Probabilistic Models of time series data

# Dirichlet Processes

- Krishna Devkota

# Motivation

- Your friends are always complaining about food in the mensa
- So, you decide to start a restaurant nearby

- But, there is a problem ...

  - you have no clue about the food preferences of the students *(some can’t do without beer and pizza, some prefer vegan tofu, some want sushi maybe)*

- So, you start with a survey of people around you
- **Goal:**

  - Find the food preferences of the students and cluster them into appropriate groups

# *Clustering problem:*

- Given observations: $x_1, x_2, \ldots. x_n$

***Objective:***

- subdivide them into subsets, *i.e.* ***clusters***
- some sort of ***similarity*** sought for observations within each ***cluster***

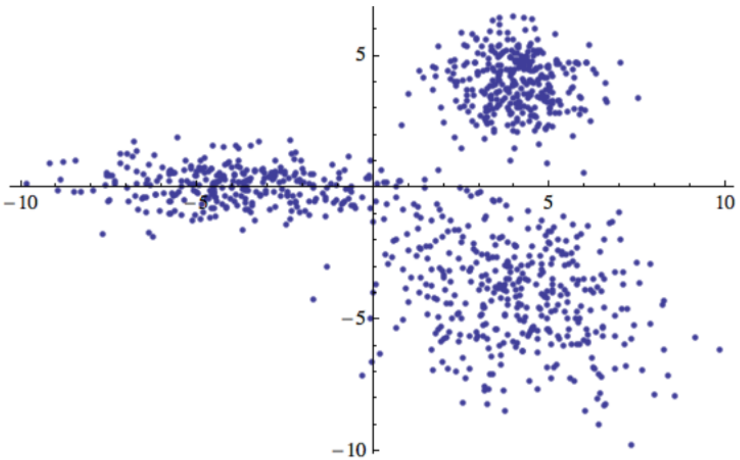
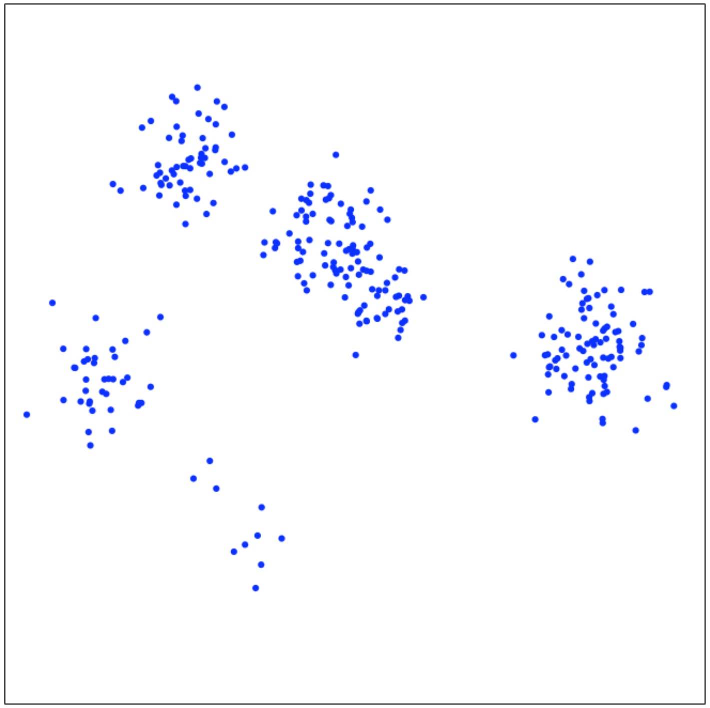



*Fig: 1000 points divided into 3 clusters*

# *Clustering in terms of mixture models:*



First, some basics!

***Modeling assumption:***

- Data is partitioned into groups / clusters

- Each observation $\boldsymbol{x_i}$ belongs to a single cluster $\boldsymbol{k}$

Express ***cluster assignment*** as ***random variable*** $\boldsymbol{L_i}$

- so, $L_i = \boldsymbol{k}$ means $X_i$ belongs to cluster $\boldsymbol{k}$

- As cluster assignments unknown, $L_i$ remains *latent* (*unobserved*)

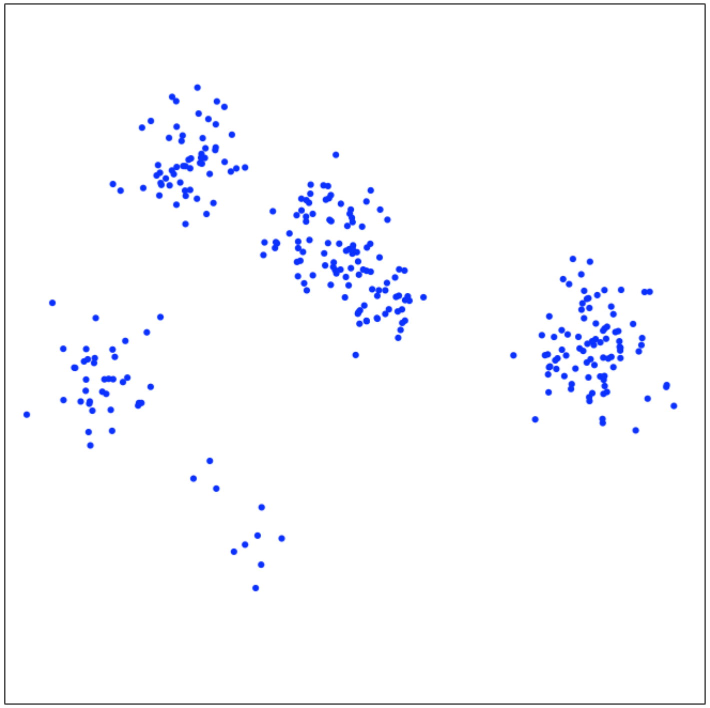
Stating it differently,

what is *unknown* is the ***partition*** of data-space *{1,…..,N}*

*N = number of observed data-points*

- Distribution characterizing a ***single cluster*** *k*, given as:

