BML Lecture #6

Bayesian nonparametrics

Julyan Arbel

Outline

Bayesian nonparametrics

- Pitman-Yor process, Gibbs-type process
- Hierarchical Dirichlet process
- Indian buffet process
- Practical: Dirichlet process mixture models in Pyro

Outline

Bayesian nonparametrics

- Pitman-Yor process, Gibbs-type process
- Hierarchical Dirichlet process
- Indian buffet process
- Practical: Dirichlet mixture models in Pyro

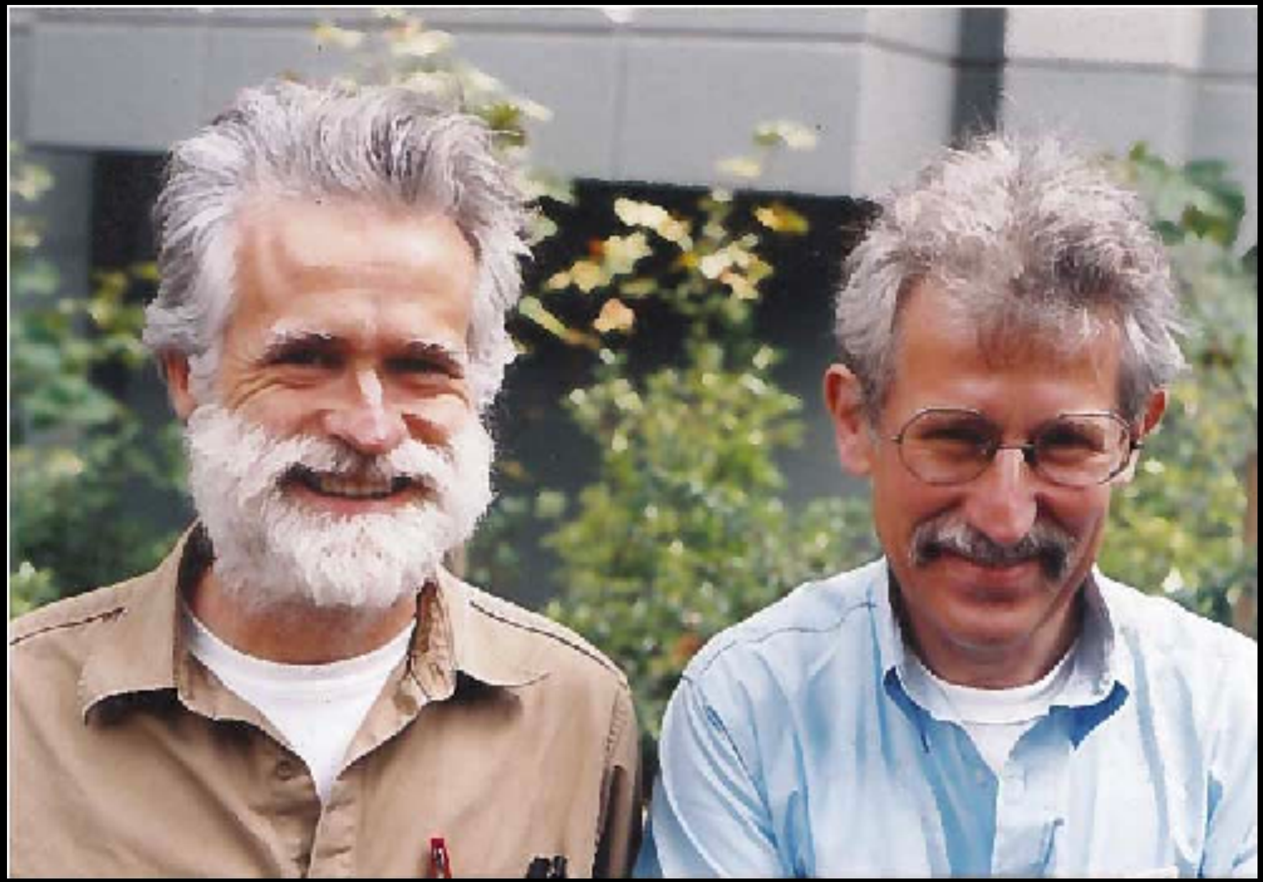
Bayesian deep learning

- Maximum a posteriori = Regularized maximum likehood
- Laplace approximation (MacKay, 1992, Neur. Comp.)
- Variational inference (Hinton and van Kamp, 1993, Barber & Bishop, 1998, NIPS)
- Monte Carlo dropout (Gal & Ghahramani, 2016, ICML)
- Practical: Bayesian neural networks in Pyro

