Go Nonparametric?!
An Introduction to Bayesian Nonparametrics (BNP)

Renate Meyer
Department of Statistics, University of Auckland, New Zealand
Bayes on the Beach, November 26, 2019

Introduction
BNP

References
Books:
- Ghosal and van der Vaart (2017)
- Müller, Quintana, Jara and Hanson (2015)
- Müller and Rodriguez (2013)
- Hjort, Holmes and Müller (2010)

Tutorials:
- Müller, Xu, Jara (2016)
  - https://users.soe.ucsc.edu/ thanos/
  - https://georgek.people.uic.edu/BayesSoftware.html
- lg.eng.cam.ac.uk/zoubin/talks/uai05tutorial-b.pdf

Software: R packages: DPpackage, dirichletprocess, msBP, JAGS, OpenBUGS, NIMBLE, Karabatsos’ Bayesian regression package, ...

Bayesian Nonparametric Model
- nonparametric does not mean NO parameters
- infinite unknown quantities
- infinite-dim. parameter is function or measure
- examples: unknown sampling distribution, cdf, density, link function in regression, hazard function, ...
- prior assigned to all unknown quantities: finite or infinite dimensional
Introduction

**BP or BNP**

**Parametric = Strong Prior Belief**

- $X_i \in \mathcal{X}$, $X_i | P \ iid \sim P$, $P \in \mathcal{P}^* = \{N(x|\mu, \sigma^2), \mu \in \mathbb{R}, \sigma^2 \in \mathbb{R}^+\}$

- $\mathcal{P}^*$ small compared to $\mathcal{P} = \{\text{all distributions on } \mathcal{X}\}$

- prior on $(\mu, \sigma^2)$ assigns probability one to tiny subset $\mathcal{P}^*$ of $\mathcal{P}$

- nonparametric prior on $P$

- What makes a *good* nonparametric model:
  - robust
  - interpretable
  - tractable

---

**Some Targets of Nonparametric Inference**

- Random Probability Measure – Density estimation
  
  Given $X_i \ iid \sim P$, $i = 1, \ldots, n$,
  need prior probability model for probability measure $P$

- Regression
  
  $Y_i \ ind \sim P_{x_i}$ for some covariates $x_i$

- Classification
  
  $Y_i | s_i = k \ iid \sim P_k$
  $s_i$ is cluster indicator

---

**Dirichlet Process**

**Dirichlet Process**

*Definition:* (Ferguson, 1973)

Let $G_0$ be a cdf on $\mathcal{X}$ and $\alpha > 0$.

The **Dirichlet Process** $DP(\alpha, G_0)$ with base measure $P_{G_0}$ and concentration parameter $\alpha$ generates random probability measure $P$ on ($\mathcal{X}, \mathcal{B}$) such that for any finite measurable partition $A_1, \ldots, A_k$ of $\mathcal{X}$

$$(P(A_1), \ldots, P(A_k)) \sim \text{Dirichlet}(\alpha P_{G_0}(A_1), \ldots, \alpha P_{G_0}(A_k))$$
Let $P \sim \text{DP}(\alpha, G_0)$ then for any $A \subseteq \mathcal{X}$:

- $E[P(A)] = P_{G_0}(A)$
- $G_0$ is the center of the DP
- $\text{Var}(P(A)) = \frac{P_{G_0}(A)(1 - P_{G_0}(A))}{1 + \alpha}$

$\alpha$ is a precision parameter

- The support of $\text{DP}(\alpha, G_0)$ contains all measures that are absolutely continuous wrt $G_0$

### Simulating sample paths from DP

When $\mathcal{X} = \mathbb{R}$:

Ferguson's definition to simulate cdf's $G$ of $\text{DP}(\alpha, G_0)$:

- Let $x_1 < x_2 < \ldots < x_k$ in $\mathcal{X} \subseteq \mathbb{R}$
- $(G(x_1), G(x_2) - G(x_1), \ldots, G(x_k) - G(x_{k-1}), 1 - G(x_k))$ has Dirichlet distribution with parameters $\alpha(G_0(x_1), G_0(x_2) - G_0(x_1), \ldots, G_0(x_k) - G_0(x_{k-1}), 1 - G_0(x_k))$
- Draw $(u_1, u_2, \ldots, u_k)$ from this Dirichlet distribution, then $(u_1, u_1 + u_2, \ldots, \sum_{j=1}^k u_j)$ is a draw of $(G(x_1), G(x_2), \ldots, G(x_k))$

### R-code: DPsimFerguson.R

```r
library(MCMCpack)
alpha <- 0.1
n <- 6
par(mfrow = c(2, 2))
k <- 10
x <- u <- matrix(0, nrow = n, ncol = (k + 2))
for (i in 1:n) {
  x[i, ] <- c(0, sort(runif(k))), 1)
  u[i, ] <- c(0, cumsum(rdirichlet(1, alpha * diff(x[i, ]))))
}
plot(stepfun(x[1, ][-1], u[1, ], col = "red", lty = 2, xlab = "x", ylab = "cdf",
            ylim = c(0, 1), main = "alpha=0.1", pch = NA)
plot(stepfun(x[2, ][-1], u[2, ], col = "blue", pch = NA)
plot(stepfun(x[3, ][-1], u[3, ], col = "purple", pch = NA)
plot(stepfun(x[4, ][-1], u[4, ], col = "green", pch = NA)
plot(stepfun(x[5, ][-1], u[5, ], col = "brown", pch = NA)
plot(stepfun(x[6, ][-1], u[6, ], col = "orange", pch = NA)
pants(x, x, type = "l")
# repeat for different alpha
```
Sethuraman Representation of DP

Constructive series representation of $\text{DP}(\alpha, G_0)$:

$$P \sim \text{DP}(\alpha, G_0) \iff P = \sum_{\ell=1}^{\infty} w_\ell \delta_{\theta_\ell}$$

with

- $\theta_\ell \overset{iid}{\sim} G_0$ for $\ell = 1, 2, \ldots$
- $z_r \overset{iid}{\sim} \text{Beta}(1, \alpha)$ for $r = 1, 2, \ldots$
- $w_1 = z_1$ and $w_\ell = z_\ell \prod_{r=1}^{\ell-1} (1 - z_r)$ for $\ell = 2, 3, \ldots$

Sethuraman (1994)

Simulating using Sethuraman Representation

For simulation, use truncated Sethuraman representation:

$$P = \sum_{\ell=1}^{L} \tilde{w}_\ell \delta_{\theta_\ell}$$

with $\tilde{w}_\ell = w_\ell$, $\ell = 1, \ldots, L - 1$ and $\tilde{w}_L = 1 - \sum_{\ell=1}^{L-1} w_\ell = \prod_{r=1}^{L-1} (1 - z_r)$