$$P_k(\bullet) := \mathbb{P}[X \in \bullet | L = k]$$



- Probability that newly generated observation belongs to cluster *k* (*basically the weights*):

$$c_k := \mathbb{P}\{L = k\}$$

As $\boldsymbol{c_k}$ are probabilities of *mutually exclusive events*, they sum up to *1*.

So, now the distribution of ***X*** is of the form:

$$P(\bullet) = \sum_{k \in \mathbb{N}} c_k P_k(\bullet)$$

- A *model* of this form is called a ***mixture distribution.***

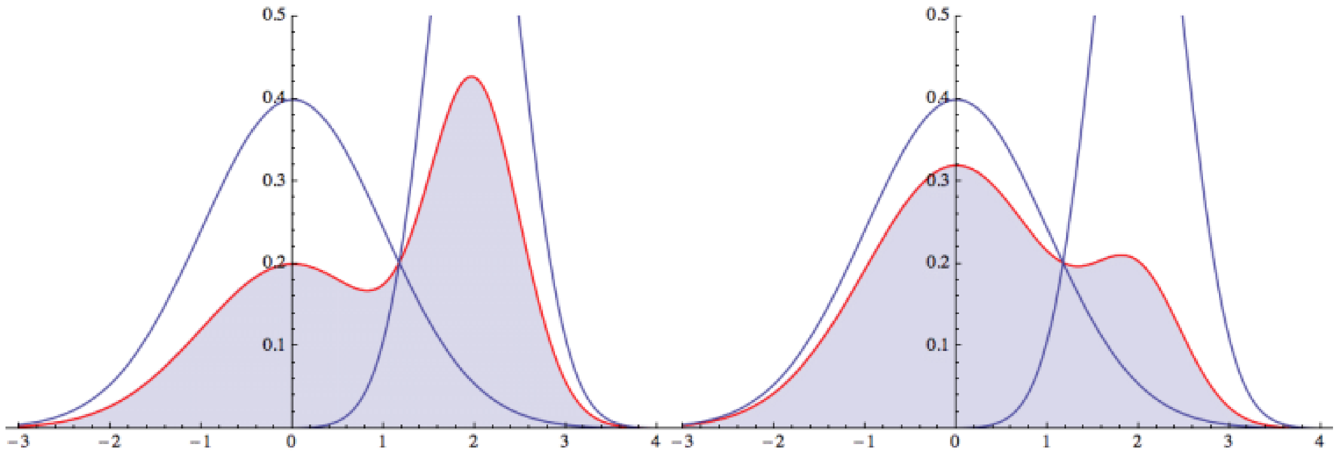



*Figure: mixture of two Gaussian, left: c1 = 0.5, c2 = 0.5, right: c1 = 0.8, c2 = 0.2*

- If the ***number*** of ***clusters*** is finite, meaning if there are finite number of ***K*** with non-zero probabilities $\boldsymbol{c_k}$, we call the mixture a ***finite mixture***

# ***Mixture models:***

$$p(x|m) = \int_{\Omega_\theta} p(x|\theta) m(d\theta)$$

*Where, **p (x|θ)** is the component distribution*

***m(dθ)** is called the mixing measure*

***To see what this means,***

*Think of **p(x| θ)** as a Gaussian distribution with parameter **θ,***

- The second part ***m(dθ)*** is a distribution over θ

- ***m(dθ)*** is a distribution of finitely many *delta spikes*.

*(**p(x|θ)** defines what kind of mixture model it is, i.e. if p (x|θ) is a Gaussian then a Gaussian mixture model)*

## *Mixture models can be looked upon as a two stage sampling:*

For a given mixture model:

$$p(x|m) = \int_{\Omega_\theta} p(x|\theta)m(d\theta)$$

- first sample ***θ*** from the *mixing measure* ***'m'***

- plug-in the theta as a parameter of the distribution ***p(x|θ)***

- then sample the random variable ***x*** from the resulting distribution ***p(x|θ)***

Sample $X \sim p(\,.\,|m)$ as:

1. $\Theta \sim m$
2. $X \sim p(\,.\,|\theta)$

### ***Parameter space for a mixture model:***

- Set of all *discrete* probabilities on the *parameter space* $\Omega_\Phi$ defined by *p(x|φ)*

- Φ = location of *atoms*

# *Finite mixture model*

$$p(x|\boldsymbol{\theta}, \mathbf{c}) = \int_{\Omega_\theta} p(x|\theta) m(d\theta) \qquad \text{with} \qquad m(\,.\,) = \sum_{k=1}^{K} c_k \delta_{\theta_k}(\,.\,)$$

- where, mixing measure ***m(.)*** is simply sum over finite number of delta spikes at ***k*** different locations

- $\boldsymbol{c_k}$'s are the *convex coefficients* to weigh the ***Dirac deltas***
*(convex coefficients simply mean that c values are non-negative and sum to 1)*

***Note:*** *The mixture model is parameterized by the mixing distribution/ mixing measure!*

## *Bayesian mixture model*

A ***Bayesian mixture model*** is simply a mixture model with a random mixing measure.

