An Introduction to Dirichlet Process Priors for Random Effects and an Extension to Model-Based Clustering[‡]

JEFF GILL
Distinguished Professor
Department of Government, Department Mathematics & Statistics
Member, Center for Behavioral Neuroscience
Founding Director, Center for Data Science
American University

‡This Work Supported by NSF Grants: DMS-0631632 and SES-0631588

Dirichlet Process Mixtures Models On Random Effects

- The usual random effects model:

$$\mathbf{Y}|\psi \sim \mathcal{N}(\mathbf{X}\beta + \psi, \sigma^2 I), \quad \psi_j \sim \mathcal{N}(0, \tau^2)$$

  where $\mathbf{X}$ is $(n \times p)$, and $\mathbf{Y}$, $\boldsymbol{\psi}$ are $(n \times 1)$ (the number of unique ψ_j is less than n).

- But subject-specific random effects can be from non-viewable groupings.

- The Dirichlet Process Random Effects Model we employed in the past for this problem:

$$\mathbf{Y}|\psi \sim \mathcal{N}(\mathbf{X}\beta + \psi, \sigma^2 I), \quad \psi_j \sim \mathcal{DP}(\lambda, G_0)$$

- Note that we are *not* putting Bayesian *nonparametric* priors on the regression coefficients.

- From this approach we observed:
  - Fewer Parametric Assumptions
  - More Accurate Estimates
  - Shorter Credible Intervals

Some Background Definitions

- Y is a random variable taking values on the measurable space $(\mathcal{Y}, \mathfrak{B})$, defined by the support of Y and an arbitrary (for now) abstract space $\mathfrak{B}$.
- The "parameter" of interest here is P, the associated, but *unknown*, probability measure taking values in $\mathcal{P}$, the collection of *all* probability measures on $(\mathcal{Y}, \mathfrak{B})$.
- Define $\mathcal{S}$ as the smallest σ-field (closed under countable unions) generated by sets of the form:

$$\{P\colon P(A) < r\}, \qquad \text{for all } A \in \mathfrak{B}, \text{ and } r \in [0:1].$$

- Now define ν as a probability measure on $(\mathcal{P}, \mathcal{S})$, which can be used as a prior distribution for the unknown P.
- We are interested in computing ν^*, the posterior distribution of $P|Y$.
- ν is called a Dirichlet Measure if for every measurable partition $\{B_1, \ldots, B_K\}$ (and finite K) of the parameter space $\mathfrak{B}$, the distribution of $P(B_1), \ldots, P(B_K)$ under ν is Dirichlet:

$$f(P(\mathbf{B})|\alpha_1, \ldots, \alpha_K) \propto P(B)_1^{\alpha_1 - 1} \cdots P(B)_K^{\alpha_K - 1},$$

$$0 \leq P(B)_i \leq 1, \textstyle\sum_{i=1}^{K} P(B)_i = 1, 0 < \alpha_i, \forall i \in [1, 2, \ldots, K].$$

The Distributional Structure

- Ferguson (1973, 1974, 1983) and Antoniak (1974) introduced the Dirichlet process prior for non-parametric G, which is this random probability measure on the space of all measures.
- We notate this distribution conventionally over the space of distributions by:
  - G_0, a base distribution (finite non-null measure) which is analogous to an "expected value" of the distributions,
  - $\lambda > 0$, a concentration/precision parameter (finite and non-negative scalar) giving the spread of distributions around G_0,
  - therefore $\phi_0 = \lambda G_0$ is a base measure,
  - leading to the prior specification $G \sim \mathcal{DP}(\lambda, G_0) \in \mathcal{P}$ (in the collection of all probability measures on $(\mathcal{Y}, \mathfrak{B})$).
- For *any* finite partition of the parameter space, $\{B_1, \ldots, B_K\}$, the joint distribution of these probabilities has the Dirichlet distribution, now according to:

$$\{\mathcal{G}(B_1), \ldots, \mathcal{G}(B_K) \sim \mathcal{D}(\lambda G_0(B_1), \ldots, \lambda G_0(B_K)\},$$

where for some *observed* partition, these are just multinomial probabilities.

Setting Up the Estimation Process

- Since realizations of the $\mathcal{DP}$ select a discrete distribution with probability one (even though the generating mechanism is continuous), the model for the random effect vector $\boldsymbol{\psi}$ is a countably infinite mixture (some key papers: Ferguson 1973, Antoniak 1974, Berry & Christensen 1979, Lo 1984, Escobar & West 1995, MacEachern & Müller 1998).
- Blackwell and MacQueen (1973) noted the following (generally, not just for random effects):
  - If G is a $\mathcal{DP}$, where $\psi_1, \ldots, \psi_n$ iid from $G \sim \mathcal{DP}(\lambda, G_0)$,
  - then the marginal distribution of $\psi_1, \ldots, \psi_n$ (marginalized over any prior parameters) is equal in distribution to the first n steps of a Pólya process.
- Blackwell and MacQueen then proved that the joint distribution of the $\boldsymbol{\psi}$ is a product of *successive* conditional distributions of the form:

$$\psi_i | \psi_1, \ldots, \psi_{i-1} \sim \frac{\lambda}{i-1+\lambda}\, \phi_0(\psi_i) + \frac{1}{i-1+\lambda} \sum_{l=1}^{i-1} \delta(\psi_i = \psi_l),$$

  where δ denotes the Dirac delta function.
- Therefore reference can be made to finite rather than infinite dimensions, and Dirichlet process posterior calculations involve a single parameter over this space (Ferguson's Theorem 1, 1973).

