Bayesian Models in Machine Learning

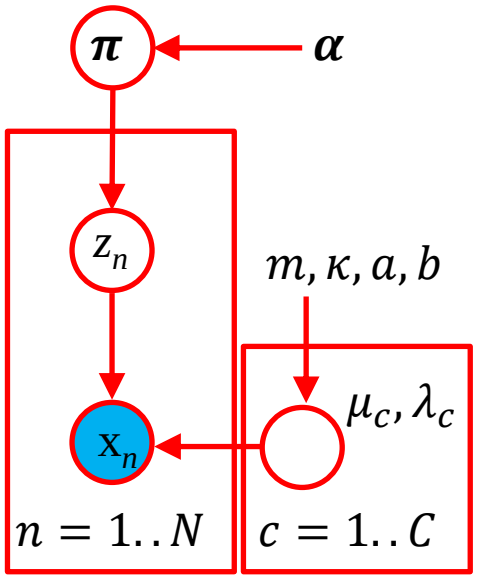
Approximate inference in Bayesian models

Lukáš Burget

BRNO UNIVERSITY OF TECHNOLOGY
FACULTY OF INFORMATION TECHNOLOGY

BAYa lectures, November 2024

Bayesian Gaussian Mixture Model



- We assume that the observed data were generated as follows:
  - $\boldsymbol{\pi} \sim \mathrm{Dir}(\boldsymbol{\alpha})$
  - For Gaussian component $c = 1 \ldots C$
    - $\mu_c, \lambda_c \sim \mathrm{NormalGamma}(\mu_c, \lambda_c | m, \kappa, a, \mathrm{b})$
  - For each observation $n = 1 \ldots N$
    - $z_n \sim \mathrm{P}(z_n|\boldsymbol{\pi}) = \mathrm{Cat}(z_n|\boldsymbol{\pi})$
    - $x_n \sim p(x_n|z_n, \boldsymbol{\mu}, \boldsymbol{\lambda}) = \mathcal{N}\left(x_n|\mu_{z_n},\ \lambda_{z_n}^{-1}\right)$
- The task is to infer the posterior distribution of parameters $\mathrm{p}(\boldsymbol{\pi}, \mu_1, \lambda_1, \ldots \mu_C, \lambda_C|\mathbf{x})$ given some observed data $\mathbf{x} = [x_1, x_2, \ldots, x_N]$
- Intractable: need for approximations

$$p(\mathbf{x}, \mathbf{z}, \boldsymbol{\mu}, \boldsymbol{\lambda}, \boldsymbol{\pi}) = \prod_n p(x_n|z_n, \boldsymbol{\mu}, \boldsymbol{\lambda}) \prod_n P(z_n|\boldsymbol{\pi}) \prod_c p(\mu_c, \lambda_c) p(\boldsymbol{\pi})$$

Approximate inference (for Bayesian GMM)

- Variational Bayes
  - Approximate intractable $p(\boldsymbol{\mu}, \boldsymbol{\lambda}, \boldsymbol{\pi}, \mathbf{z}|\mathbf{X})$ with tractable $q(\boldsymbol{\mu}, \boldsymbol{\lambda}, \boldsymbol{\pi}, \mathbf{z})$
  - Iteratively tune parameters of $q(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\lambda}, \mathbf{z})$ minimize $D_{KL}\ (q(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\lambda}, \mathbf{z})||p(\boldsymbol{\mu}, \boldsymbol{\lambda}, \boldsymbol{\pi}, \boldsymbol{z}|\mathbf{X}))$
- Gibbs sampling
  - Instead of obtaining $p(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\lambda}, \boldsymbol{z}|\mathbf{X})$, we only generate samples from this distribution
  - Integrating over $p(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\lambda}, \boldsymbol{z}|\mathbf{X})$ (e.g. for predictive distribution) can be approximated with *empirical expectations*
- …