Randomizing our *parameters* **c** and $\boldsymbol{\theta}$ will give us a random *mixing measure*:

$$M(\,.\,) = \sum_{k=1}^{K} C_k \delta_{\Theta_k}(\,.\,)$$

*We basically want to put a prior for the mixing distribution!*

***How to choose the priors?***

*Idea from the very start:* Use a ***conjugate prior***!

$$M(\,.\,) = \sum_{k=1}^{K} C_k \delta_{\Theta_k}(\,.\,)$$

What is a ***conjugate prior***?

- ***Posterior*** belongs to the same class of distribution as the *prior*!

***- Conjugate priors*** generally occur only in the ***exponential families*** in the ***parametric*** case!

*This accounts for the **θ's** but we still have the **C**'s!*

$$M(\,.\,) = \sum_{k=1}^{K} C_k \delta_{\Theta_k}(\,.\,)$$

**What could be the distribution of *c*'s?**

Observation:

- when sampling from the ***mixture-model*** the c's (*index of the clusters*) are found to be: ***multinomial distributed***

- The parameter of this distribution is the *vector* of *coefficients* [c1,c2,….ck]

***The conjugate prior of a multinomial is a Dirichlet!***

So, we'll use a Dirichlet distribution on our weight vector [c1,c2,….ck]

## *Let's quickly look back at our restaurant problem:*

We could use one of the standard algorithms based on *FMM:*
- k-means clustering *(not a probabilistic model)*
- Gaussian mixture models

- But remember, we surveyed very limited group of students!

- In reality, there are infinite (well, very large) number of foodies out there!

- How do you account for *new food choices* of other students (***hidden clusters***) which may arise later on?

To use a *Finite mixture model,* we (need to) start by assuming a certain (finite) number of clusters

But, what we really want is to allow newer clusters as our data (*students*) grow!

***So, we have to do it in a non-parametric way!***

*Very quick look at **parametric** vs. **non-parametric** model*

***Parametric:***

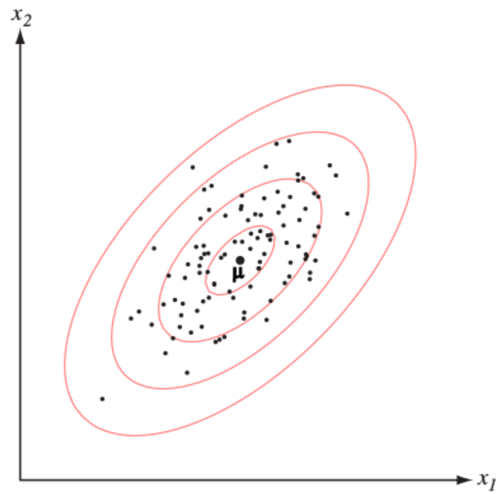
Number of parameters fixed w.r.t sample size

***Non-parametric:***

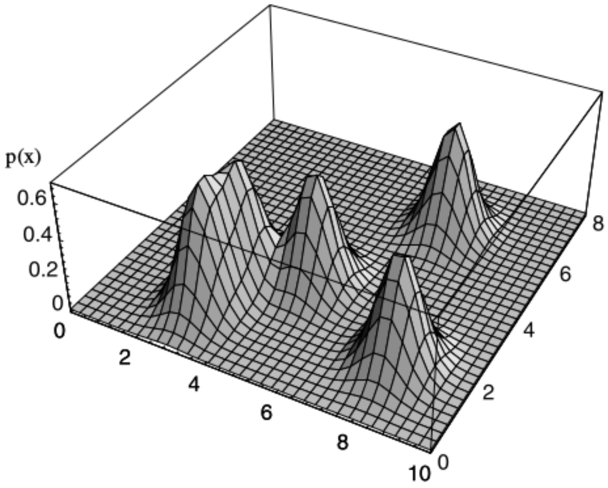
- number of parameters grow with sample size
- infinite dimensional parameter space

***(parameter space: space of models that can explain our data)***

***Quick example of density estimation:***




Parametric




Nonparametric

*Note: Fig. not necessarily a Bayesian!*

# *Dirichlet process mixture: Infinite limit of finite mixture models:*

In the *Bayesian mixture model*, the *mixing measure* was:

$$M(\,.\,) = \sum_{k=1}^{K} C_k \delta_{\Theta_k}(\,.\,)$$

If K → ∞, in our previous model, we get a ***non parametric*** model :

$$M(\,.\,) = \sum_{k=1}^{\infty} C_k \delta_{\Theta_k}(\,.\,) \qquad \text{where} \qquad \sum_{k=1}^{\infty} C_k = 1$$

When nonparametric, we need to generate M(.) at random. Think about how we could do that?

Also, note the restriction: *infinite sum* over **c**'s must be 1.

Simple distribution that does this: Dirichlet process!

## What do we mean by simple?

**- as much *independence* as possible**

(if c's and *θ's highly dependent, very difficult to do inference on that model)*

## *Dirichlet Process:*

A Dirichlet process is a distribution on random probability measures of the form:

$$M(\,.\,) = \sum_{k=1}^{\infty} C_k \delta_{\Theta_k}(\,.\,) \qquad \text{where} \qquad \sum_{k=1}^{\infty} C_k = 1$$

Take some base distribution $\mathbf{G_o}$ and sample *atoms* ***θ's*** iid from it *(infinite seq of thetas)*

### What about the weights?

- they can't be independent, because should sum up to one!

