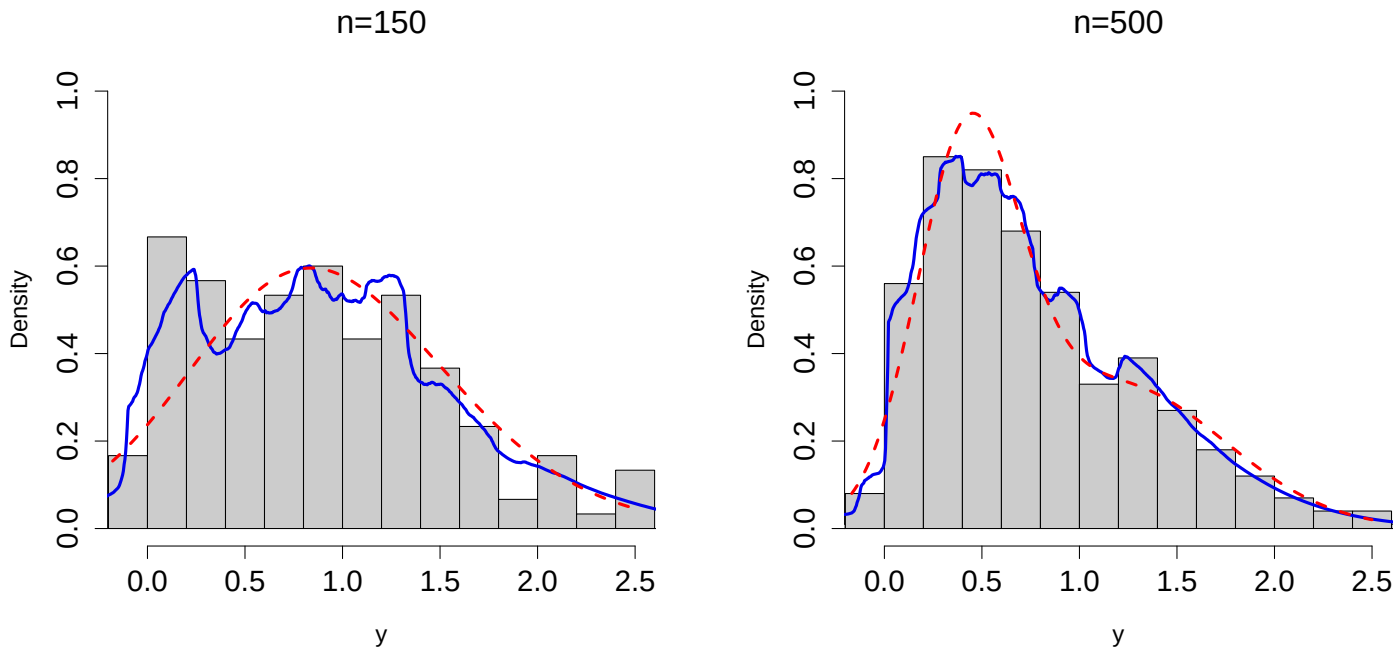
$0.3\phi_{SN}(y \mid \mu = -2, \sigma^2 = 1.5^2, \lambda = 6) + 0.7\phi_{SN}(y \mid \mu = 2, \sigma^2 = 1.5^2, \lambda = -3)$. $J = 4$ was considered.



Mixtures of finite Polya trees

Example

True data generating mechanism: $\phi_{\text{SN}}(y \mid \mu = 0, \sigma^2 = 1, \lambda = 10)$. MFPT estimate (solid blue line) agains DPM estimate (dashed red line).



Linear dependent tailfree process

- A Polya tree define the conditional probabilities $\pi_{j+1,2k-1}$, $\pi_{j+1,2k}$ as beta distributions.
- To accommodate covariates, and in a spirit of density regression, Jara and Hanson (2011) proposed to model these probabilities through logistic regression.
- Specifically, given covariates $\mathbf{x}$, the probabilities $(\pi_{j+1,2k-1}, \pi_{j+1,2k})$ are defined as

$$\log\left(\frac{\pi_{j+1,2k-1}}{\pi_{j+1,2k}}\right) = \mathbf{x}^T \boldsymbol{\tau}_{j,k}.$$

- The resulting model is known as linear dependent tail free process. For further details see Jara and Hanson, 2011.
- The function `LDTFPdensity` in `DPpackage` implements this model.

Other prior distributions

- Throughout this presentation we have focused on Dirichlet process mixtures and mixtures of finite Polya trees.
- We did not mean to be exhaustive. The aim was to provide, as the name says, an *introduction*.
- Other popular Bayesian nonparametric models include:
  - Gausian processes,
  - Bernstein polynomials,
  - Splines/ wavelets/ neural networks, etc.

Thank you...

...for your attention!!!