Bayesian nonparametrics

Pitman-Yor process

- Dirichlet process
  - Ferguson, 1973

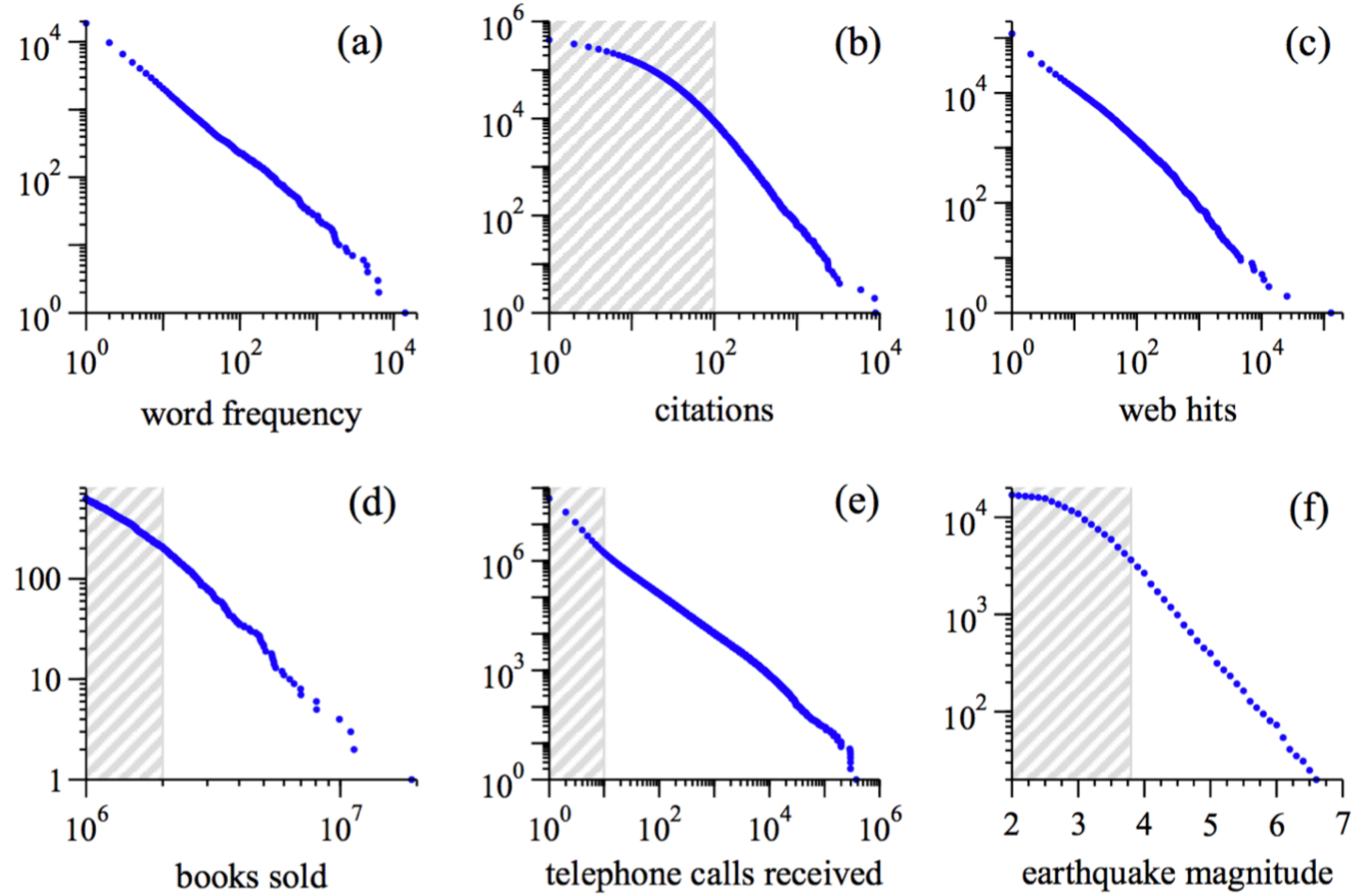


- Pitman-Yor process
  - Perman, Pitman & Yor, 1992
  - Pitman & Yor, 1997

Bayesian nonparametrics

Pitman-Yor process

- Depart from log #clusters to power-law #clusters



[Image from Newman [2005]]