*(so if 1st weight 0.5, second can't be greater than 0.5)*

**Next best thing we can do?**

- sample *independent proportions* and use them to generate weights.

This brings us to: ***stick breaking construction!***

# Constructive definition of $DP(\alpha,G_0)$ [stick-breaking analogy]

***STEPS:***

- Start with a stick of length one

- Generate a random variable: *β1~Beta(1,α)*

- By the definition of the Beta distribution, this will be a real number between 0 and 1, with expected value *1/(1+α)*

- Break off the stick at ***b1***

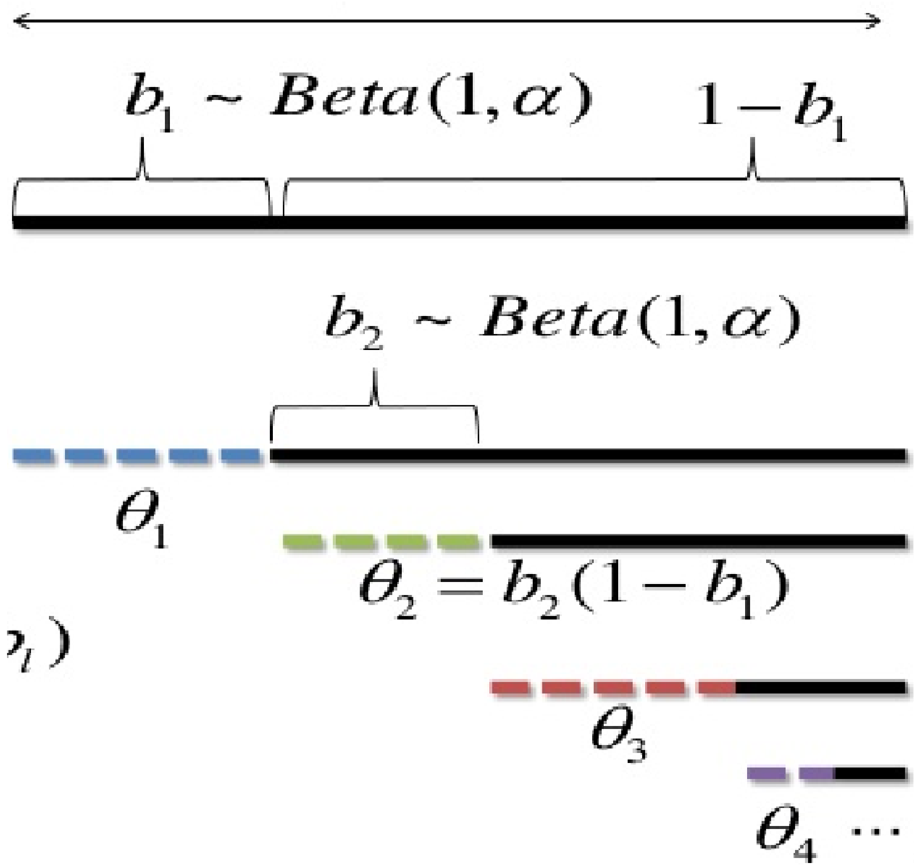
- *w1 = 1-b1* is then the length of the stick on the left

- Now, generate *β2~Beta(1,α)*

- Break off the part of stick *b2*

- Again, ***w2*** is the length of the stick to the left, *i.e.*, *w2=(1−b1)b2*

- Repeat the process.

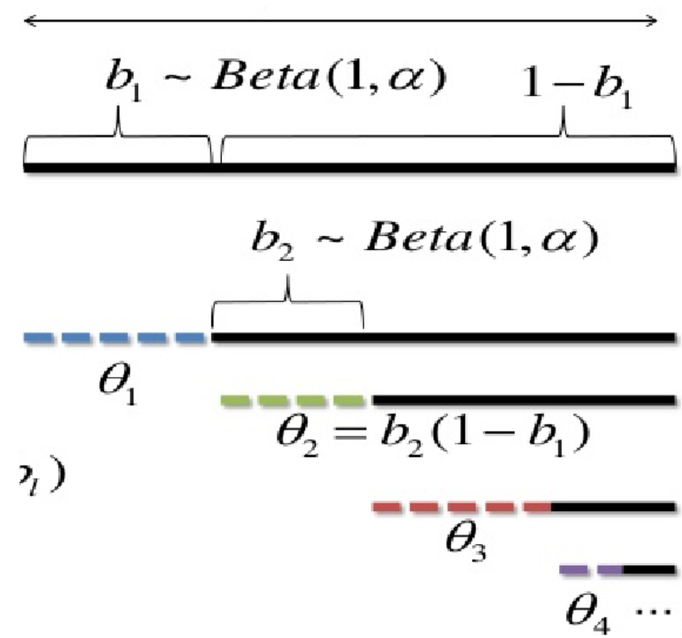


Mathematically,
the process can be described as:

$$\Theta_k \sim_{\text{iid}} G_0$$
$$V_k \sim_{\text{iid}} \text{Beta}(1, \alpha)$$