Review of the Pólya Process

- The Pólya Process for sampling ψ is equivalent to the following permutation scheme:
  - a restaurant has many large circular tables.
  - n diners enter one-at-a-time to be seated, where the first person sits at the first table.
  - For a given weight, λ, the ith person sits at the unoccupied ith table with probability $\lambda/(i-1+\lambda)$.
  - Otherwise this diner selects the jth ($j < i$) *previously occupied* table with probability $n_j/(i-1+\lambda)$, where n_j is the number seated at that table already.
- Now the table locations of the seated diners, $\xi_1, \ldots, \xi_n$, is a dependent exchangeable sequence (Blackwell and MacQueen 1973).
- $\xi^* = (\xi_1, \ldots, \xi_k)$ with $k \leq n$, the set of non-empty tables, is a *sample* from G.
- This process can be iterated many times to numerically integrate over this space in an MCMC context.
- The MCMC iteration is an integration process and therefore integrates of all possible G.

Models and Likelihood

- A general *random effects Dirichlet Process* model can now be written definitionally as:

$$(Y_1, \ldots, Y_n) \sim f(y_1, \ldots, y_n \mid \theta, \psi_1, \ldots, \psi_n) = \prod_i f(y_i|\theta, \psi_i), \quad \psi_i \sim \mathcal{DP}(\lambda, G_0), \quad i = 1, \ldots, n$$

  (the vector θ here is a placeholder for all of other the estimated parameters, $\mathbf{X}$ assumed).

- Applying the successive conditional distributions of Blackwell and McQueen, we integrate over the random effects to get the likelihood function:

$$L(\theta \mid \mathbf{y}) = \int\cdots\int f(y_1, \ldots, y_n \mid \theta, \psi_1, \ldots, \psi_n)\pi(\psi_1, \ldots, \psi_n)\, d\psi_1 \cdots d\psi_n$$

$$= \frac{\Gamma(\lambda)}{\Gamma(\lambda + n)} \sum_{k=1}^{n} \lambda^k \left[\sum_{C:|C|=k} \prod_{j=1}^{k} \Gamma(n_j) \int_\Psi f(\mathbf{y}_{(j)} \,|\theta,\ \psi_j)\phi_0(\psi_j)\, d\psi_j \right]$$

  where the second form is derived in Lo (1984 Annals) Lemma 2 and Liu (1996 Annals), and:

  - C is a partition of the sample of size n into k groups, $k = 1, \ldots n-1$
  - $\mathbf{y}_{(j)}$ is the vector of y_is in *sub*cluster (henceforth "bundle") j
  - ψ_j is the common random effects parameter applied to that bundle.

Matrix Representation of Partitions

- Since every "diner" at a given table gets the same random effects value, we want an efficient way to keep track of assignments on each cycle of the sampler.
- Associate a binary matrix $A_{n\times k}$ with a given partition C, for example:

$$C = \{S_1, S_2, S_3\} = \{\{1,2\},\{3,4,6\},\{5\}\} \leftrightarrow A = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

- Rows: a_i is a $1 \times k$ vector of all zeros except for a 1 in its bundle.
- Columns: The column sums of A are the number of observations in the groups
- Variables: thus $\psi_i \in S_j \Rightarrow \psi_i = \eta_j$ (constant in bundless), reducing the parameter space for estimation.
- This is similar to (but different from) the matrix approach in McCullagh and Yang (2006).

Mapping Partitions to the Underlying Random Effects

- Continuing with the contrived example:

$$C = \{S_1, S_2, S_3\} = \{\{1,2\},\{3,4,6\},\{5\}\} \leftrightarrow A = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

- This leads to the matrix representation:

$$\boldsymbol{\psi} = A\boldsymbol{\eta} \quad \text{where } A = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix} \quad \text{so} \quad \begin{pmatrix} \psi_1 \\ \psi_2 \\ \vdots \\ \psi_6 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} \eta_1 \\ \eta_2 \\ \eta_3 \end{pmatrix}.$$

- So we only need to generate three random variables in the sampler.

Incorporating the A Matrix

- Return to:

$$\mathbf{Y}|\psi \sim \mathcal{N}(\mathbf{X}\beta + \psi, \sigma^2 I), \text{ where } \psi_i \sim \mathcal{DP}(\lambda, \mathcal{N}(0, \tau^2)), \quad i = 1, \ldots, n$$

  where we are explicitly averaging over all normals with mean zero as our DPP choice.

- Introduce the A matrices to get

$$\mathbf{Y}|\mathbf{A}, \eta \sim \mathcal{N}(\mathbf{X}\beta + A\eta, \sigma^2 I), \quad \eta \sim N_k(0, \tau^2 I),$$

  meaning that η is now the focus of the Bayesian nonparametric process.

- If we could marginalize over these η (we can) and the A were known, it would be the case that:

$$\mathbf{Y}|\mathbf{A} \sim \mathcal{N}(\mathbf{X}\beta, \Sigma^*), \qquad \Sigma^* = \left(I + \frac{\tau^2}{\sigma^2} AA'\right)$$

  since the DPP is applied to the random effects only.