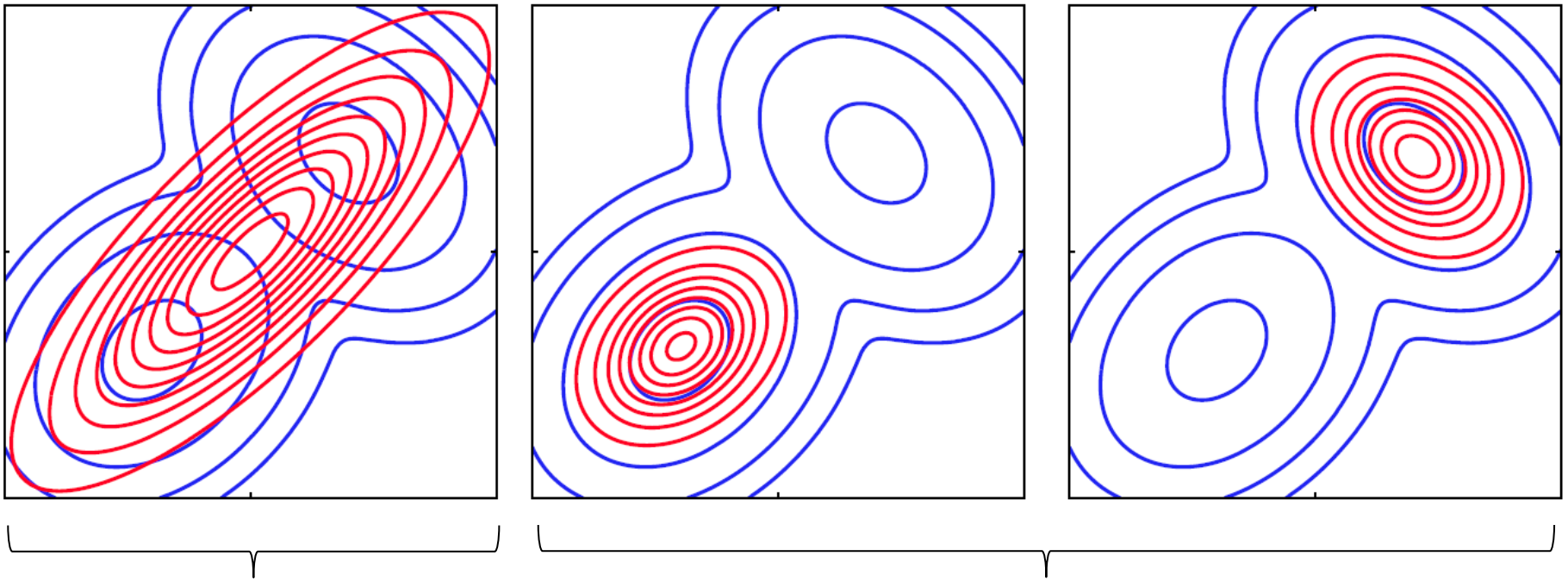
Variational Bayes

$$\ln p(\mathbf{X}) = \underbrace{\int q(\mathbf{Y}) \ln p(\mathbf{X},\mathbf{Y})\, \mathrm{d}\mathbf{Y} - \int q(\mathbf{Y}) \ln q(\mathbf{Y})\, \mathrm{d}\mathbf{Y}}_{\mathcal{L}(q(\mathbf{Y}))} \underbrace{- \int q(\mathbf{Y}) \ln \frac{p(\mathbf{Y}|\mathbf{X})}{q(\mathbf{Y})}\, \mathrm{d}\mathbf{Y}}_{D_{KL}(q(\mathbf{Y})||p(\mathbf{Y}|\mathbf{X}))}$$

- Find $q(\mathbf{Y})$, which is good approximation for the true posterior $p(\mathbf{Y}|\mathbf{X})$
- Maximize $\mathcal{L}\big(q(\mathbf{Y})\big)$ w.r.t. $q(\mathbf{Y})$, which in turn minimizes $D_{KL}(q(\mathbf{Y})\big|\big|p(\mathbf{Y}|\mathbf{X})\big)$
  - "Handcraft" a reasonable parametric distribution $q(\mathbf{Y}|\boldsymbol{\eta})$ and optimize $\mathcal{L}\big(q(\mathbf{Y}|\boldsymbol{\eta})\big)$ w.r.t. its parameters $\boldsymbol{\eta}$.
  - Mean field approximation assuming factorized form $q(\mathbf{Y})=q(\mathbf{Y}_1)q(\mathbf{Y}_2)q(\mathbf{Y}_3)\ldots$

Minimizing Kullback-Leibler divergence

- We optimize parameters of (simpler) distribution $q(\mathbf{Y})$ to minimize Kullback-Leibler divergence between $q(\mathbf{Y})$ and $p(\mathbf{Y}|\mathbf{X})$.