Bayesian nonparametrics

Pitman-Yor process: diversity

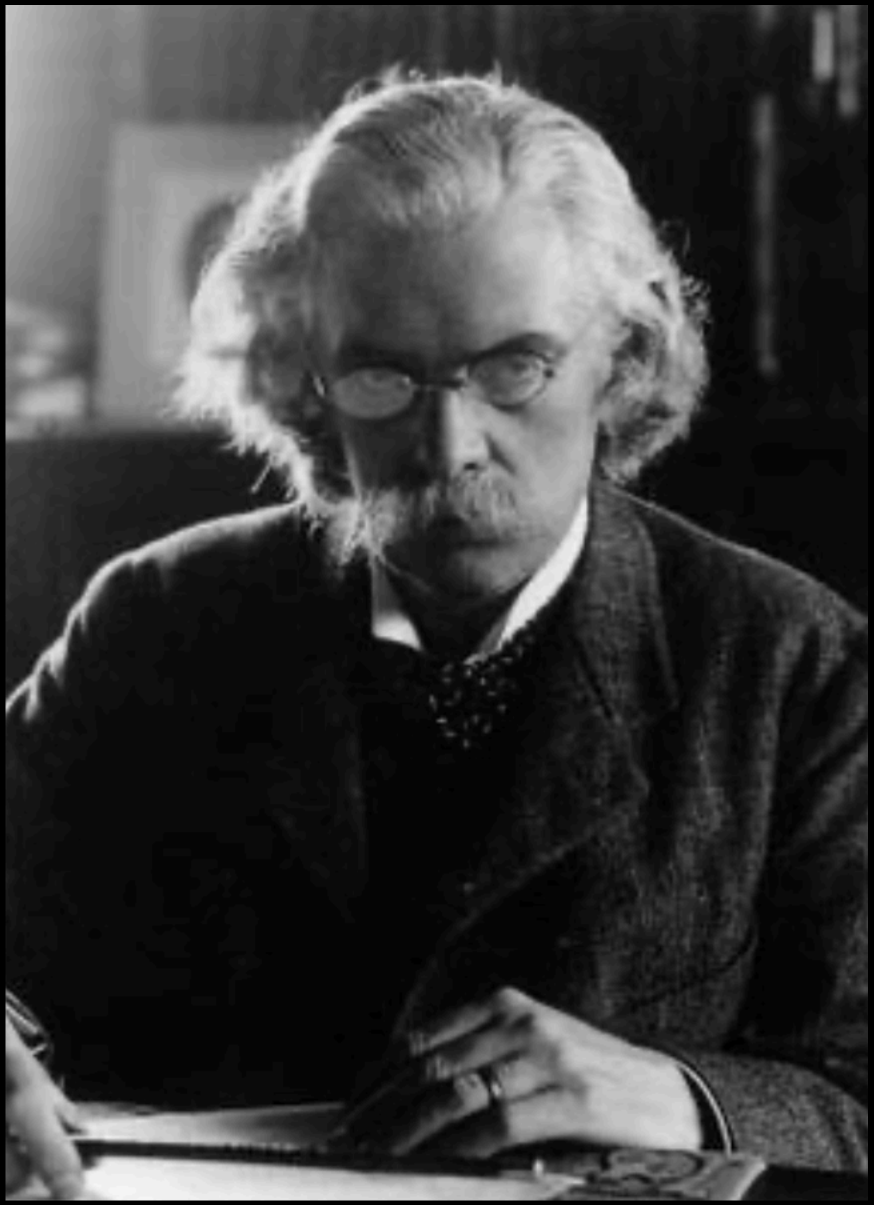
Power Law and σ-diversity
For $\sigma > 0$ we have the almost sure convergence

$$n^{-\sigma} K_n \to S_{\sigma,\alpha},$$

where $S_{\sigma,\alpha}$ is called σ-diversity of the PY, whose density is a polynomially tilted **Mittag–Leffler density** *(ML):*