Dirichlet Process Prior Bundles **Are Not** Clusters

- An unfortunately common strategy is to use DPP models to generate a very large number of candidate "clusters," which are actually bundles, then choose the best of these by a post-hoc scheme that processes the MCMC output through some objective function to find the best grouping.
- *This is wrong.*
- The supposed-clusters (bundles) produced by the MCMC process in repeated realizations of the Dirichlet process are:
  - not substantive in any way,
  - not able to reflect any real cluster structure driven by the covariates,
  - temporary random effect assignments to make the model fit better in the context of the sampler.
- Since there is *no over-fitting penalty in the Dirichlet process*, we can expect there to always be more bundles than actual substantive clusters in the data.
- Therefore we seek to complement the DPP modeling approach just described with a feature that leads to the simultaneous estimation of real clustering in the data with a *product partition model*.

Mixture and Product Partition Models

- The standard mixture model begins with the assumption that $Y_1, \ldots, Y_n$ are realizations of independent and identically distributed (iid) random variables with m-component mixture weights applied, giving the density:

$$f(\mathbf{y}|\boldsymbol{\Theta}) = \prod_{i=1}^{n} \sum_{\ell=1}^{m} \pi_\ell \, f(y_i|\theta_\ell) \; ,$$

  where $m < n$ is a *fixed* positive integer, $0 \leq \pi_\ell \leq 1$, $\sum_{\ell=1}^{m} \omega_\pi = 1$, and θ_ℓ is a mixture parameter.

- An alternative, the product partition model, starts by conditioning on a given partition, and then determines the posterior probabilities of these.

- Given a partition $\mathbb{N}_n := \{1, 2, \ldots, n\}$, $\mathcal{C}$ that has $m < n$ clusters denoted by $\mathcal{C}_1, \ldots, \mathcal{C}_m$, the data are a realization from a density of the form:

$$f(y|\boldsymbol{\beta}_{\mathcal{C}}, \boldsymbol{\sigma}^2_{\mathcal{C}}, \mathcal{C}) = \prod_{\ell=1}^{m} \prod_{i \in \mathcal{C}_\ell} f(y_i|\boldsymbol{\beta}_\ell, \sigma^2_\ell) \; .$$

- So unlike the mixture model, this model recognizes a parameter, $\mathcal{C}$, that is directly connected to the basic clustering problem and is part of the estimation process.

- This model was developed by Hartigan (1990) (see also Barry & Hartigan 1992, Crowley 1997).

Reasons Not to Prefer the Standard Mixture Model *for Clustering*

- Parameterization: the mixture model lacks a *model parameter* that defines the clusters, which can confound standard estimation processes (McCullagh & Yang 2008, Booth, Casella and Hobert 2008).
- Cluster Identification: even if the mixture parameters of the model are known, there needs to be some way of generating a latent variable to identify clusters (McLachlan & Peel 2004).
- Ad Hoc Selection: the final model needs to be run with a fixed m, with the typical strategy running a user-defined selection of m values and choosing the one with the best BIC, or similar criteria (Si and Reiter 2013).
- Label Switching: the mixture model is prone to the label switching problem for Bayesian specifications: nonidentifiability of the parameter posteriors under symmetric priors (Jasra, Holmes & Stephens 2005, Stephens 2000, Celeux 1998). There are $m!$ permutations of the m mixture components, and every MCMC iteration is a sample from permutation invariant posterior distributions. So unless the markov Chain visits every permutation with equal frequency (unlikely) the posterior results are meaningless.

Reasons To Prefer the Product Partition Model *for Clustering*

- Computation: the product partition model partition process can be predictor-dependent and computationally efficient (Park and Dunson 2010).
- Model Dimensionality: a stochastic search algorithm can be setup to move between different size partitions at each iteration of a sampler (Booth, Casella and Hobert 2008).
- Cluster Identification: Contrary to the mixture model, the product partition model clearly identifies the parameter that determines the cluster, and has no restriction on m, the number of clusters, other than $m < n$ (Crowley 1997).
- Label Switching: since the product partition model is label-free (the clusters are all defined by unique partitions of $\mathbb{N}_n = \{1, 2, \ldots, n\}$), we can easily identify mappings of cases to clusters (Hartigan 1990, Barry & Hartigan 1992).

Substantive Clustering Strategy

- *In addition to the DPP component for random effects* we search for partitions of $\mathbf{Y}$ into clusters $\mathcal{C}_\ell, \ell = 1, \ldots, m$, where m (the number of clusters) is an unknown parameter.
- Let $\mathbf{Y}_\ell$ be a vector of length n_ℓ containing the Y_i in cluster $\mathcal{C}_\ell$, then:

$$\mathbf{Y}_\ell = \mathbf{X}_\ell \boldsymbol{\beta}_\ell + \mathbf{A}_\ell \boldsymbol{\eta}_\ell + \boldsymbol{\epsilon}_\ell$$