- Minimizing $D_{KL}(p(\mathbf{Y}|\mathbf{X})||q(\mathbf{Y}))$.
- Not VB objective
- Expectation propagation

- Two local optima when (numerically) minimizing $D_{KL}(q(\mathbf{Y})||p(\mathbf{Y}|\mathbf{X}))$.
- VB performs this optimization

VB – Mean field approximation

- Popular Variational Bayes optimization method
- Variant of Variational Bayes, where the set of model variables $\mathbf{Y}$, can be split into subsets $\mathbf{Y}_1, \mathbf{Y}_2, \mathbf{Y}_3$, ..., with conditionally conjugate priors
  - $p(\mathbf{Y}_\mathrm{i}|\mathbf{X}, \mathbf{Y}_{\forall \mathrm{j} \neq \mathrm{i}})$ is tractable with conjugate prior
  - E.g. for Bayesian GMM $p(\mu_c, \lambda_c|\mathbf{X}, \mathbf{z})$ has $\mathrm{NormalGamma}$ prior
- We assume factorized approximate posterior

$$q(\mathbf{Y}) = q(\mathbf{Y}_1) q(\mathbf{Y}_2) q(\mathbf{Y}_3) \ldots = \prod_{\boldsymbol{i}} q(\mathbf{Y}_\mathrm{i})$$

- This factorization dictates the optimal (conjugate) distributions for the factors $q(\mathbf{Y}_\mathrm{i})$ and brings well defined iterative update formulas:

$$q(\mathbf{Y}_i)^* \propto \exp\left(\int q(\mathbf{Y}_{\forall \mathrm{j} \neq \mathrm{i}}) \ln p(\mathbf{X}, \mathbf{Y}) \, \mathrm{d}\mathbf{Y}_{\forall \mathrm{j} \neq \mathrm{i}} \right)$$

Mean field - update

$$\mathcal{L}(q(\mathbf{Y})) = \int q(\mathbf{Y}) \ln p(\mathbf{X}, \mathbf{Y}) \, d\mathbf{Y} - \int q(\mathbf{Y}) \ln q(\mathbf{Y}) \, d\mathbf{Y} = \int \prod_{i=1}^{M} q(\mathbf{Y}_i) \left[\ln p(\mathbf{X}, \mathbf{Y}) - \ln \prod_i q(\mathbf{Y}_i) \right] d\mathbf{Y}$$

$$= \int \prod_{i=1}^{M} q(\mathbf{Y}_i) \left[\ln p(\mathbf{X}, \mathbf{Y}) - \sum_i \ln q(\mathbf{Y}_i) \right] d\mathbf{Y}$$

- For example, let $M = 3$
- Now, lets optimize the lower bound $\mathcal{L}(q(\mathbf{Y}_1))$ w.r.t only one distribution $q(\mathbf{Y}_1)$

$$\mathcal{L}(q(\mathbf{Y}_1)) = \iiint q(\mathbf{Y}_1) q(\mathbf{Y}_2) q(\mathbf{Y}_3) \left[\ln p(\mathbf{X}, \mathbf{Y}_1, \mathbf{Y}_2, \mathbf{Y}_3) - \ln q(\mathbf{Y}_1) - \ln q(\mathbf{Y}_2) - \ln q(\mathbf{Y}_3) \right] d\mathbf{Y}_1 \, d\mathbf{Y}_2 \, d\mathbf{Y}_3$$

$$= \int q(\mathbf{Y}_1) \underbrace{\iint q(\mathbf{Y}_2) q(\mathbf{Y}_3) \ln p(\mathbf{X}, \mathbf{Y}_1, \mathbf{Y}_2, \mathbf{Y}_3) \, d\mathbf{Y}_2 \, d\mathbf{Y}_3}_{\ln \tilde{p}(\mathbf{Y}_1) + const} d\mathbf{Y}_1 - \int q(\mathbf{Y}_1) \ln q(\mathbf{Y}_1) \, d\mathbf{Y}_1 + const$$

$$= \int q(\mathbf{Y}_1) \ln \tilde{p}(\mathbf{Y}_1) \, d\mathbf{Y}_1 - \int q(\mathbf{Y}_1) \ln q(\mathbf{Y}_1) \, d\mathbf{Y}_1 + const = -D_{KL}(q(\mathbf{Y}_1) || \tilde{p}(\mathbf{Y}_1)) + const$$

where $\tilde{p}(\mathbf{Y}_1)$ is normalized to be a valid distribution (therefore $+const$)