Because

$$E \left[ \sum_{\ell=1}^{L} w_\ell \right] = 1 - \prod_{\ell=1}^{L} E[1 - z_\ell] = 1 - \prod_{\ell=1}^{L} \frac{\alpha}{\alpha + 1} = 1 - \left( \frac{\alpha}{\alpha + 1} \right)^L$$

choose $L$ such that $\left( \frac{\alpha}{\alpha + 1} \right)^L < \epsilon$

R-code in DPsimSethuraman.R

```r
DPsim <- function(alpha = 1, rdistr = runif, L = 20) {
  z <- rbeta(L, 1, alpha)
  theta <- rdistr(L)
  w <- rep(0, L)
  w[1] <- z[1]
  keep <- (1 - z[1])
  for (l in 2:(L - 1)) {
    w[l] <- z[l] * keep
    keep <- keep * (1 - z[l])
  }
  w[L] <- 1 - sum(w[1:(L - 1)])
  return(list(w = w, theta = theta))
}
```

Kottas (2018)
### Dirichlet Process

#### Sampling from \( \text{DP}(\alpha, \text{Unif}(0,1)) \)

![Graphs showing sampling from different \( \alpha \) values]

#### RJAGS Implementation

```r
BUGSmodel <- "model

# Constructive DP
for (j in 2 : (L-1)) {
}
p[L] <- 1 - sum(p[1:(L-1)])
for (j in 1:L){
  theta[j] ~ dnorm(0,1)
  r[j] ~ dbeta(1, alpha)
}

#draw n iid samples from P~DP(alpha,G_0)
for (i in 1:n){
  S[i] ~ dcat(p[])
  x[i] <- theta[S[i]]
}
"
```

### RJAGS Output

Histogram of 1000 iid samples from \( \text{P} \sim \text{DP}(20, \text{N}(0,1)) \)

![Histogram of samples]

Histogram of 1000 iid samples from \( \text{P} \sim \text{DP}(20, \text{N}(0,1)) \)

![Histogram of samples]
Dirichlet Process DP Conjugacy

Conjugacy

The DP prior is conjugate for cdf estimation:

If \( X_i \mid P \sim P \) and \( P \sim \text{DP}(\alpha, G_0) \) then

\[
P(X_1, \ldots, X_n) \sim \text{DP}(\alpha_n, G_n)
\]

with

\[
\alpha_n = \alpha + n
\]

\[
G_n = \frac{\alpha}{\alpha + n} \times G_0 + \frac{n}{\alpha + n} \times \sum_{i=1}^{n} \delta_{X_i}
\]

\( \alpha \) measures degree of belief in prior guess \( G_0 \)

Estimation of Unknown Distribution

Generate \( X_i \overset{iid}{\sim} N(2,1) \), prior \( P \sim \text{DP}(25, N(0,1)) \)

R code in \( \text{DPsimpost.R} \)

\[
\text{DPsimpost} <- \text{function}(\alpha = 1, \text{rdistr} = \text{runif}, L = 20, n = 50, x = \text{rnorm}(n)) {
\begin{align*}
z &\leftarrow \text{rbeta}(L, 1, \alpha) \\
p &\leftarrow \alpha/(\alpha + n) \\
\theta &\leftarrow \text{rdistr}(L) \\
u &\leftarrow \text{runif}(L) \\
\text{for } (i \in 1:L) \{ \\
&\text{if } (u[i] > p) \\
&\quad \theta[i] \leftarrow \text{sample}(x, \text{size} = 1, \text{prob} = \text{rep}(1/n, n)) \\
&\}\w \leftarrow \text{rep}(0, L) \\
&w[1] \leftarrow z[1] \\
\text{keep} &\leftarrow (1 - z[1]) \\
\text{for } (l \in 2:(L - 1)) \{ \\
&\quad w[l] \leftarrow z[l] * \text{keep} \\
&\quad \text{keep} \leftarrow \text{keep} * (1 - z[l]) \\
&\}\w[L] \leftarrow 1 - \text{sum}(w[1:(L - 1)]) \\
\text{return} \left(\text{list}(w = w, \theta = \theta)\right)
\}
\]
\]

Estimates using \( \text{DP}(25,N(0,1)) \) prior and sample from \( N(2,1) \)

DP as a Pólya Urn Scheme

If \( X_i \overset{iid}{\sim} P \), \( P \sim \text{DP}(\alpha, G_0) \), then

1. \( X_1 \sim G_0 \)
2. \( X_2 \mid X_1 \) puts point mass \( 1/(\alpha + 1) \) at \( X_1 \) and continuous mass \( \alpha/(\alpha + 1) \) on \( G_0 \)
3. \( X_3 \mid X_1, X_2 \) puts point mass \( 1/(\alpha + 2) \) each at \( X_1, X_2 \) and continuous mass \( \alpha/(\alpha + 2) \) on \( G_0 \)

i.e. generalized Pólya urn scheme:

\[
X_i \mid X_1, \ldots, X_{i-1} \sim \begin{cases} 
\delta_{X_i}, & \text{with prob. } 1/(\alpha + i - 1) \\
\delta_{X_{i-1}}, & \text{with prob. } 1/(\alpha + i - 1) \\
\vdots & \vdots \\
\delta_{X_1}, & \text{with prob. } 1/(\alpha + i - 1) \\
P_{G_0}, & \text{with prob. } \alpha/(\alpha + i - 1)
\end{cases}
\]
Pólya urn scheme is like a Chinese Restaurant process:

A new guest in a Chinese restaurant joins a table with probability proportional to the no. of guests already sitting at the table, or takes a seat at a new table with probability proportional to $\alpha$.

Let $X_i | P \overset{iid}{\sim} P$, $i = 1, \ldots, n$

$P \sim \text{DP}(\alpha, G_0)$ and $G_0$ atomless

so $X_i | X_1, \ldots, X_{i-1}$, if drawn from $G_0$ in Pólya urn scheme, is distinct from $X_1, \ldots, X_{i-1}$.