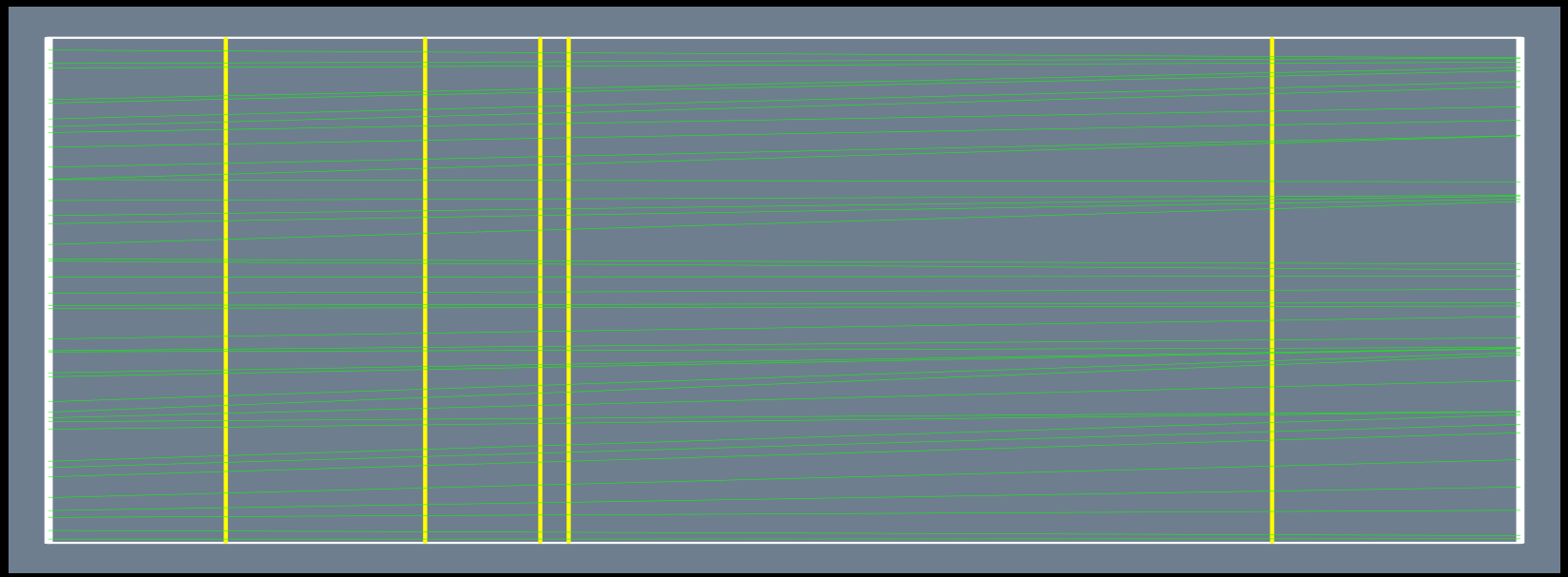
  where:

  - $\mathbf{X}_\ell$ and $\mathbf{A}_\ell$ are composed of the rows corresponding to the Y_i in cluster $\mathcal{C}_\ell$,
  - unknown $\boldsymbol{\beta}_\ell$ and σ^2_ℓ (where $\boldsymbol{\epsilon}_\ell \sim \mathcal{N}\left(0, \sigma^2_\ell I_{n_\ell}\right)$) are specific to cluster $\mathcal{C}_\ell$.

- This model accounts for heterogeneity in the data in two distinct ways:
  - it utilizes $\mathcal{DP}$ random effects to model unobserved heterogeneity in the data via DPP bundles.
  - the product partition model, using $\mathcal{C}$, provides substantive clusters to the data that serve to provide insights into how that data can be broken into groups that have different behavior.
- Note that these groupings *do not nest*, and so observations in the same cluster $\mathcal{C}_\ell$ can belong to different bundles defined by the columns of $\mathbf{A}$ (unlike Hartigan and Barry 1992).

Substantive Clustering Strategy

- Our goal is to find the best partition $\mathcal{C} = (\mathcal{C}_1, \ldots, \mathcal{C}_m)$, but the $\mathbf{A}$ matrix defining k bundles cannot be ignored.
- Using the DPP we want to find the posterior probability of $\mathcal{C}$, marginalized over the coefficients and random effects, which requires both integration over $\boldsymbol{\eta}$ and summation over the $\mathbf{A}$ matrices.
- Note that use of the $\mathcal{DP}$ random effects produces a correlation between individuals both within the same bundles and in different clusters, a non-nested hierarchical specification.



Cluster Prior Probabilities

- Each $\boldsymbol{\beta}_\ell$ is given a multilevel model structure with common underlying mean $\boldsymbol{\beta}_0$ and locally scaled precision matrix $\mathbf{S}$:

$$\boldsymbol{\beta}_\ell \sim \mathcal{N}\left(\boldsymbol{\beta}_0, \sigma_\ell^2 \mathbf{S}^{-1}\right).$$

- Each cluster-specific variance parameter σ_ℓ^2 is assigned an inverse-gamma prior with common assigned hyperparameters:

$$\sigma_\ell^2 \sim \mathcal{IG}\left(\frac{a_{\sigma^2}}{2}, \frac{b_{\sigma^2}}{2}\right).$$