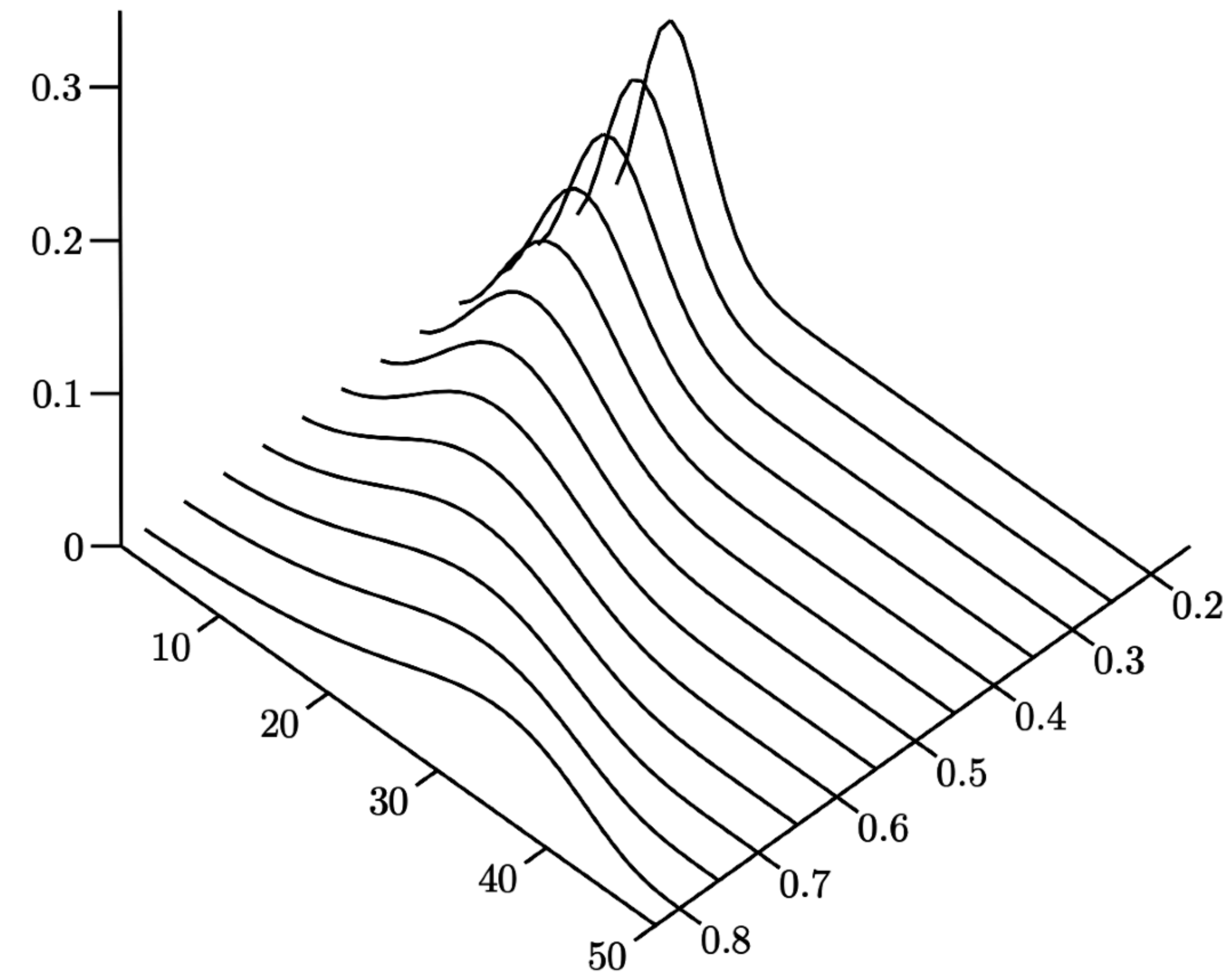
$$g_{\sigma,\alpha}(x) \propto x^{\alpha/\sigma} g_{\alpha}(x),$$

and g_α is ML density.