- $\mathcal{L}(q(\mathbf{Y}_1))$ is maximized by setting the D_{KL} term to zero, which implies

$$\ln q(\mathbf{Y}_1) = \ln \tilde{p}(\mathbf{Y}_1) = \iint q(\mathbf{Y}_2) q(\mathbf{Y}_3) \ln p(\mathbf{X}, \mathbf{Y}_1, \mathbf{Y}_2, \mathbf{Y}_3) \, d\mathbf{Y}_2 \, d\mathbf{Y}_3 + const$$

- In general, we can iteratively update each $q(\mathbf{Y}_i)$ given the others $q(\mathbf{Y}_{i \neq j})$ as:

$$q(\mathbf{Y}_j) \propto \exp \int q(\mathbf{Y}_{\forall j \neq i}) \ln p(\mathbf{X}, \mathbf{Y}) \, d\mathbf{Y}_{\forall j \neq i}$$

where each update guaranties to improve the lower bound $\mathcal{L}(q(\mathbf{Y}))$

Variational Bayes for GMM

- Joint likelihood for Bayesian GMM

$$p(\mathbf{x}, \mathbf{z}, \boldsymbol{\mu}, \boldsymbol{\lambda}, \boldsymbol{\pi}) = \prod_n p(x_n|z_n, \boldsymbol{\mu}, \boldsymbol{\lambda}) \prod_n P(z_n|\boldsymbol{\pi}) \prod_c p(\mu_c, \lambda_c) p(\boldsymbol{\pi})$$

$$\ln p(\mathbf{x}, \mathbf{z}, \boldsymbol{\mu}, \boldsymbol{\lambda}, \boldsymbol{\pi}) = \sum_n \ln p(x_n|z_n, \boldsymbol{\mu}, \boldsymbol{\lambda}) + \sum_n \ln P(z_n|\boldsymbol{\pi}) + \sum_c \ln p(\mu_c, \lambda_c) + \ln p(\boldsymbol{\pi})$$

where

$$\begin{aligned} p(x_n|z_n = c, \boldsymbol{\mu}, \boldsymbol{\lambda}) &= \mathcal{N}(x_n; \mu_c, \lambda_c^{-1}) \\ P(z_n = c|\boldsymbol{\pi}) &= \mathrm{Cat}(z_n = c|\boldsymbol{\pi}) = \pi_c \\ p(\mu_c, \lambda_c) &= \mathrm{NormalGamma}(\mu_c, \lambda_c|m, k, a, b) \\ p(\boldsymbol{\pi}) &= \mathrm{Dir}(\boldsymbol{\pi}|\boldsymbol{\alpha}) \end{aligned}$$

- Mean field approximation $q(\boldsymbol{\mu}, \boldsymbol{\lambda}, \boldsymbol{\pi}, \mathbf{z}) = q(\mathbf{z})q(\boldsymbol{\mu}, \boldsymbol{\lambda}, \boldsymbol{\pi})$ dictates updates:

$$q(\mathbf{z})^* \propto \exp\left(\int q(\boldsymbol{\mu}, \boldsymbol{\lambda}, \boldsymbol{\pi}) \ln p(\mathbf{x}, \mathbf{z}, \boldsymbol{\mu}, \boldsymbol{\lambda}, \boldsymbol{\pi}) \, \mathrm{d}\boldsymbol{\mu} \, \mathrm{d}\boldsymbol{\lambda} \, \mathrm{d}\boldsymbol{\pi}\right)$$

$$q(\boldsymbol{\mu}, \boldsymbol{\lambda}, \boldsymbol{\pi})^* \propto \exp\left(\sum_{\mathbf{z}} q(\mathbf{z}) \ln p(\mathbf{x}, \mathbf{z}, \boldsymbol{\mu}, \boldsymbol{\lambda}, \boldsymbol{\pi})\right)$$