- The remaining assigned priors have the forms:

$$\text{DP:} \quad \phi_0 \sim N\left(0, \tau^2\right) \qquad \tau^2 \sim \mathcal{IG}\left(\tfrac{a_{\tau^2}}{2}, \tfrac{b_{\tau^2}}{2}\right) \qquad \lambda \sim G\left(\tfrac{a_\lambda}{2}, \tfrac{b_\lambda}{2}\right)$$

$$\text{PP:} \quad \boldsymbol{\beta}_0 \sim \mathcal{N}(0, \sigma_{\boldsymbol{\beta}}^2 \mathbf{S}^{-1}) \qquad \sigma_{\boldsymbol{\beta}}^2 \sim \mathcal{IG}\left(\tfrac{a_{\sigma_{\boldsymbol{\beta}}^2}}{2}, \tfrac{b_{\sigma_{\boldsymbol{\beta}}^2}}{2}\right) \qquad \mathbf{S} \sim \mathcal{W}(V^{-1}, a_{\mathbf{S}})$$

$$V = \text{Diag}(v_1, \ldots, v_p) \qquad v_i \sim G\left(\tfrac{a_v}{2}, \tfrac{b_v}{2}\right) \qquad \boxed{\mathcal{C} \sim ???}$$

Outcome Variable Distributions

- The scaling of $\mathbf{S}$ (the precision parameter on $\boldsymbol{\beta}_\ell$ terms) produces a multivariate-t sampling distribution assumption:

$$\mathbf{Y}_\ell|\mathbf{X}_\ell, \boldsymbol{\beta}_\ell, \mathbf{S}, \mathbf{A}_\ell, \boldsymbol{\eta}, \mathcal{C} \sim \mathcal{MVT}_{n_\ell}\left(\mathbf{X}_\ell\boldsymbol{\beta}_\ell + \mathbf{A}_\ell\boldsymbol{\eta}, \left(I_{n_\ell} + \mathbf{X}_\ell\mathbf{S}^{-1}\mathbf{X}_\ell'\right)\frac{b_{\sigma^2}}{a_{\sigma^2}}\right)$$

  for outcome variable in the ℓth cluster.

- The above result along with prior specifications for the cluster-specific parameters $\boldsymbol{\beta}_\ell, \sigma^2_\ell$ provides the *complete* conditional sampling distribution for the outcome variable:

$$f(\mathbf{Y}|\mathbf{X}, \boldsymbol{\beta}_0, \mathbf{S}, \mathbf{A}, \boldsymbol{\eta}, \mathcal{C}) = \prod_{\ell=1}^{m} f(\mathbf{Y}_\ell|\mathbf{X}_\ell, \boldsymbol{\beta}_\ell, \mathbf{S}, \mathbf{A}_\ell, \boldsymbol{\eta}, \mathcal{C}),$$

  where the product is taken over the m substantive clusters.

- One remaining question: how do you put a prior on the number of clusters?

Example of Cluster Configurations

Partitions for $n = 4$

$p = 1$	$p = 2$		$p = 3$	$p = 4$
$\mathfrak{R}_1$	$\mathfrak{R}_2$		$\mathfrak{R}_3$	$\mathfrak{R}_4$
$y_1y_2y_3y_4$	$y_1 \mid y_2y_3y_4$	$y_1y_2 \mid y_3y_4$	$y_1 \mid y_2 \mid y_3y_4$	$y_1 \mid y_2 \mid y_3 \mid y_4$
	$y_2 \mid y_1y_3y_4$	$y_1y_3 \mid y_2y_4$	$y_1 \mid y_3 \mid y_2y_4$	
	$y_3 \mid y_1y_2y_4$	$y_1y_4 \mid y_2y_3$	$y_1 \mid y_4 \mid y_2y_3$	
	$y_4 \mid y_1y_2y_3$		$y_2 \mid y_3 \mid y_1y_4$	
			$y_2 \mid y_4 \mid y_1y_3$	
			$y_3 \mid y_4 \mid y_1y_2$	

- Four Cluster Classes, Five Configuration Classes
- The number of configuration classes in each cluster class is $b(n, p)$
  - $b(n, p)$ = partitions of the integer n into p components ≥ 1
  - $b(4, 1) = 1, \quad b(4, 2) = 2, \quad b(4, 3) = 1, \quad b(4, 4) = 1,$

Notes On Cluster Configurations

- So for this $n = 4$ illustration in each of the cluster classes, $p = (1, 2, 3, 4)$, there are: $b(n, p) = (1, 2, 1, 1)$ configuration classes, and $(1, 7, 6, 1)$ partition types.
- The number of partition types, for a given n and p is a Stirling number of the second kind from:

$$\left\{ \begin{array}{c} n \\ p \end{array} \right\} = \frac{1}{p!} \sum_{j=0}^{p} (-1)^{p-j} \binom{p}{j} j^n.$$

- In the example there are 15 total possible partitions (models), the Bell number for $n = 4$ from:

$$B_n = \frac{1}{e} \sum_{j=0}^{\infty} \frac{j^n}{j!}$$

- We connect these because a Bell number can be expressed as the sum of Stirling numbers of the second kind:

$$B_n = \sum_{p=0}^{n} \left\{ \begin{array}{c} n \\ p \end{array} \right\}$$

- For a fixed m, the number of configuration classes $b(n, m)$ grows as $\frac{n^{m-1}}{m!(m-1)!}$ with increasing n.