Bayesian nonparametrics

Pitman-Yor process: number of clusters

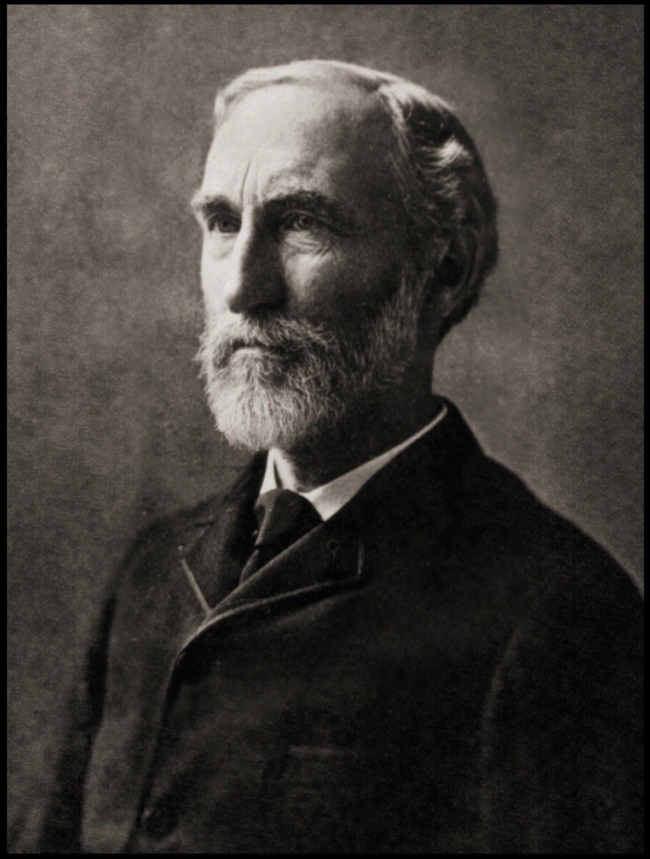


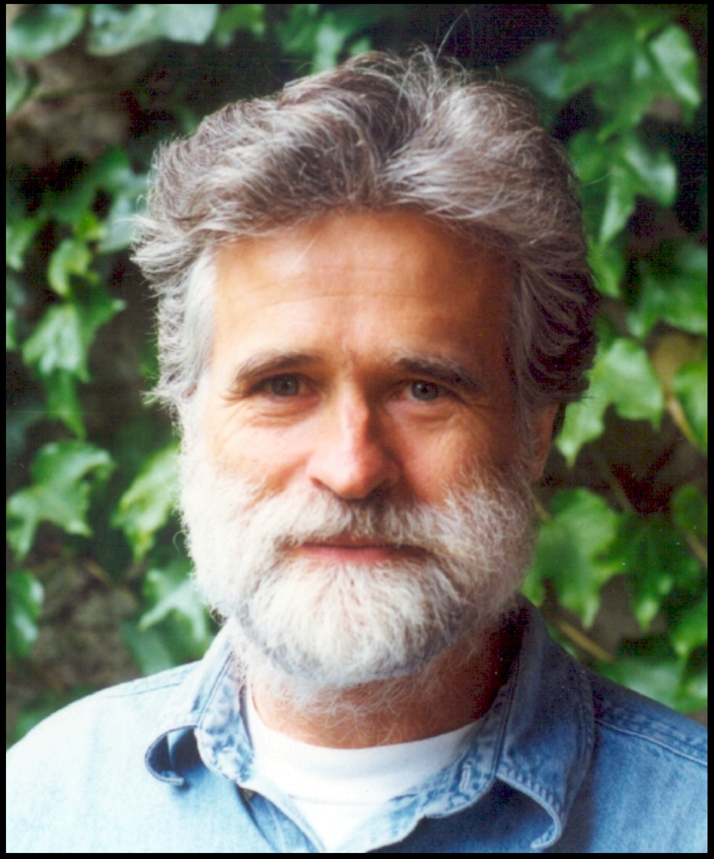
[Image by De Blasi et al. [2015]]

Bayesian nonparametrics

Gibbs-type process

- Pitman, 2003





Bayesian nonparametrics

Dirichlet ⊂ Pitman-Yor ⊂ Gibbs-type

	Conjugacy	Stick-break.	Discovery proba	K_n growth
Gibbs	No	not ind not id	(n, K_n)	$\dots n^{\sigma}$
PY	No	ind		
DP	Yes	iid	n	$\alpha \log n$

Bayesian nonparametrics

Hierarchical Dirichlet process

- Extension of Latent Dirichlet Allocation (Blei, Ng, and Jordan, 2003, JMLR)
- to infinite dimensional parameter space (Teh, Jordan, Beal, and Blei, JASA, 2006)