Then the expected number $n^*$ of distinct values (as $n \to \infty$)

$$n^* \approx \alpha \log n$$

$\alpha$ controls the level of discreteness of the DP

Blackwell and MacQueen (1973)

Definition: $\{X_n\}_{n \in \mathbb{N}}$ is a Pólya$(\alpha, G_0)$ sequence if for any $B \subseteq \mathbb{R}$

$$P(X_1 \in B) = P_{G_0}(B)$$

$$P(X_{n+1} \in B) = \frac{\alpha}{\alpha + n} \times P_{G_0}(B) + \frac{n}{\alpha + n} \times \frac{1}{n} \sum_{i=1}^{n} \delta_{X_i}(B)$$

If $X_n$ is a Pólya$(\alpha, G_0)$ sequence, then

- $\frac{\alpha}{\alpha + n} \times P_{G_0} + \frac{n}{\alpha + n} \times \frac{1}{n} \sum_{i=1}^{n} \delta_{X_i} \overset{P}{\to} P$ a.s. for $n \to \infty$
- $P \sim \text{DP}(\alpha, G_0)$
- $X_i | P \overset{iid}{\sim} P$
Dirichlet Process Mixture (DPM) Models

- Posterior draws from DP are a.s. discrete distributions
- Not suitable to model continuous distributions
- Similar to kernel density estimation: smooth estimates
- Generalization of finite mixture models

From Finite to Infinite Mixture Models

- Write
  \[ w_1 N(y_i | \mu_1, \sigma_1^2) + \ldots + w_K N(y_i | \mu_K, \sigma_K^2) = \int N(y_i | \mu, \sigma^2) dG(\mu, \sigma^2) \]
  where
  \[ G = w_1 \delta(\mu_1, \sigma_1^2) + \ldots + w_K \delta(\mu_K, \sigma_K^2) \]
- \( G \) is discrete and random
- Instead: use a DP prior for \( G \)
  - known as DPM
  - yields a countably infinite mixture
  - number \( K \) of mixture components is estimated from data

DPM\( (k, \alpha, G_0) \) Model

Let \( K(\cdot|\theta) \) parametric cdf with density \( k(\cdot|\theta) \)

- \( F(\cdot|G) = \int K(\cdot|\theta) dG(\theta) \) with \( G \sim DP(\alpha, G_0) \)
- \( f(\cdot|G) = \int k(\cdot|\theta) dG(\theta) \) with \( G \sim DP(\alpha, G_0) \)

Using the Sethuraman representation of \( G \sim \sum_{\ell=1}^{\infty} w_\ell \delta_{\theta_\ell} \), the DPM density is countable mixture of parametric densities:

\[ f(\cdot|G) = \sum_{\ell=1}^{\infty} w_\ell k(\cdot|\theta_\ell) \]

- weights \( w_1 = z_1, w_\ell = z_\ell \prod_{r=1}^{\ell-1} (1 - z_r), z_\ell \overset{iid}{\sim} Beta(1, \alpha) \)
- parameters: \( \theta_\ell \overset{iid}{\sim} G_0 \)
**Simulation of DPM**

**Simulation of DPM** $(N(\cdot, \theta, 0.25), 2, N(0, 1))$

R code in `DPMsimSethuraman.R`

```r
par(mfrow = c(2, 3))
alpha <- 2
L = 20
out <- DPsim(alpha = alpha, rdistr = rnorm, L = 20)
w <- out$w
theta <- out$theta
thw <- cbind(theta, w)
thwinc <- thw[order(thw[, 1]), ]
plot(stepfun(thwinc[, 1], c(0, cumsum(thwinc[, 2]))), xlab = "x", ylab = "cdf",xlim = c(-5, 5), ylim = c(0, 1), main = "draw from DP(2,N(0,1))", pch = NA, lty = 2)
x <- seq(-5, 5, length = 1000)
y <- w[1] * dnorm(x, mean = theta[1], sd = 0.5)
z <- w[1] * pnorm(x, mean = theta[1], sd = 0.5)
for (l in 2:L) {
y <- y + w[l] * dnorm(x, mean = theta[l], sd = 0.5)
z <- z + w[l] * pnorm(x, mean = theta[l], sd = 0.5)
}
plot(x, z, type = "l", main = "Draw from DPM F(.|G)"
plot(x, y, type = "l", main = "Draw from DPM f(.|G)"
```

**Hierarchical Representation of DPM**

Alternative representation with latent variables $\theta_i$:

\[
Y_i | \theta_i, \phi, G \sim \text{ind}_i \sim k(\cdot | \theta_i, \phi)
\]

\[
\theta_i | G \sim \text{id}_i \sim G
\]

\[
G \sim \text{DP}(\alpha, G_0(\cdot | \psi))
\]

\[
\alpha \sim \rho(\alpha)
\]

\[
\psi \sim \rho(\psi)
\]

\[
\phi \sim \rho(\phi)
\]

DPMs can be used as priors for
- discrete distributions (e.g. Poisson kernel)
- continuous univariate (e.g. Normal kernel)
- continuous multivariate (e.g. multivariate Normal kernel)

**Prior Specification**

Use (by taking the expectation over $G$ wrt $\text{DP}(\alpha, G_0)$)

\[
E[F(\cdot | G)] = F(\cdot | G_0)
\]

\[
E[f(\cdot | G)] = f(\cdot | G_0)
\]

the hyperparameter of $G_0(\cdot | \psi)$ can be chosen accordingly

If $\alpha \approx 0$: one distinct component in mixture
If $\alpha \approx \infty$: $n$ distinct components in mixture

$\alpha$ controls the prior distribution of the number of distinct mixture components: approx. $\alpha \log(1 + n/\alpha)$ (Ghosal, van der Vaart, 2017)
Posterior Computation

\[ p(G, \theta, \alpha, \psi, \phi | y) = p(G | \theta, \alpha, \psi)p(\theta, \alpha, \psi, \phi | y) \]

- \( G | \theta, \alpha, \psi \sim \text{DP}(\alpha, G_\alpha) \) where
  \[ \begin{align*}
  \alpha_n &= \alpha + n \\
  G_n &= \frac{\alpha}{\alpha + n} G_0 + \frac{1}{\alpha + n} \sum_{i=1}^{n} \delta_{\theta_i}
  \end{align*} \]

- \( p(\theta, \alpha, \psi, \phi | y) \) usually difficult to sample from

Various methods based on Gibbs sampling, see e.g. Section 5.2 of Ghosal and van der Vaart (2017)

OpenBUGS/JAGS (Congdon, 2001), DPpackage (Jara et al. 2011), dirichletprocess (Ross and Markwick, 2019)

Applications of DP and DPM

Used as building block with many applications:
- density estimation
- linear mixed models, generalized linear mixed models, longitudinal data
- longitudinal clustering
- regression modeling
- nonparametric ANOVA
- binary and ordinal data
- errors-in-variables models
- meta analysis
- time series modeling

Density Estimation

DPM for Density Estimation

Let \( y_{n+1} \) denote a new observation, \( y = (y_1, \ldots, y_n) \).