Compute $C_k$ as

$$C_k := V_k \prod_{i=1}^{k-1} (1 - V_i)$$

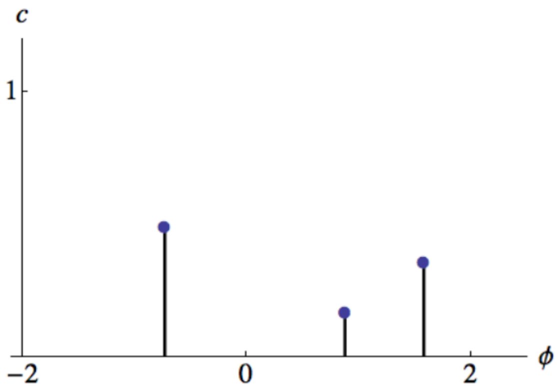


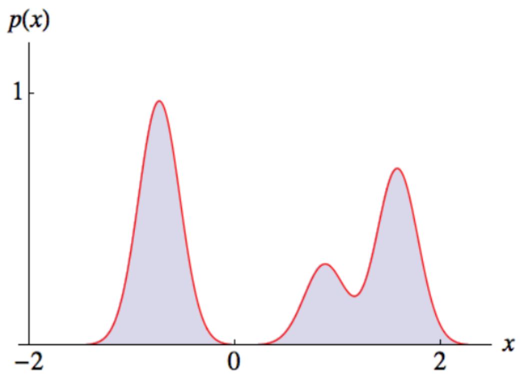
## *Dirichlet process in our clustering problem:*

$$M(\,.\,) = \sum_{k=1}^{\infty} C_k \delta_{\Theta_k}(\,.\,) \qquad \text{where} \qquad \sum_{k=1}^{\infty} C_k = 1$$

*Intuitive* understanding of the above expression:

Let's say the centers of our clusters come from a base distribution (Go) that is a Gaussian:




- each line here is a weighted atom
- each draw here will be our cluster
- these are the center of our clusters

- For each atoms in the left figure, new Gaussian is created around the point that we drew

***If our data comes from cluster 1, lands in first area with high probability, and so on!***

So far we,

- described a *generative model* that allows us to calculate probability of assigning any particular *set of groups* to our *data points*

But, how do we actually learn a good set of group assignments!

**In other words, how do we infer the model!**

# *Inference in the DP mixture (Gibbs sampling):*

***Approach:*** *MCMC* sampling (particularly Gibbs)

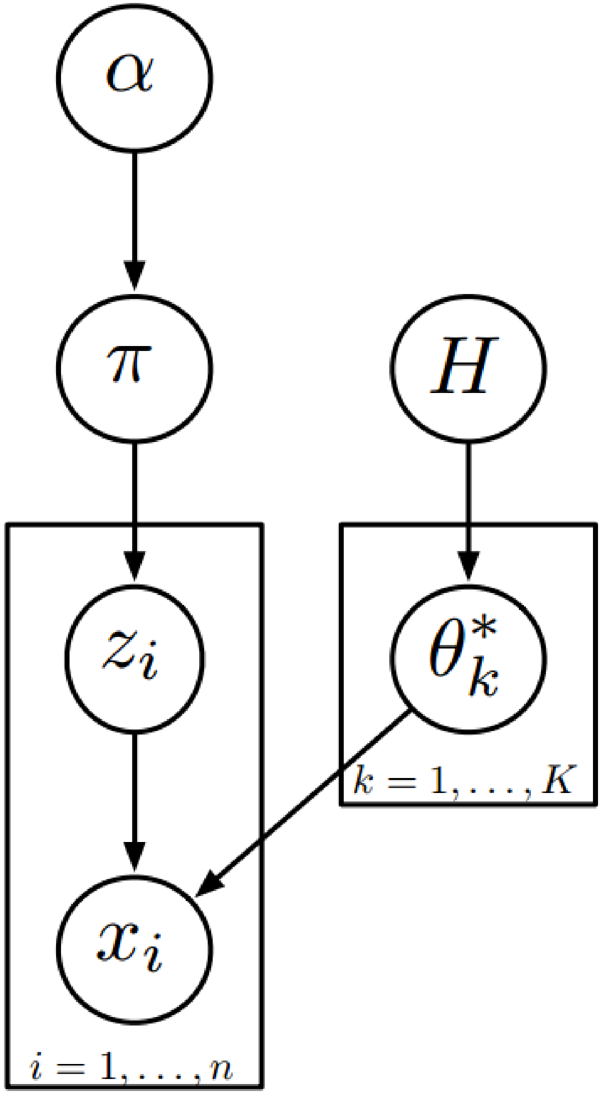
***Three unknowns:***

*$\pi$ (parameter), $z_i$ (latent variable), $\theta_k^*$ (parameter)*

***Observation:***

*$x_i$ (data)*

***Objective:*** learn about $\pi$, $z_i$, $\theta_k^*$ in an *iterative* fashion



## *Inference (Gibbs sampling):*

**STEPS:**

***Step 1:***

- Start with some estimate of parameters $\theta_k^*$ and $\pi$
- Compute the "conditional distribution" of $z_i$ given the parameters and data
- Draw a sample from that conditional distribution
 *(hope: to achieve high probability values of $z_i$ that are more likely given the param and data)*