Bayesian nonparametrics

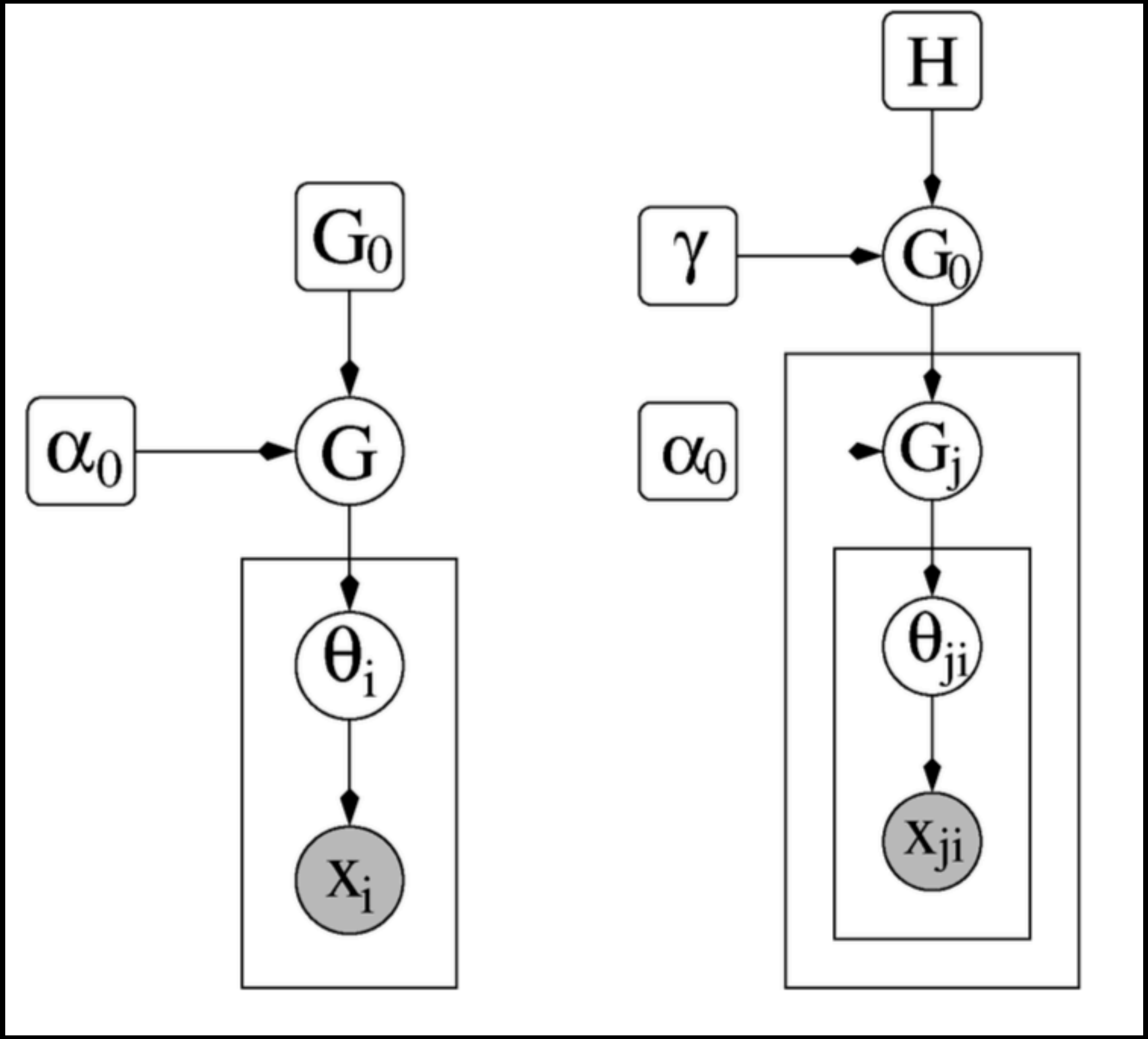
Hierarchical Dirichlet process

- Extension of Latent Dirichlet Allocation (Blei, Ng, and Jordan, 2003, JMLR)
- to infinite dimensional parameter space (Teh, Jordan, Beal, and Blei, JASA, 2006)