The Bayesian density estimate is the posterior predictive density:

\[ p(y_{n+1} | y) = E[f(y_{n+1} | G) | y] \]

\[ = \int k(y_{n+1} | \theta_{n+1}) p(\theta_{n+1} | w, \theta, \alpha) p(w, \theta, \alpha | y) d\theta_{n+1} dw d\alpha \]

Using the Polya urn scheme:

\[ p(\theta_{n+1} | w, \theta, \alpha) = \frac{\alpha}{\alpha + n} \delta_0(\theta_{n+1}) + \frac{1}{\alpha + n} \sum_{i=1}^{n} \delta_0(\theta_{n+1}) \]

MCMC scheme: Given a posterior sample \((w_t, \theta_t, \alpha_t)\) for \( t = 1, \ldots, T \):

- draw \( \theta_{n+1, t} \sim p(\theta_{n+1} | w_t, \theta_t, \alpha_t) \)
- draw \( y_{n+1, t} \sim k(\cdot | \theta_{n+1, t}) \)

Velocities (km/s) for \( n = 82 \) galaxies, drawn from 6 well-separated conic sections of the Corona Borealis region.

Location-scale DPM of Gaussians with conjugate Normal-Inverse Gamma baseline

\[
f(y|\mathcal{G}) = \int N(y|\mu, \sigma^2) dG(\mu, \sigma^2), \quad G \sim DP(\alpha, G_0)
\]

with

\[
G_0(\mu, \sigma^2) = N(\mu|m_1, \sigma^2_0) \times IG(\sigma^2|\nu_1, \nu_2)
\]

\( \kappa | \tau_1, \tau_2 \sim Gamma\left(\frac{\tau_1}{2}, \frac{\tau_2}{2}\right) \)

Implementation in DPpackage using `DPdensity` with \( m_1 = 0, \nu_1 = 2, \nu_2 = 1, \tau_1 = 1, \tau_2 = 100 \)

**R Code for Galaxy Example**

Recall: \( n^* \approx \alpha \log(1 + \frac{n}{\alpha}) \)

```r
data(galaxy)
attach(galaxy)
speeds <- speed/1000
state = NULL
mcmc = list(nburn = 1000, nsave = 10000, nskip = 10, ndisplay = 100)
prior1 = list(alpha = 1, m1 = 0, psiinv1 = 0.5, nu1 = 4, tau1 = 1, tau2 = 100)
fit.1 = DPdensity(y = speeds, prior = prior1, mcmc = mcmc, state = state, status = TRUE)
prior2 = list(alpha = 0.1, m1 = 0, psiinv1 = 0.5, nu1 = 4, tau1 = 1, tau2 = 100)
fit.2 = DPdensity(y = speeds, prior = prior2, mcmc = mcmc, state = state, status = TRUE)
prior3 = list(alpha = 3, m1 = 0, psiinv1 = 0.5, nu1 = 4, tau1 = 1, tau2 = 100)
fit.3 = DPdensity(y = speeds, prior = prior3, mcmc = mcmc, state = state, status = TRUE)
prior4 = list(a0 = 0.1, b0 = 0.1, m1 = 0, psiinv1 = 0.5, nu1 = 4, tau1 = 1, tau2 = 100)
fit.4 = DPdensity(y = speeds, prior = prior4, mcmc = mcmc, state = state, status = TRUE)
```

**Galaxy Data, DPM Density Estimates**
The DPM induces a probability model on clusters:

- DP is discrete $\rightarrow$ ties among latent $\theta_i$:
  - Let $\theta^*_k$ be the $n^* \leq n$ distinct values, $k = 1, \ldots, n^*$
  - $S_k = \{i : \theta_i = \theta^*_k\}, n_k = |S_k|
  - $S_1, \ldots, S_{n^*}$ is a random partition of $\{1, \ldots, n\}$
  - DP induced prior probability on $S_1, \ldots, S_{n^*}$:
    $$\mathbb{P}(S_1, \ldots, S_{n^*}|\alpha) = \frac{\alpha^* \prod_{k=1}^{n^*} (n_k - 1)!}{(\alpha + 1) \ldots (\alpha + n - 1)}$$

see e.g. Chapter 8 of Müller et al (2015)

Random Effects

Random Effects

Introduction

Dirichlet Process

DP Mixtures

Density Estimation

Random Effects

Regression

Model Based Clustering

Based on finite mixture model:

$$Y_i|C_i = k, \theta^*_k \sim k(\cdot|\theta^*_k) \quad k = 1, \ldots K$$

$$\mathbb{P}(C_i = k) = w_k$$

If $\theta^*_k \sim G_0$ and $(w_1, \ldots, w_K) \sim \text{Dir}(\alpha/K, \ldots, \alpha/K)$
then as $K \rightarrow \infty$ model based clustering is equivalent to a DPM

$$Y_i|G \sim \int k(\cdot|\theta) dG(\theta) \quad \text{with} \quad G \sim \text{DP}(\alpha, G_0)$$

Linear Random Effects Models

Often used for repeated measurements $Y_{ij}, j = 1, \ldots, n_i, i = 1, \ldots, n$

$$Y_i = X_i \beta + Z_i b_i + \epsilon_i, \quad i = 1, \ldots, n$$

where

- $Y_i$ response vector for $i$-th subject
- $X_i, Z_i$ covariate matrices
- $\beta$ fixed effects regression parameter
- $\epsilon_i \sim N(0, \sigma^2 I)$ vector of errors
- $b_i$ vector of random effects
- $b_i \sim N(0, D)$

But: random effects often have multimodal distribution!
Bayesian Parametric Random Intercept Model

Special case:
\[ Y_{ij} = \mu + \theta_i + \epsilon_{ij}, \quad \theta_i \sim N(0, \tau^2), \quad \epsilon_{ij} \sim N(0, \sigma^2) \]
with priors:
\[ \mu \sim N(\mu_0, \kappa^2) \quad \sigma^2 \sim IG(a, b) \quad \tau^2 \sim IG(c, d) \]

Is the Gaussian assumption for the random effects valid?

Normal distribution is chosen for convenience rather than genuine prior belief. Covariates that have not been taken into account can cause multimodalities, outliers, skewness.