Cluster Prior Probabilities

- Our prior for $\mathcal{C}$ is the *hierarchical uniform prior* (Casella, Moreno, and Girón, 2014):
  - ▷ First, a prior distribution is placed on m (# of cluster classes), which is taken to be a Poisson distribution truncated to $1, \ldots, n$, with the intensity parameter, ζ, whose default value is 1.
  - ▷ Given this generated m, there are $n_1, \ldots, n_m > 0$ partition types.
  - ▷ Within a partition type, the particular partitions of that type are given a uniform prior.
- For the partition type given by $n_1, \ldots, n_m$ there are $\mathcal{NP}(n_1, \ldots, n_m)$ unique partitions where:

$$\mathcal{NP}(n_1, \ldots, n_m) = \binom{n}{n_1, \ldots, n_m} \frac{1}{R(n_1, \ldots, n_m)}$$

and

$$R(n_1, \ldots, n_m) = \prod_{i=1}^{n} \left[\sum_{\ell=1}^{m} I(n_\ell = i) \right] !$$

accounts for over-counting of partitions in the multinomial coefficient from reordered identical partitions: $[y_1|y_2|y_3, y_4]$ is the same as $[y_2|y_1|y_4, y_3]$.

Marginal Cluster Probabilities

- At each step of the sampler, a unique model $\mathcal{C}$ is generated by a partition from the sampler, defining a new likelihood function for the data since $\mathcal{C}$ is conditioned on cluster model-specific parameters,

$$\theta_{\mathcal{C}} = (\boldsymbol{\beta}_{\mathcal{C}}, \boldsymbol{\sigma}^2_{\mathcal{C}}),$$

  as well as the other parameters whose prior forms do *not* depend on the partition status,

$$\boldsymbol{\Upsilon} = (\boldsymbol{\beta}_0, \sigma^2_{\boldsymbol{\beta}}, \mathbf{S}, \boldsymbol{v}, \tau, \boldsymbol{\eta}, A, \lambda).$$

  (here A is a collector for the set of A_ℓ, $\boldsymbol{\eta}$ is a collector for the set of $\boldsymbol{\eta}_\ell$, and the vector $\boldsymbol{v}$ is the diagonal of $\mathbf{S}$).

- So $\theta_{\mathcal{C}}$ typically differs in structure on each iteration since $\mathcal{C}$ differs on each iteration, yet $\boldsymbol{\Upsilon}$ retains its original dimension since it depends only on the class of models specified independent of the product partitioning.

- Dimension changing is based on Booth, Casella and Hobert (2008), *Appendix* A and B.

Overview of Stochastic Search

- Sampler steps from chosen starting points in the sample space:
  1. Gibbs sample $\boldsymbol{\Upsilon}$ (parameters whose prior forms do *not* depend on the partition status) conventionally.
  2. Then generate a candidate: $\{\mathcal{C}, (\boldsymbol{\beta}_{\mathcal{C}}, \boldsymbol{\sigma}^2_{\mathcal{C}})\}$ which is conditional on the current $\boldsymbol{\Upsilon}$.
  3. Accept or reject the latter as a complete block with a Metropolis step.
  4. Repeat.

- Last major remaining challenge: efficient mixing through cluster-space is difficult.

Sampling the $\mathbf{A}$ Matrix Transition Kernel From Υ

- Let $\mathbf{A}^{(t)}$ be the current bundle configuration.
- A proposal $\mathbf{A}'$ is generated according to the "restaurant" algorithm with the current λ.
- Then $\mathbf{A}'$ or $\mathbf{A}^{(t)}$ is selected with a Metropolis decision.

$$\mathbf{A}^{(t+1)} = \begin{cases} \mathbf{A}' & \text{with probability} \quad P\left[\min\left(a(\mathbf{A}', \mathbf{A}^{(t)}), 1\right)\right] \\ \mathbf{A}^{(t)} & \text{with probability} \quad 1 - P\left(\min\left(a(\mathbf{A}', \mathbf{A}^{(t)}), 1\right)\right] \end{cases}$$

  where $a(\mathbf{A}', \mathbf{A}^{(t)})$ is the acceptance ratio.

- The larger λ is, the more diffuse the sampling from the Multinomial-Dirichlet and thus for large λ many smaller bundles will be sampled.
- For small λ (say $\lambda = 1/n$), the sampling produces bundle sizes in $\mathbf{A}'$ which are more like those from $\mathbf{A}^{(t)}$.
- Sampling the Dirichlet process precision term, λ, is aided by a parameter expansion process simplified from that of Escobar & West (1995).