Bayesian nonparametrics

Hierarchical Dirichlet process



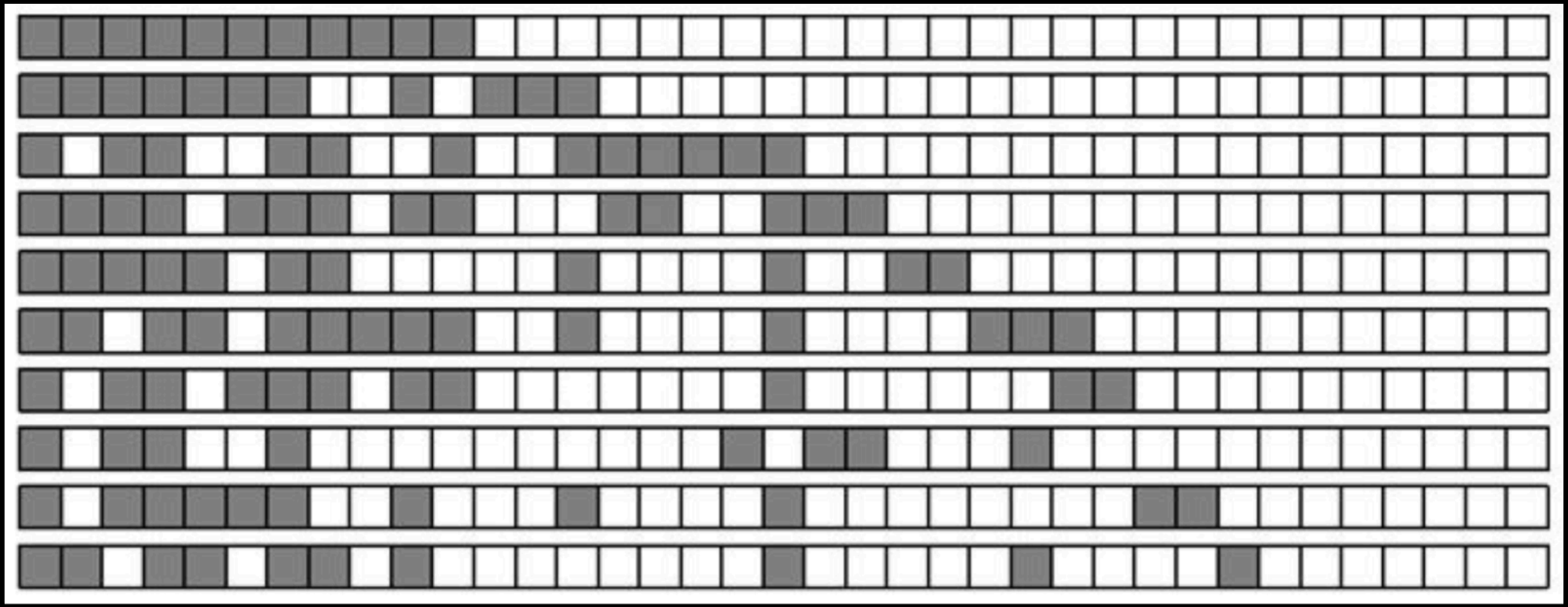
Bayesian nonparametrics

Indian buffet process

- Extension of Dirichlet process (Ferguson, 2003, AOS) to multiple classes/allocation models
-