Bayesian Semiparametric Random Effects Model

General:
\[ Y_{ij} | \beta, b_i, \sigma^2 \sim N(X_i\beta + Z_i b_i, \sigma^2 I) \quad i = 1, \ldots, n \]
\[ b_i | G \sim G \quad i = 1, \ldots, n \]
\[ G | \alpha, D \sim DP(\alpha, N(0, D)) \]
\[ \beta, \sigma^2, \alpha, D \sim p(\beta, \sigma^2, \alpha, D) \]

Example: Growth Curves of Schoolgirls

Verbeke and Molenberghs (2000): growth curves of \( n = 20 \) schoolchildren, \( n_i = 5 \) height measurements from age 6–10

- height: numeric, height in cm
- child: unique identifier for subject \( i \)
- age: numeric, age of child in years

Figure: Individual growth curves of schoolgirls
Random Effects | DP and LMM
---|---
**Fitting in DPpackage**

With $y_{ij} = j$-th height of $i$-th child and $x_{ij} = \text{corresponding age}$

\[
Y_{ij} | (\theta_i, b_i), \sigma^2 \sim N(\theta_i + x_{ij}b_i, \sigma^2)
\]

\[
(\theta_i, b_i) | G \sim G
\]

\[
G \sim DP(\alpha, N(\mu, \Sigma))
\]

\[
\mu \sim N(\mu_b, S_b)
\]

\[
\Sigma \sim IW(\nu_0, T)
\]

Using `DPlmm` function of DPpackage:

\[
DPlmm(fixed=\text{height} \sim 1, \text{random}=\text{~age|child}, \text{prior}=\text{prior}, \text{mcmc}=\text{mcmc}, \text{state}=\text{state}, \text{status}=\text{FALSE})
\]

Note: this example is included in the help file of `DPlmm`

---

Random Effects | DP and LMM
---|---
**Output**

**Model's performance:**

<table>
<thead>
<tr>
<th>Dbar</th>
<th>Dhat</th>
<th>pD</th>
<th>DIC</th>
<th>LPML</th>
</tr>
</thead>
<tbody>
<tr>
<td>248.54</td>
<td>223.61</td>
<td>24.93</td>
<td>273.47</td>
<td>-141.53</td>
</tr>
</tbody>
</table>

**Regression coefficients:**

<table>
<thead>
<tr>
<th>(Intercept)</th>
<th>age</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td>Median</td>
</tr>
<tr>
<td>82.5162</td>
<td>82.5114</td>
</tr>
<tr>
<td>5.7245</td>
<td>5.7201</td>
</tr>
</tbody>
</table>

**Residual variance:**

<table>
<thead>
<tr>
<th>residual</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
</tr>
<tr>
<td>0.7154</td>
</tr>
</tbody>
</table>

**Baseline distribution:**

<table>
<thead>
<tr>
<th>mu-(Intercept)</th>
<th>sigma-(Intercept)</th>
<th>sigma-(Intercept)-age</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td>Median</td>
<td>Std. Dev.</td>
</tr>
<tr>
<td>82.7044</td>
<td>82.7099</td>
<td>0.9013</td>
</tr>
<tr>
<td>4.3884</td>
<td>4.3884</td>
<td>0.4384</td>
</tr>
<tr>
<td>0.1150</td>
<td>0.1150</td>
<td>0.9746</td>
</tr>
<tr>
<td>1.5221</td>
<td>1.5221</td>
<td>1.3086</td>
</tr>
</tbody>
</table>

**Precision parameter:**

<table>
<thead>
<tr>
<th>ncluster</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
</tr>
<tr>
<td>8.4200</td>
</tr>
</tbody>
</table>

**Number of Observations:** 100

**Number of Groups:** 20

---

Random Effects | DP and LMM
---|---
**R code**

Code in `schoolgirlsDPlmm.R`

```r
data(schoolgirls)
attach(schoolgirls)

# Prior information
prior <- list(alpha=1, nu0=4.01, tau1=0.01, tau2=0.01,
              tinv=diag(10,2), mub=rep(0,2), Sb=diag(1000,2))

# Initial state
state <- NULL

# MCMC parameters
mcmc <- list(nburn=5000, nsave=10000, nskip=20, ndisplay=1000)

# Fit the model: First run
fit1 <- DPlmm(fixed=height~1, random=~age|child, prior=prior, mcmc=mcmc, state=state, status=TRUE)
summary(fit1)

dPrandom(fit1)
plot(fit1, ask=FALSE, nfigr=1, nfigc=2, param="ncluster")
plot(DPrandom(fit1), ask=TRUE)
```

---

Random Effect information for the DP object:

Call:

`DPlmm.default(fixed = height ~ 1, random = ~age | child, prior = prior, mcmc = mcmc, state = state, status = TRUE)`

Posterior mean of subject-specific components:

<table>
<thead>
<tr>
<th>(Intercept)</th>
<th>age</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>81.144</td>
</tr>
<tr>
<td>2</td>
<td>81.092</td>
</tr>
<tr>
<td>3</td>
<td>81.387</td>
</tr>
<tr>
<td>4</td>
<td>81.579</td>
</tr>
<tr>
<td>5</td>
<td>83.100</td>
</tr>
<tr>
<td>6</td>
<td>81.140</td>
</tr>
<tr>
<td>7</td>
<td>83.319</td>
</tr>
<tr>
<td>8</td>
<td>82.692</td>
</tr>
<tr>
<td>9</td>
<td>83.558</td>
</tr>
<tr>
<td>10</td>
<td>81.579</td>
</tr>
<tr>
<td>11</td>
<td>82.622</td>
</tr>
<tr>
<td>12</td>
<td>81.581</td>
</tr>
<tr>
<td>13</td>
<td>81.519</td>
</tr>
<tr>
<td>14</td>
<td>85.060</td>
</tr>
<tr>
<td>15</td>
<td>83.469</td>
</tr>
<tr>
<td>16</td>
<td>82.590</td>
</tr>
<tr>
<td>17</td>
<td>83.502</td>
</tr>
<tr>
<td>18</td>
<td>83.610</td>
</tr>
<tr>
<td>19</td>
<td>81.761</td>
</tr>
<tr>
<td>20</td>
<td>83.496</td>
</tr>
</tbody>
</table>

---

Random Effects | DP and LMM
---|---
**Output**

---
Random Effects  DPM and LMM

Output

Bayesian Semiparametric Random Effects Model with DPM

General:

\[ Y_{ij} | \beta, b_i, \sigma^2 \sim N(X_{ij} \beta + Z_{ij} b_i, \sigma^2 I) \quad i = 1, \ldots, n \]

\[ b_i | G, \Sigma \sim \int N(b_i | \mu, \Sigma) dG(\mu) \quad i = 1, \ldots, n \]

\[ G(\alpha, \mu_b, \Sigma_b) \sim DP(\alpha, N(\mu_b, \Sigma_b)) \]

\[ \beta, \sigma^2, \alpha, D \sim p(\beta, \sigma^2, \alpha, D) \]