Two-Level Cluster Transition Kernel for $\theta_{\mathcal{C}}$

- Remember that our sampler has to explore *cluster space*, which is not a very typical application.
- Since clustering searches need big steps to mix through the sample space as well as small steps to refine high probability clustering outcomes, we mix big moves and with small moves in a micro-step:.
  - ▷ Jain-Neal (2000) split-merge sampling scheme (large moves),
  - ▷ random walk Metropolis (small moves),

  where the choice ratio is a tuning parameter.
- The macro-step then uses this candidate position, $(\mathcal{C}', \boldsymbol{\beta}'_{\mathcal{C}}, \boldsymbol{\sigma}^{2\prime}_{\mathcal{C}})$ for a Metropolis-Hastings accept/reject step comparing $\pi(\mathcal{C})$ and $\pi(\mathcal{C}')$ with the corresponding posterior (up to a proportion):

$$\pi(\mathcal{C}|\boldsymbol{\Upsilon}', \boldsymbol{\eta}, \mathbf{A}, \lambda, \mathbf{y}) \propto \left(\prod_{\ell=1}^{m} f_\ell \left(\mathbf{y}_\ell | \boldsymbol{\Upsilon}', \boldsymbol{\eta}, \mathbf{A}, \lambda \right) \right) P(\mathcal{C}),$$

- Finally once the sampler selects a *cluster configuration*, the set of cluster-specific parameters $(\boldsymbol{\beta}_{\mathcal{C}}, \boldsymbol{\sigma}^2_{\mathcal{C}})$ are directly drawn from their individual full conditional distributions.

Calculating Cluster Posterior Probabilities After the Sampler

- In order to produce an estimate of $\pi(\mathcal{C}|\mathbf{y}, \mathbf{X})$, we utilize the described draws from the random walk through the model space.
- For each unique partition $\mathcal{C}^{(i)}$ and each draw of the hyperparameters and Dirichlet process random effects $\boldsymbol{\Upsilon}^{(i)}$, we compute and estimate the conditional posterior probability of the partition, given by

$$\hat{P}(\mathcal{C}^{(i)}|\boldsymbol{\Upsilon}^{(i)}, \mathbf{y}, \mathbf{X}) = \frac{f(\mathbf{y}|\mathcal{C}^{(i)}, \boldsymbol{\Upsilon}^{(i)}, \mathbf{X})p(\mathcal{C}^{(i)})}{\sum_{\mathcal{C}'} f(\mathbf{y}|\mathcal{C}', \boldsymbol{\Upsilon}^{(i)}, \mathbf{X})p(\mathcal{C}')}$$

  where the sum is taken over all partitions $\mathcal{C}'$ visited by the chain, post-convergence.

- The fact that the sum in the denominator is not over all possible partitions is what makes the calculated quantity only an estimate.
- The true value of $\pi(\mathcal{C}|\mathbf{y}, \mathbf{X})$ could be directly computed if the sum could be taken over the set of all possible partitions, but this is impossible due to the size of the model space.

Motivation

- The safety of millions of people depends on the understanding of the workings of terrorist networks and their behavior.
- To protect people, governments and nongovernmental organizations invest enormous amounts of time and energy to detect covert networks and to thwart terrorist events and other kinds of attacks.
- Terrorism is an important *political* and *public health* problem because it affects:
  - government stability,
  - personal safety,
  - immediate epidemiological concerns,
  - internal government policies,
  - public perception and panic,
  - and possibly widespread health effects.
- Complex empirical modeling work on terrorism has increased dramatically in recent decades for obvious reasons, but still remains somewhat under-developed.

Application: Terrorism Data Analysis

- Terrorism is an important problem because it affects internal government policy, public perception, relations between states, and of course, personal safety.
- To protect citizens, governments and nongovernmental organizations invest enormous amounts of time and energy to understand and to thwart terrorist attacks.
- This remains a challenging social, political, and military problem because many of the variables that we would like to see are unobservable under even the most highly visible situations.
- Furthermore, the study of terrorism belongs to *every* subfield in political science.

Application: Terrorism Data Analysis

- The study of terrorism has not made enough *empirical* progress due to inherent problems in the available data:
  - selection on the outcome (a visible event),
  - non-granular discrete measurement,
  - insufficient explanatory variables,
  - non-ignorable missingness,
  - the lack of access to classified collections,
  - often filtered through government sources,
  - studied actors are deceptive,
  - unmeasured clusters in almost all terrorism data.

The Data

- *Big Allied and Dangerous* (BAAD) Database 1 (Asal, Rethemeyer & Anderson 2008).
- Assembled from several established databases: Memorial Institute for the Prevention of Terrorism's (MIPT) Terrorism Knowledge Base (TKB), Correlates of War (COW), Polity, and Polity2.
- This aggregates 395 worldwide lethal attacks from 1998-2005 by terrorist organizations.
- We use the version of their dataset that excludes Al Qaeda since its scope, profile, and effectiveness place it in a unique category during this period.
- The variable `fatalities` (total number) is used as the outcome variable to focus on the primary purpose of these attacks.

The Explanatory Variables Used

- `statespond` indicates whether the group is financially or logistically supported by one or more recognized governments (coded 1, $n_1 = 32$), or not (coded 0, $n_0 = 363$).
- `masterccode` denotes the COW `CCODE` value: where (country/region) attack took place (so roughly distance from the US).
- `ordsize` is size according to 0 for less than 100 members ($n_0 = 261$), 1 for 101-1,000 members ($n_1 = 77$)), 2 for 1,001-10,000 members ($n_2 = 45$), and 3 for more than 10,000 members ($n = 12$).
- `terrStrong` is coded 1 ($n_1 = 43$) if they possess territory and 0 if they do not ($n_0 = 352$).
- `degree` gives a count of alliance connections in the network sense.