VBGMM – update for $q(\boldsymbol{z})$

$$
\begin{aligned}
q(\mathbf{z})^* &\propto \exp\left(\int q(\boldsymbol{\mu}, \boldsymbol{\lambda}, \boldsymbol{\pi}) \ln p(\mathbf{x}, \mathbf{z}, \boldsymbol{\mu}, \boldsymbol{\lambda}, \boldsymbol{\pi}) \, \mathrm{d}\boldsymbol{\mu} \, \mathrm{d}\boldsymbol{\lambda} \, \mathrm{d}\boldsymbol{\pi}\right) \\
&\propto \exp\left(\int q(\boldsymbol{\mu}, \boldsymbol{\lambda}, \boldsymbol{\pi}) \left(\sum_n \ln p(x_n|z_n, \boldsymbol{\mu}, \boldsymbol{\lambda}) + \sum_n \ln p(z_n|\boldsymbol{\pi})\right) \mathrm{d}\boldsymbol{\mu} \, \mathrm{d}\boldsymbol{\lambda} \, \mathrm{d}\boldsymbol{\pi}\right) \\
&= \exp\left(\sum_n \int q(\boldsymbol{\mu}, \boldsymbol{\lambda}, \boldsymbol{\pi}) \left(\ln p(x_n|z_n, \boldsymbol{\mu}, \boldsymbol{\lambda}) + \ln p(z_n|\boldsymbol{\pi})\right) \mathrm{d}\boldsymbol{\mu} \, \mathrm{d}\boldsymbol{\lambda} \, \mathrm{d}\boldsymbol{\pi}\right) \\
&\propto \prod_n q(z_n)^*
\end{aligned}
$$

- We see that $q(\mathbf{z})$ further factorizes - so called induced factorization

Similar to responsibilities from EM

$$
\begin{aligned}
q(z_n = c)^* &= \gamma_{nc} \\
&\propto \exp\left(\int q(\boldsymbol{\mu}, \boldsymbol{\lambda}) \ln \mathcal{N}(x_n; \mu_c, \lambda_c^{-1}) \, \mathrm{d}\boldsymbol{\mu} \, \mathrm{d}\boldsymbol{\lambda} + \int q(\boldsymbol{\pi}) \ln \mathrm{Cat}(z_n = c|\boldsymbol{\pi}) \, \mathrm{d}\boldsymbol{\pi}\right)
\end{aligned}
$$

VBGMM – update for $q(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\lambda})$

$$q(\boldsymbol{\mu}, \boldsymbol{\lambda}, \boldsymbol{\pi})^* \propto \exp\left(\sum_{\mathbf{z}} q(\mathbf{z}) \ln p(\mathbf{x}, \mathbf{z}, \boldsymbol{\mu}, \boldsymbol{\lambda}, \boldsymbol{\pi})\right)$$

$$= \exp\left(\sum_{\mathbf{z}} \prod_{n} q(z_n) \sum_{n} \{\ln p(x_n|z_n, \boldsymbol{\mu}, \boldsymbol{\lambda}) + \ln p(z_n|\boldsymbol{\pi})\} + \sum_{c} \ln p(\mu_c, \lambda_c) + \ln p(\boldsymbol{\pi})\right)$$

$$= \exp\left(\sum_{c} \sum_{n} \gamma_{nc} \{\ln p(x_n|z_n = c, \boldsymbol{\mu}, \boldsymbol{\lambda}) + \ln p(z_n = c|\boldsymbol{\pi})\} + \ln p(\mu_c, \lambda_c) + \ln p(\boldsymbol{\pi})\right)$$

$$= \prod_{c} \left[\exp\left(\sum_{n} \gamma_{nc} \ln \mathcal{N}(x; \mu_c, \lambda_c^{-1})\right) p(\mu_c, \lambda_c)\right] \exp\left(\sum_{c} \sum_{n} \gamma_{nc} \ln p(z_n = c|\boldsymbol{\pi})\right) p(\boldsymbol{\pi})$$

$$\propto \prod_{c} q(\mu_c, \lambda_c)^* q(\boldsymbol{\pi})^*$$

- Again, we obtain induced factorization for $q(\boldsymbol{\mu}, \boldsymbol{\lambda}, \boldsymbol{\pi})$

$$q(\mu_c, \lambda_c)^* \propto \exp\left(\sum_{n} \gamma_{nc} \ln \mathcal{N}(x; \mu_c, \lambda_c^{-1})\right) \text{NormalGamma}(\mu_c, \lambda_c|m, \kappa, a, b)$$

$$q(\boldsymbol{\pi})^* \propto \exp\left(\sum_{c} \sum_{n} \gamma_{nc} \ln \text{Cat}(z_n = c|\boldsymbol{\pi})\right) \text{Dir}(\boldsymbol{\pi}|\boldsymbol{\alpha})$$