Random Effects  DPM and LMM

Schoolgirls Example

With \( y_{ij} \) = \( j \)th height of \( i \)-th child and \( x_{ij} \) = corresponding age

\[ Y_{ij} | (\theta_i, b_i), \sigma^2 \sim N(\theta_i + x_{ij} b_i, \sigma^2) \]

\[(\theta_i, b_i) | a_i \sim N(a_i, \Sigma) \]

\[ a_i | G \sim G \]

\[ G(\mu, \Sigma) \sim DP(\alpha, N(\mu, \Sigma)) \]

\[ \mu \sim N(\mu_b, Sb) \]

\[ \Sigma \sim IW(\nu_0, T) \]

Using DPMlmm function of DPpackage:

DPMlmm(fixed=height ~ 1, random=~ age|child, prior=prior, mcmc=mcmc, state=state, status=FALSE)

Note: this example is included in the help file of DPMlmm

Random Effects  DPM and LMM

Output

Model's performance:

<table>
<thead>
<tr>
<th>Dbar</th>
<th>Dhat</th>
<th>pD</th>
<th>DIC</th>
<th>LPML</th>
</tr>
</thead>
<tbody>
<tr>
<td>214.15</td>
<td>180.69</td>
<td>33.46</td>
<td>247.62</td>
<td>-130.79</td>
</tr>
</tbody>
</table>

Regression coefficients:

<table>
<thead>
<tr>
<th>(Intercept)</th>
<th>Mean</th>
<th>Median</th>
<th>Std. Dev.</th>
<th>Naive Std.Error</th>
<th>95%HPD-Low</th>
<th>95%HPD-Upp</th>
</tr>
</thead>
<tbody>
<tr>
<td>82.460079</td>
<td>82.470357</td>
<td>0.733714</td>
<td>0.007337</td>
<td>81.104328</td>
<td>83.975548</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>age</th>
<th>Mean</th>
<th>Median</th>
<th>Std. Dev.</th>
<th>Naive Std.Error</th>
<th>95%HPD-Low</th>
<th>95%HPD-Upp</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.71585</td>
<td>5.718871</td>
<td>0.269035</td>
<td>0.002690</td>
<td>5.184606</td>
<td>6.225978</td>
<td></td>
</tr>
</tbody>
</table>

Residual variance:

<table>
<thead>
<tr>
<th>residual</th>
<th>Mean</th>
<th>Median</th>
<th>Std. Dev.</th>
<th>Naive Std.Error</th>
<th>95%HPD-Low</th>
<th>95%HPD-Upp</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.507086</td>
<td>0.496389</td>
<td>0.097299</td>
<td>0.000973</td>
<td>0.333554</td>
<td>0.702218</td>
<td></td>
</tr>
</tbody>
</table>

Kernel variance:

<table>
<thead>
<tr>
<th>(Intercept)</th>
<th>Mean</th>
<th>Median</th>
<th>Std. Dev.</th>
<th>Naive Std.Error</th>
<th>95%HPD-Low</th>
<th>95%HPD-Upp</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.493393</td>
<td>4.848922</td>
<td>3.048031</td>
<td>0.030484</td>
<td>0.912542</td>
<td>11.502018</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>age</th>
<th>Mean</th>
<th>Median</th>
<th>Std. Dev.</th>
<th>Naive Std.Error</th>
<th>95%HPD-Low</th>
<th>95%HPD-Upp</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.150676</td>
<td>-0.111753</td>
<td>0.551287</td>
<td>0.055128</td>
<td>-1.365566</td>
<td>0.899699</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>sigma-age</th>
<th>Mean</th>
<th>Median</th>
<th>Std. Dev.</th>
<th>Naive Std.Error</th>
<th>95%HPD-Low</th>
<th>95%HPD-Upp</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.833220</td>
<td>0.777575</td>
<td>0.294365</td>
<td>0.029436</td>
<td>0.379347</td>
<td>1.409258</td>
<td></td>
</tr>
</tbody>
</table>

Baseline distribution:

<table>
<thead>
<tr>
<th>(Intercept)</th>
<th>Mean</th>
<th>Median</th>
<th>Std. Dev.</th>
<th>Naive Std.Error</th>
<th>95%HPD-Low</th>
<th>95%HPD-Upp</th>
</tr>
</thead>
<tbody>
<tr>
<td>82.00767</td>
<td>82.19174</td>
<td>2.762564</td>
<td>0.02763</td>
<td>77.45301</td>
<td>87.11786</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>age</th>
<th>Mean</th>
<th>Median</th>
<th>Std. Dev.</th>
<th>Naive Std.Error</th>
<th>95%HPD-Low</th>
<th>95%HPD-Upp</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.68397</td>
<td>5.69705</td>
<td>3.048031</td>
<td>0.030484</td>
<td>0.912542</td>
<td>11.502018</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>sigma-age</th>
<th>Mean</th>
<th>Median</th>
<th>Std. Dev.</th>
<th>Naive Std.Error</th>
<th>95%HPD-Low</th>
<th>95%HPD-Upp</th>
</tr>
</thead>
<tbody>
<tr>
<td>8.11346</td>
<td>3.95120</td>
<td>30.23247</td>
<td>0.3023</td>
<td>0.51281</td>
<td>12.14806</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>sigma-age</th>
<th>Mean</th>
<th>Median</th>
<th>Std. Dev.</th>
<th>Naive Std.Error</th>
<th>95%HPD-Low</th>
<th>95%HPD-Upp</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.05560</td>
<td>-0.0181</td>
<td>11.84700</td>
<td>0.11847</td>
<td>-9.73176</td>
<td>10.59203</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>sigma-age</th>
<th>Mean</th>
<th>Median</th>
<th>Std. Dev.</th>
<th>Naive Std.Error</th>
<th>95%HPD-Low</th>
<th>95%HPD-Upp</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.59765</td>
<td>2.98700</td>
<td>11.61949</td>
<td>0.11619</td>
<td>0.38185</td>
<td>16.92056</td>
<td></td>
</tr>
</tbody>
</table>

Precision parameter:

<table>
<thead>
<tr>
<th>ncluster</th>
<th>Mean</th>
<th>Median</th>
<th>Std. Dev.</th>
<th>Naive Std.Error</th>
<th>95%HPD-Low</th>
<th>95%HPD-Upp</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2254</td>
<td>2.0000</td>
<td>1.1500</td>
<td>0.0115</td>
<td>1.0000</td>
<td>4.0000</td>
<td></td>
</tr>
</tbody>
</table>

Note: this example is included in the help file of DPMlmm
A response variable $Y_i$ is generated from some probability distribution $F_{x_i}$ indexed by covariates $x_i$.