The Explanatory Variables Used

- `LeftNoReligEthno`, where a 1 indicates that the group's ideology is leftist and it is not compounded with another ideological orientation ($n_1 = 94$), and a 0 indicates that group's ideology is either not leftist or is a mix of leftist and at other ideological dimensions ($n_0 = 301$).
- `PureRelig` indicates with a 1 whether the group's ideology is purely religious and not associated with other political or social factors ($n_1 = 50$), and 0 otherwise ($n_0 = 345$).
- `PureEthno` indicates with a 1 whether the group is ethnonationalist (nationalist causes tied to ethnic identity) and not associated with other ideological factors ($n_1 = 26$), and 0 otherwise ($n_0 = 369$).
- `Islam` where a 1 is assigned to groups inspired by some form of Islam ($n_1 = 287$) and 0 otherwise ($n_0 = 108$).

Model Results

- We estimate the DPP/Product Partition model using the sampler described.
- The Markov chain is run for 10,000 iterations disposing of the first 5,000 as burn-in.
- Convergence is assessed with `superdiag`, a diagnostic suite provided by an `R` package (Tsai and Gill 2012) that calls all of the conventional convergence diagnostics typically used (Gelman & Rubin, Geweke, Heidelberger & Welch, Raftery & Louis).
- We also found no evidence of non-convergence with standard graphical tools (traceplots, cumsum diagrams, etc.).
- The highest posterior probability cluster arrangement

1	2	3	4
272	7	52	64

  or something very close is visited 0.45191 of the time.
- Now we run a regular Bayesian linear model (diffuse proper priors) with a single shared random effects and a true multilevel linear model (diffuse proper priors) with the estimated clusters as group definitions.

	Standard Linear Model				Multilevel Linear Model		
	Mean	Std.Err.	95% HPD		Mean	Std.Err.	95% HPD
α	-0.290	1.287	[-2.811:2.232]	α_1	-3.835	0.843	[-5.486:-2.184]
				α_2	0.383	1.480	[-2.517: 3.283]
				α_3	-1.905	1.040	[-3.942: 0.133]
				α_4	19.235	1.139	[17.002:21.468]
`statespond`	0.514	1.193	[-1.824:2.851]		3.590	0.840	[1.945: 5.235]
`masterccode`	0.006	0.032	[-0.057:0.069]		-0.054	0.019	[-0.092:-0.016]
`ordsize`	4.749	0.719	[3.339:6.159]		3.163	0.452	[2.277: 4.049]
`terrStrong`	3.849	1.355	[1.193:6.504]		1.886	0.974	[-0.022: 3.795]
`degree`	2.307	0.298	[1.723:2.890]		1.169	0.179	[0.818: 1.520]
`LeftNoreligEthno`	0.290	1.070	[-1.808:2.388]		0.838	0.707	[-0.548: 2.224]
`PureRelig`	1.131	1.307	[-1.431:3.694]		1.669	0.955	[-0.202: 3.540]
`PureEthno`	-0.948	1.410	[-3.713:1.816]		-1.378	1.045	[-3.427: 0.670]
`Islam`	2.851	1.203	[0.492:5.210]		3.179	0.857	[1.499: 4.858]
τ	0.009	0.001	[0.007:0.020]		0.027	0.002	[0.023: 0.031]
	Summed Deviance 3002				Summed Deviance 2553		
					Variance	Std.Dev.	
				σ_α	113.44	10.65	
				σ_y	1.31	1.15	

Notes

- Restricting the cluster sample space by putting a *substantive* upper bound on the number of clusters is extremely helpful.
- Testing the model with data having a *known* number of clusters shows excellent properties.
- Currently developing a cluster-space convergence diagnostic (this is really hard).
- The regular Dirichlet process prior model described is in package `glmdm` (Jeff Gill, George Casella, Minjung Kyung, and Jonathan Rapkin, authors).
- Currently thinking about other ways to do post-processing.

THANK YOU!

Theoretical Papers: Dirichlet Process Priors on Random Effects Terms in GLMMs

- Minjung Kyung, Jeff Gill and George Casella. "Sampling Schemes for Generalized Linear Dirichlet Process Random Effects Models." ***Statistical Methods and Applications***, 20:3, 259-290 (2012). For DPP slice sampling worse than KS mixture representation or MH algorithm.
- Minjung Kyung, Jeff Gill and George Casella. "New Findings from Terrorism Data: Dirichlet Process Random Effects Models for Latent Groups." ***Journal of the Royal Statistical Society, Series C***, 60:5, 701-721, (2011). DPP on RE remove more of the underlying variability from the data, uncovering latent information with difficult data.
- Minjung Kyung, Jeff Gill and George Casella. "Estimation in Dirichlet Random Effects Models." ***Annals of Statistics***, 38, 979-1009 (2010). Putting the precision parameter directly into the sampler avoids typical problems with ML estimation.
- Minjung Kyung, Jeff Gill and George Casella. "Characterizing the Variance Improvement in Linear Dirichlet Random Effects Models." ***Statistics and Probability Letters***, 79, 2343-2350, (2009). DPP on RE produce lower SE for regression parameters on average.
- Jeff Gill and George Casella. "Nonparametric Priors For Ordinal Bayesian Social Science Models: Specification and Estimation." ***Journal of the American Statistical Association***, 104, 453-464 (June 2009). DPP on RE can uncover latent heterogeneity information.