Bayesian nonparametrics

Indian buffet process



Bayesian nonparametrics

Practical: Pyro, pyro.ai

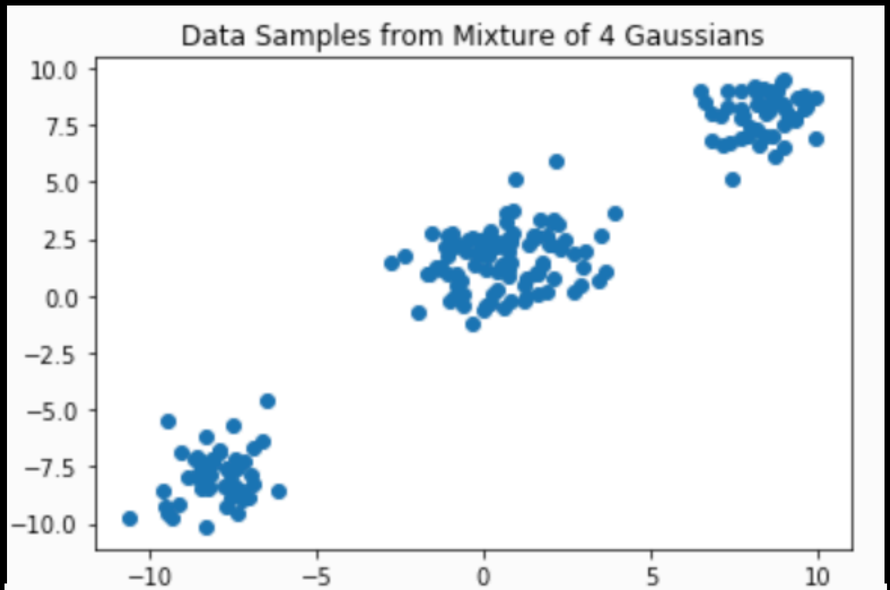
Pyro is a universal probabilistic programming language (PPL) written in Python and supported by PyTorch on the backend. Pyro enables flexible and expressive deep probabilistic modeling, unifying the best of modern deep learning and Bayesian modeling. It was designed with these key principles:

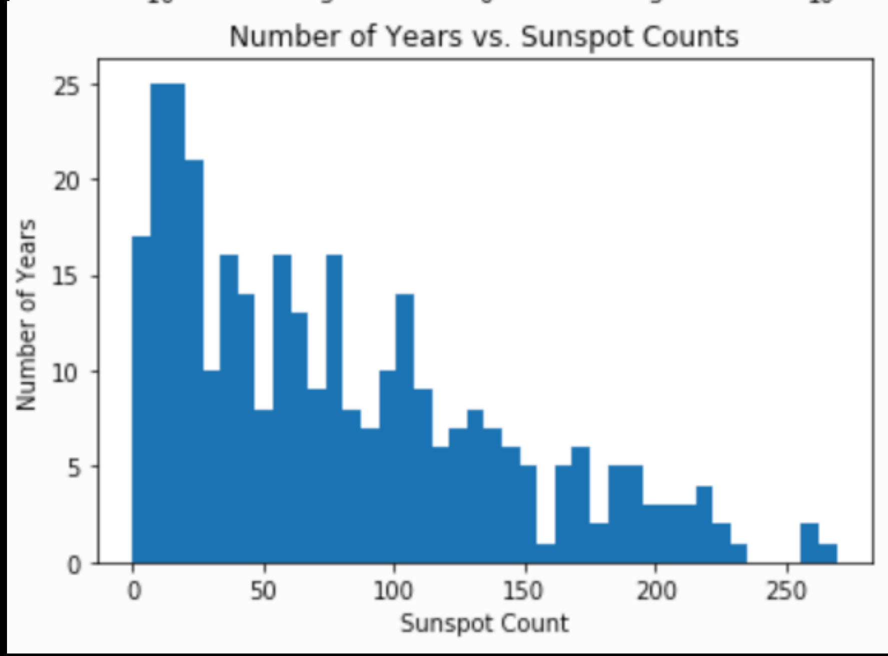
Universal: Pyro can represent any computable probability distribution.
Scalable: Pyro scales to large data sets with little overhead.
Minimal: Pyro is implemented with a small core of powerful, composable abstractions.
Flexible: Pyro aims for automation when you want it, control when you need it.

Bayesian nonparametrics

Practical: Dirichlet process mixture models in Pyro

- https://pyro.ai/examples/dirichlet_process_mixture.html
- Implements Variational inference based on stick-breaking
    - On Synthetic Mixture of Gaussians
    - On Long Term Solar Observations
- Objectives:
    - Install Pyro & PyTorch (or work on Colab)
    - Run the code
    - Write your own code for Pitman-Yor mixture models





Bayesian deep learning

Bayesian neural networks

- Maximum a posteriori = Regularized maximum likehood
- Laplace approximation (MacKay, 1992, Neur. Comp.)
- Variational inference (Hinton and van Kamp, 1993, Barber & Bishop, 1998, NIPS)
- Monte Carlo dropout (Gal & Ghahramani, 2016, ICML)
- Implementation: Pyro & PyTorch

Bayesian deep learning

Pyro example on Bayesian neural networks

- Demonstrates how to use NUTS to do inference on a simple (small) Bayesian neural network with two hidden layers.