**Parametric regression model:**

e.g. $F_{x_i} = N(\beta^T x_i, \sigma^2)$, the sampling model is

$$Y_i = f_\beta(x_i) + \epsilon_i, \quad \epsilon_i \sim N(0, \sigma^2)$$

with $f_\beta(x_i) = \beta^T x_i$ and $\theta = (\beta, \sigma^2)$

BNP does not restrict $F_{x}$ to parametric family

BNP can relax assumptions on:

- the residual distribution
- the centering function $f_\beta(x)$
- both

**Nonparametric Residual Distribution**

parametric mean function and nonparametric error distribution:

$$Y_i = f_\beta(x_i) + \epsilon_i, \quad \epsilon_i \sim F, \text{ BNP prior on } F$$

Posterior Inference based on blocked Gibbs sampling:

- conditional on $F$: sample $\beta$
- conditional on $\beta$: estimate density of $\epsilon_i = Y_i - f_\beta(x_i)$
Regression
Nonparametric Residual Distribution

Example: Old Faithful Data

Mueller et al. (2015): duration and intervals between subsequent eruptions of the geyser from August 1-15, 1985

\[ x_i = \text{centered durations and } Y_i = \text{centered intervals} \]

\[ Y_i = f_i(x_i) + \epsilon_i \]

polynomial regression function:

\[ f(x_i) = \beta_1 x_i + \beta_2 x_i^2 + \beta_3 (x_i^3)^+ \]

DPM prior on residuals:

\[ \epsilon_i | F \overset{iid}{\sim} F \]

\[ F | G \sim \int N(\mu, \sigma^2 = 9) dG(\mu) \]

\[ G \sim \text{DP}(\alpha_0 = 1, G_0 = N(0, 15^2)) \]

Code: https://web.ma.utexas.edu/users/pmueller/bnp/

Data and fitted regression curve, with posterior draws of \( f \) (grey), Code: OldFaithfulMueller.R

---

Regression
Nonparametric Mean Function

Example: Old Faithful Data

nonparametric mean function and parametric error distribution:

\[ Y_i = f(x_i) + \epsilon_i, \quad \epsilon_i \sim F_\nu \]

General approach is based on expansion wrt basis functions \( \{\phi_k\} \):

\[ f(\cdot) = \sum_{k=1}^{K} \theta_k \phi_k(\cdot) \]

with prior on \( \theta_k \)'s

\( K = \infty \) for truly nonparametric prior, truncated in practice
Examples of basis functions are:
- wavelets
- sines/cosines
- Legendre or Bernstein polynomials
- B-splines/P-splines for generalized additive models
- neural networks

These are essentially parametric Bayesian models.

Posterior inference is simplified by
- orthonormal basis
- equally spaced data and
- independent normal residuals

Software for posterior computation:
R, JAGS, PSgam, BayesX

Lenk (1999): $Y_i \sim N(f_0(x_i), \sigma^2)$

$$f_0(x_i) = \mu + \beta_0 x_i + \sum_{k=1}^{K} \beta_k \cos \left( \frac{\pi (x_i - x(1))}{x(n) - x(1)} \right)$$

Prior:
- $\mu \sim N(0, 10000)$
- $\beta_0 \sim N(0, \tau^2)$
- $\beta_k \sim N(0, \tau^2/j)$
- $\sigma^{-2} \sim \text{Gamma}(a, b)$
- $\tau^{-2} \sim \text{Gamma}(c, d)$

Code from [https://web.ma.utexas.edu/users/pmueller/bnp/](https://web.ma.utexas.edu/users/pmueller/bnp/)
OldFaithfulCosines.R

Design matrix with row $i$:
$$[B_1(x_i), \ldots, B_K(x_i)]$$

Implementation with PSgam in DPpackage

```r
fit <- PSgam(formula=interval ~ ps(duration, k=20, degree=3, pord=1),
              family=gaussian(), prior=prior,
              mcmc=mcmc, ngrid=30,
              state=state, status=TRUE)
```

fits a sum of $K$ cubic Bsplines with 30 equally spaced knots using a first order penalty

Code: in OldFaithfulPSGam.R

Posterior mean of regression function with 95% credible band.
A Gaussian process defines a distribution $P(f)$ over functions, $f$, where $f: \mathcal{X} \to \mathbb{R}$.
Let $f(x_1), \ldots, f(x_n)$ be an $n$-dimensional vector of function values.

**Definition:** $P(f)$ is a **Gaussian process (GP)** if for any finite subset $\{x_1, \ldots, x_n\} \subset \mathcal{X}$ the marginal distribution of $P(f)$ has a multivariate Gaussian distribution.

GPs are parametrized by a mean function, $\mu(x)$ and a covariance matrix function, $\Sigma(x, x')$

$$P(f(x), f(x')) = N(\mu, \Sigma)$$

where

$$\mu = \begin{bmatrix} \mu(x) \\ \mu(x') \end{bmatrix}$$

$$\Sigma = \begin{bmatrix} \Sigma(x, x) & \Sigma(x, x') \\ \Sigma(x', x) & \Sigma(x', x') \end{bmatrix}$$

The GP prior is a BNP prior for a random mean function of Gaussian data.

$$Y_i = f(x_i) + \epsilon_i$$

$$f \sim GP(\cdot | \mu, \Sigma)$$

$$\epsilon_i \sim N(0, \sigma^2)$$

For Gaussian residuals, the posterior of $f$ is also a GP.

The predictive distributions are multivariate Gaussian.

Let $X$ be a design matrix with $m$ columns:

$$Y | \beta, \sigma^2, K \sim N_0(X \beta, \sigma^2 K)$$

$$\beta | \beta_0, \tau^2 W \sim N_0(\beta_0, \tau^2 W)$$

$$\beta_0 \sim N_0(\mu, \Sigma)$$

$$W^{-1} \sim \text{Wishart}(\rho, (\rho V)^{-1})$$

$$\sigma^2 \sim IG(a/2, b/2)$$

$$\tau^2 \sim IG(d/2, d/2)$$
Covariance of Gaussian Process Prior

If the covariance $K$ is identity: Gaussian linear model

In general $K$:

- isotropic power family for $0 < \rho \leq 2$:
  \[
  K(x_j, x_k|d) = \exp \left\{ -\frac{||x_j - x_k||^\rho}{d} + \nu \delta_{jk} \right\}
  \]

- separable power family:
  \[
  K(x_j, x_k|d) = \exp \left\{ -\frac{\sum_{i=1}^m |x_{ij} - x_{ik}|^\rho}{d_i} + \nu \delta_{jk} \right\}
  \]

- Matern family

Posterior computations can be demanding, if $K$ is high-dimensional (matrix inversion required)

Gramacy and Lee (2008) implemented GP and developed treed GPs

bgp and btgp in R package tgp

Codes in OldFaithfulGaussianProcess.R

see also: https://web.ma.utexas.edu/users/pmueller/bnp/R/ch4.html

library(tgp)

d = read.table("geyser.dat", header = FALSE)
duration = d[, 1]
interval = d[, 2]
plot(duration, interval, pch = 19, bty = "l", xlab = "DURATION", ylab = "INTERVAL")
n = length(duration)
a = min(duration)
b = max(duration)  ## grid for prediction
pred = seq(a, b, length = 100)
fit = bgp(X = duration, Z = interval, XX = pred, verb = 2)
fit.treed = btgp(X = duration, Z = interval, XX = pred, verb = 2)

GP regression with 90% prediction intervals and quantile difference plots.