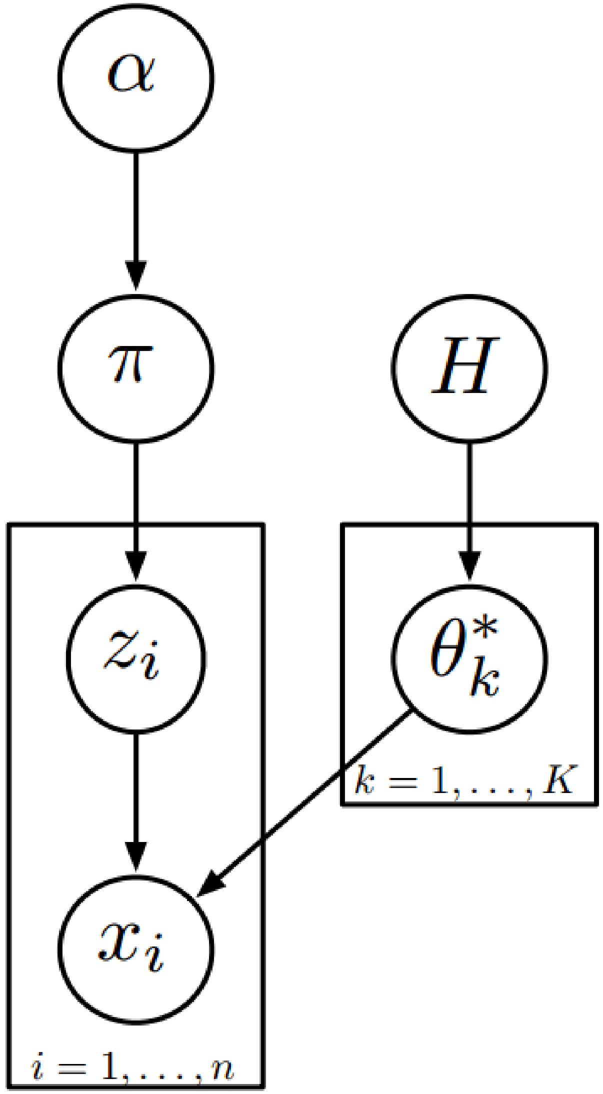
***Step 2:***

- After sampling the latent variables ($z_i$), conditioned on these latent variables and our data, compute the conditional distribution of the parameters

***Step 3:***

- Sample from the newly computed distribution of the parameters, and the update the value of latent variables.
- This completes one cycle of *MCMC* update.

Repeat the above steps till convergence!
***Will converge to true posterior distribution!***



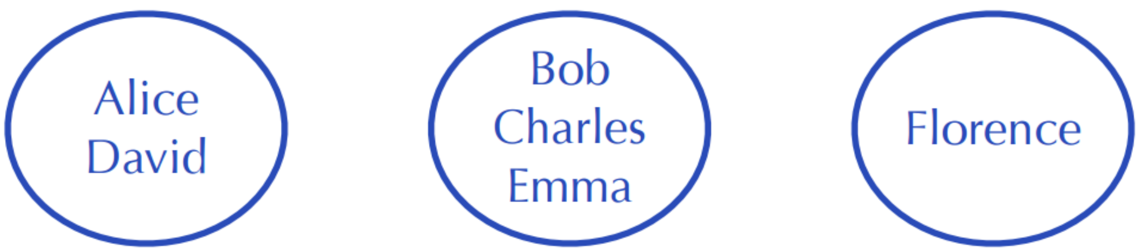
# *Chinese Restaurant Process:*

The distribution of ***random partition*** induced by the Dirichlet process can be clearly explained through the Chinese Restaurant Process.

## Quick Definition of Partition:

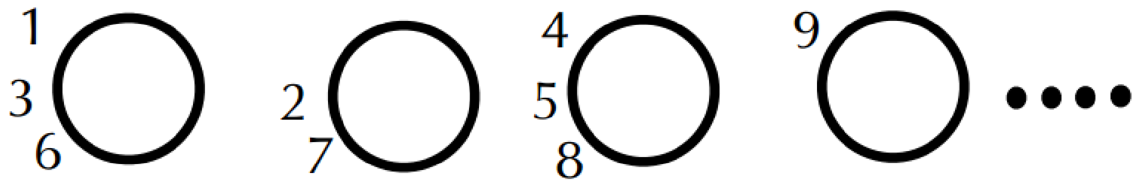
A partition of a set S is defined as a disjoint family of non-empty subsets of S whose union is S.

- $S = \{\text{Alice, Bob, Charles, David, Emma, Florence}\}$.
- $\varrho = \{\, \{\text{Alice, David}\}, \{\text{Bob, Charles, Emma}\}, \{\text{Florence}\}\, \}$.




- Denote the set of all partitions of $S$ as $\mathcal{P}_S$.
- **Random partitions** are random variables taking values in $\mathcal{P}_S$.

# *Chinese Restaurant Process*




Each customer comes into a restaurant and sits at a table

$$\mathbb{P}(\text{sit at table } c) = \frac{n_c}{\alpha + \sum_{c \in \varrho} n_c}$$