Treed GP regression with 90% prediction intervals and quantile difference plots.
Generic regression problem:

\[ Y_i | X_i \sim P_{x_i} \]

with family of probability measures \( P = \{ P_x; x \in X \} \)

Complete model with BNP prior on \( P \).

Two main approaches:

1. Dependent Dirichlet Process (DDP), MacEachern (1999)

2. Density Regression or Conditional Regression, Müller et al. (1996), Park and Dunson (2010)

---

**Linear Dependent Dirichlet Process (LDDP)**

special case: \( P_x \) Gaussian DPM:

\[
P_x = \int N(\cdot | \mu, \sigma^2) dG_x(\mu) = \sum w_i N(\cdot | \mu_i(x), \sigma^2)
\]

with \{ \( G_x \) \} \( \sim \) DDP.

Now if \( x \) is a vector of covariates and \( \mu_x(x) = x^T \beta_x \) with \( \beta_x \sim N(\mu_\beta, S_\beta) \) and \( \sigma^2 \sim \text{Gamma}(\tau_1/2, \tau_2/2) \)

this is called a **LDDP model** (Jara and Hanson, 2011).

Implementation in DPpackage using LDDPdensity
Density regression of (interval) on duration of previous eruption

Code: OldFaithfulLDDP.R
see also https://web.ma.utexas.edu/users/pmueller/bnp/R/ch4.html

```r
n = length(duration)
a = min(duration)
b = max(duration)  # save min and max for below
# evaluate B-splines --
X = cbind(rep(1, n), bs(duration, df = 6, Boundary.knots = c(a, b)))

## set up prior
m = solve(t(X) %*% X) %*% t(X) %*% interval
s2 = sum((interval - X %*% m)^2) / (n - length(m))
S = solve(t(X) %*% X) * s2
taul = 5 + 0.5 * s2
taus1 = (taul - 2) * s2 / (2 + taul - 8 - s2)
taus2 = 2
nu = length(m) + 2
prior = list(a0 = 1, b0 = 1, m0 = m, S0 = S, taul = taul, taus1 = taus1, taus2 = taus2,
            nu = nu, psilinv = 25 * S)
```

```
## set up cov's (that is, B-splines) for prediction prediction
## pred=c(quantile(duration,0.25),quantile(duration,0.75))
pred = seq(a, b, length = 50)
Xpred = cbind(rep(1, length(pred)), predict(bs(duration, df = 6, Boundary.knots = c(a, b)), pred))
mcmc = list(nburn = 1000, nsave = 5000, nskip = 4, ndisplay = 100)

## fit LDDP for B-spline coefficients
fitLDDP = LDDPdensity(formula = interval ~ X - 1, zpred = Xpred, ngrid = 100,
                      prior = prior, mcmc = mc当地，state = NULL, status = TRUE, compute.band = TRUE)
```

LDDP predictive density estimates at first and third quartile of duration

Mean of predictive regression function with 95% credible band.
Generic regression problem:
\[ Y_i | x_i \overset{\text{ind}}{\sim} G_{x_i} \]
with family of probability measures \( G = \{ G_x; x \in X \} \)

Complete model with BNP prior on \( G \).

Two main approaches:
1. **Dependent Dirichlet Process (DDP)**, MacEachern (1999)
2. **Density Regression** or **Conditional Regression**, Müller et al. (1996), Park and Dunson (2000)

Specifically: Gaussian DPM:
\[
(X_i, Y_i) | (\mu_i, \Sigma_i) \overset{\text{ind}}{\sim} N(\mu_i, \Sigma_i)
\]

\[
(\mu_i, \Sigma_i) | G \sim G
\]

\[
G \sim \text{DP}(\alpha, G_0)
\]

If \( \theta_k^* = (\mu_k^*, \Sigma_k^*) \) unique values of \( \theta_i \)'s with multiplicities \( n_k \),
\( g_0(y|x) \) is conditional normal density and
\( s_0(x) \) the marginal normal density
then the predictive density
\[
p(y|x, \theta_1^*, \ldots, \theta_K^*) \propto \alpha s_0(x) g_0(y|x) + \sum_{k=1}^{K} n_k s(x|\theta_k^*) g(y|x, \theta_k^*)
\]
i.e. a locally weighted mixture of linear regressions

Posterior computation **DPcdensity** in **DPpackage**
Bayesian nonparametric models and semiparametric mixture models provide an enormous flexibility.

Widespread applications

Data determines balance between complexity and simplicity

Challenges and Hurdles:

- consistency and convergence rates for non-iid models
- development of flexible software
- computational costs for temporal and spatial models, e.g. Argiento and Ruggiero (2018)
- scalability for big data (large $n$, large $p$) via approximation, parallel processing, e.g. Dunson (2018), Ni et al. (2019)

Thank you for your attention!
References: III

Bayesian semiparametric median regression modeling.
JASA 96, 1458-1468

Bayesian inference for semiparametric regression using a Fourier representation.
JRSSB 61: 863-879

Dependent nonparametric processes
ASA Proceedings of the Section on Bayesian Statistical Science

Bayesian curve fitting using multivariate normal mixtures.
Biometrika 83, 67-79

Nonparametric Bayesian Inference
IMS-CBMS Lecture notes, IMS, Beachwood

References: IV

Bayesian Nonparametric Data Analysis.
Springer Series in Statistics

A short tutorial on Bayesian nonparametrics.
Journal of Statistical Research 48-50,1-19

Scalable Bayesian Nonparametric Clustering and Classification
JCGS

Bayesian generalized product partition models
Statistica Sinica 20, 1203-1226

dirichletprocess: An R package for Fitting Complex Bayesian Nonparametric Models
Preprint

References: V

Probabilistic programming in Python using PyMC3.
PeerJ Computer Science 2:e55

Linear Mixed Models for longitudinal data.
Springer, NY