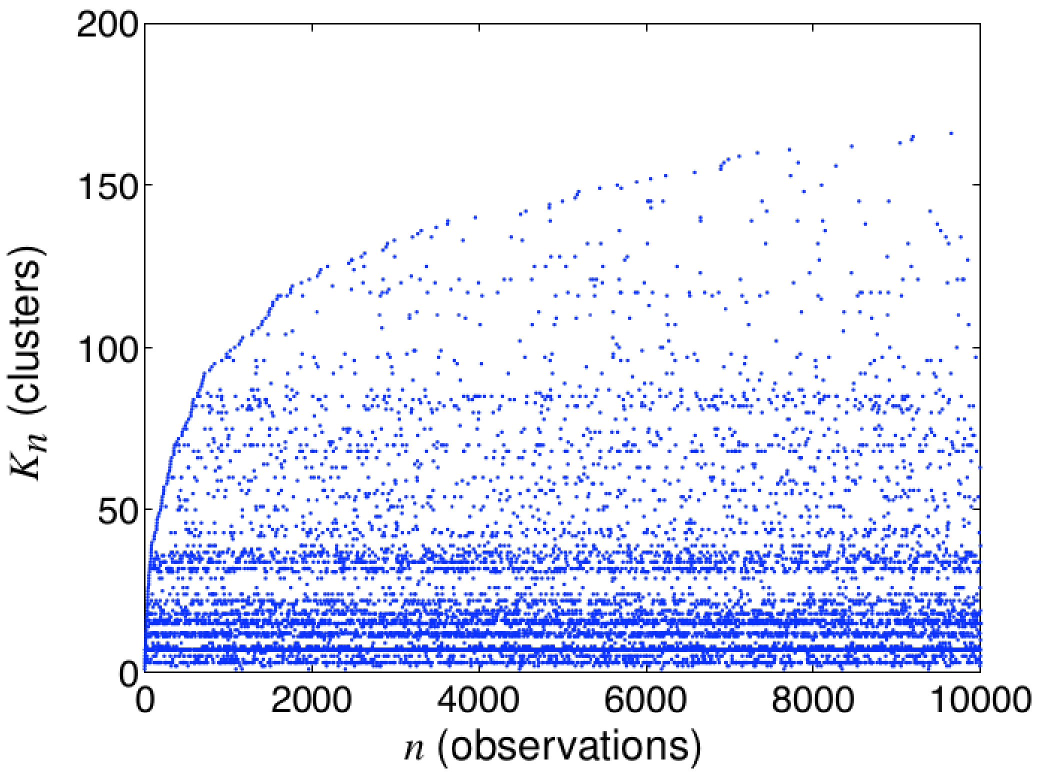
$$\mathbb{P}(\text{sit at new table}) = \frac{\alpha}{\alpha + \sum_{c \in \varrho} n_c}$$

Customers correspond to elements of : S

Tables correspond to clusters in : $\varrho$

***Rich get richer*** : large clusters more likely to attract new customers

# *Number of clusters of a DP:*




If Kn is number of clusters in sample of size n, then:

$$\mathbb{E}[K_n] = O(\log(n))$$

# *Chinese Restaurant Process*

## *Exchangeable Partition Probability function:*

Multiplying conditional probabilities together, the overall probability of $\varrho$

$$\mathbb{P}(\varrho|\alpha) = \frac{\alpha^{|\varrho|}\Gamma(\alpha)}{\Gamma(n+\alpha)} \prod_{c\in\varrho} \Gamma(|c|)$$

This is known as Exchangeable Partition Probability function.

We see that the ***order of the customers entering*** the restaurant ***does not affect*** the probability of $\varrho$

***As we see above, ex-changeability is one of the important properties of Dirichlet process!!***

# *What we have seen so far:*

### *Non-parametric Bayesian clustering:*

- Infinite number of clusters Kn <= n, where n = observations
- If partition exchangeable, it can be modeled using a random discrete distribution

### *Inference:*

- since partitions (cluster assignments) not directly observed, latent variable algorithm used
- Gibbs sampling

### *Assumptions:*

- prior assumption of number of clusters Kn
- distribution of cluster sizes

## *Dynamic Hierarchical Dirichlet Process (Extend DP to incorporate time dependence)*

But first,

### *Hierarchical Dirichlet Process:*

***Dirichlet Process*** where the base distribution $G_0$ is itself drawn from the *Dirichlet Process.*

This allows to *share a common parameter* across the ***groups*** through the base distribution, even if the different groups have their own distribution.

**To apply the Dirichlet process to a time-evolving sequence of data**

***Assumption:***

(i) Two data samples drawn at proximate times

- have a higher probability of sharing the same underlying model parameters (atoms) than parameters drawn at disparate times

(ii) Possibility that temporally distant data samples may also share model parameters,

- Possible distant repetition in the data

Statistical properties of data collected at consecutive time points are linked

BUT how?

***Steps:***

- Use a *random parameter* that controls their probabilistic similarity.

- Derive a sharing mechanism of time evolving data

- then, develop an appropriate Markov Chain Monte Carlo (MCMC) sampler

Possible applications:

- Music segmentation
- Gene expression data

# *Experimental Result:*

## *Music Segmentation:*

In music segmentation:

- interesting to infer relationship between different parts of a piece

- also similarity between different pieces

So,

- A music piece *divided* into different *contiguous sub-sequences*

- Each *subsequence* modeled via a ***HMM***

- ***dHDP*** useful here in enforcing the idea that *contiguous sub-sequences* are likely to *fall withing the same music segment*, and thus *share HMM parameters*

- when the *segment changes*, changes *detected* by *dHDP*

## ***STEPS:***

- First, ***MFCC features*** extracted

- ***discretized*** with ***vector quantization***

- Piece transformed into ***4980 discrete symbols***, and ***83 subsequences***

- Each ***subsequence*** is modeled using ***HMM*** with ***8 states***

- Each subsequence ***6 sec*** in length

***To model the time dependence between adjacent subsequences,***

- each ***subsequence*** corresponds to ***one group*** in the ***dHDP HMM*** mixture

- choose *one set of HMM parameters* according to the *corresponding mixture weights*

- In the *dHDP framework*, *one subsequence* can *share* the *old DP mixture distributions* with the *previous ones*

- Or it might be *drawn* from an *innovation DP mixture*,

- this may be also *shared* by the ***following time series*** in a similar manner

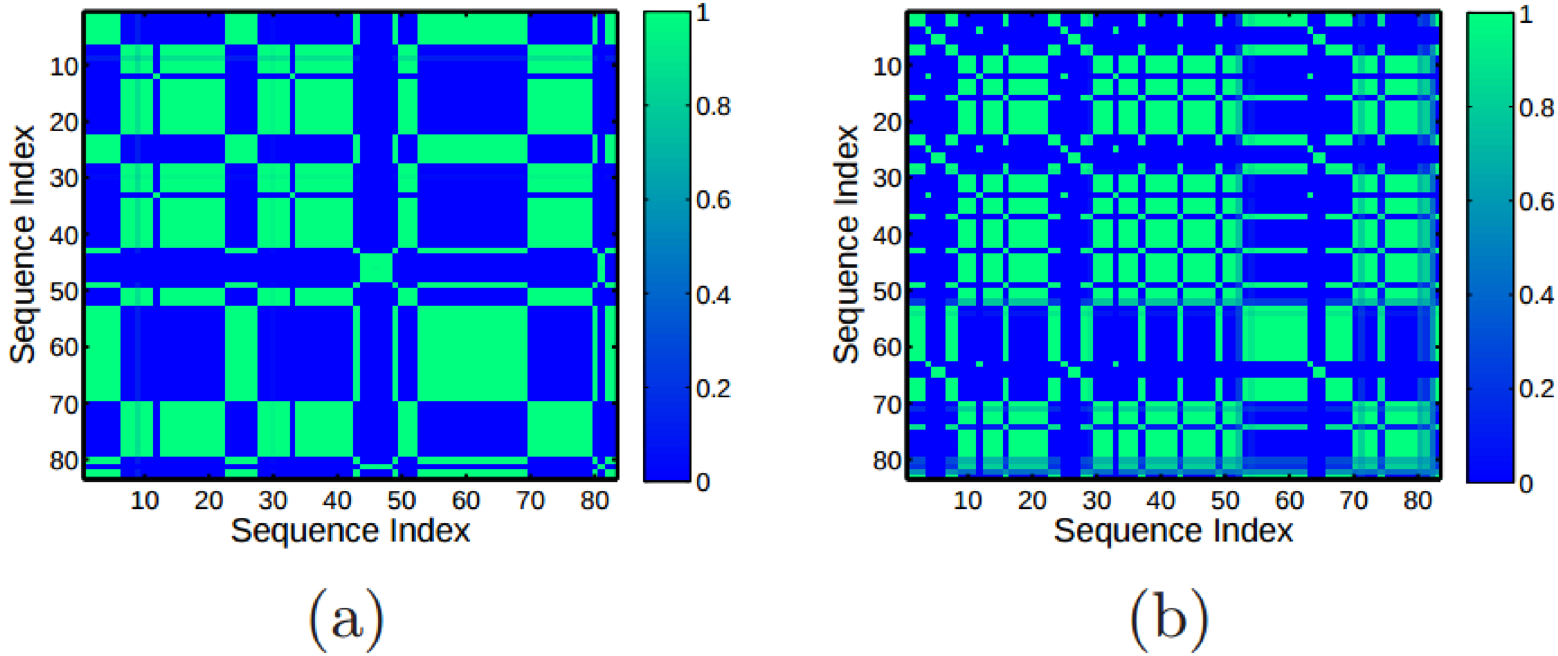



*Figure 3.* Similarity matrix $E(z'z)$ from HMM mixture modeling of the Sonata. (a) dHDP-HMM, (b) HDP-HMMs.

## *Conclusion:*

- Clustering problem

- Mixture models

    *Finite mixture model*
    *Infinite mixture model*
        -Dirichlet as a prior for IMM

- Dirichlet Process

- Construction of Dirichlet mixtures *(using stick breaking analogy)*

- Dirichlet mixture Inference
    *MCMC sampling (Gibbs sampling)*

- Random partitions introduced by Dirichlet process
    *(Explained through Chinese Restaurant process)*

- Properties of Dirichlet Process
  Exchangeability

- Hierarchical Dirichlet Process

- Hierarchical Dirichlet Process for Dynamic sequence
  *Dynamic Hierarchial Dirichlet Process*
  *(sharing some parameters across the groups)*

- Application of dHDP
  *Music segmentation*

## References:

*Lu Ren, David B. Dunson, and Lawrence Carin. 2008. The dynamic hierarchical Dirichlet process. In Proceedings of the 25th international conference on Machine learning (ICML '08). ACM, New York, NY, USA, 824-831. DOI= http://dx.doi.org/10.1145/1390156.1390260*

*Lecture notes on Bayesian Nonparametrics, Peter Orbanz*
*http://stat.columbia.edu/~porbanz/papers/porbanz_BNP_draft.pdf*

*http://stat.columbia.edu/~porbanz/talks/MLSS12_1.pdf*

*Bayesian Nonparametrics, Yee Whye Teh, Dept of Statistics, Oxford*
*http://mlss.tuebingen.mpg.de/2013/2013/slides_teh.pdf*

# Questions / Comments ??