Flashback - Factorization over components

Example with only 3 fames (i.e $\mathbf{z} = [z_1, z_2, z_3]$)

$$\sum_{\mathbf{z}} \prod_{n} q(z_n) \sum_{n} f(z_n) =$$

$$\sum_{z_1}\sum_{z_2}\sum_{z_3} q(z_1)q(z_2)q(z_3)f(z_1) + \sum_{z_1}\sum_{z_2}\sum_{z_3} q(z_1)q(z_2)q(z_3)f(z_2) + \sum_{z_1}\sum_{z_2}\sum_{z_3} q(z_1)q(z_2)q(z_3)f(z_3) =$$

$$\sum_{z_1} q(z_1) f(z_1) \sum_{z_2} q(z_2) \sum_{z_3} q(z_3) + \sum_{z_1} q(z_1) \sum_{z_2} q(z_2) f(z_2) \sum_{z_3} q(z_3) + \sum_{z_1} q(z_1) \sum_{z_2} q(z_2) \sum_{z_3} q(z_3) f(z_3) =$$

$$\sum_{z_1} q(z_1) f(z_1) + \sum_{z_2} q(z_2) f(z_2) + \sum_{z_3} q(z_3) f(z_3) =$$

$$\sum_{c=1}^{C} q(z_1 = c) f(z_1 = c) + \sum_{c=1}^{C} q(z_2 = c) f(z_2 = c) + \sum_{c=1}^{C} q(z_3 = c) f(z_3 = c) =$$

$$\sum_{c=1}^{C} \sum_{n} q(z_n = c) f(z_n = c)$$

VBGMM – update for $q(\mu_c, \lambda_c)$

$$q(\mu_c, \lambda_c)^* \propto \exp\left(\sum_n \gamma_{nc} \ln \mathcal{N}(x; \mu_c, \lambda_c^{-1})\right) \text{NormalGamma}(\mu_c, \lambda_c | m, \kappa, a, b)$$

$$= \prod_n \mathcal{N}(x; \mu_c, \lambda_c^{-1})^{\gamma_{nc}} \text{NormalGamma}(\mu_c, \lambda_c | m, \kappa, a, b)$$

$$\propto \text{NormalGamma}\left(\mu_c, \lambda_c | \frac{\kappa m + N_c \bar{x}_c}{\kappa + N_c}, \kappa + N_c, a + \frac{N_c}{2}, b + \frac{N_c}{2}\left(s_c + \frac{\kappa(\bar{x}_c - m)^2}{\kappa + N_c}\right)\right)$$

$$\propto \text{NormalGamma}\left(\mu_c, \lambda_c | m_c^*, \kappa_c^*, a_c^*, b_c^*,\right)$$

$$N_c = \sum_n \gamma_{nc}$$

$$\bar{x}_c = \frac{\sum_n \gamma_{nc} x_n}{\sum_n \gamma_{nc}}$$

$$s_c = \frac{\sum_n \gamma_{nc}(x_n - \bar{x}_c)^2}{\sum_n \gamma_{nc}}$$

Updating distribution q(μ_c, λ_c) means updating the parameters $m_c^*, \kappa_c^*, a_c^*, b_c^*$

VBGMM – update for $q(\boldsymbol{\pi})$

$$
\begin{aligned}
q(\boldsymbol{\pi})^* &\propto \exp\left(\sum_c \sum_n \gamma_{nc} \ln \mathrm{Cat}(z_n = c|\boldsymbol{\pi})\right) \mathrm{Dir}(\boldsymbol{\pi}|\boldsymbol{\alpha}) \\
&\propto \mathrm{Dir}(\boldsymbol{\pi}|\boldsymbol{\alpha} + \mathbf{N}) \\
&\propto \mathrm{Dir}(\boldsymbol{\pi}|\boldsymbol{\alpha}^*)
\end{aligned}
$$

$$
\begin{aligned}
\mathbf{N} &= [N_1, N_2 \ldots, N_C] \\
N_c &= \sum_n \gamma_{nc}
\end{aligned}
$$

Updating distributions$\mathrm{q}(\boldsymbol{\pi})$ means updating the vector $\boldsymbol{\alpha}^* = [\alpha_1^*, \alpha_2^*, \ldots \alpha_C^*]$

VBGMM – update for $q(z_n)$

$$
\begin{aligned}
q(z_n = c)^* &\propto \exp\left(\int q(\mu_c, \lambda_c) \ln \mathcal{N}(x_n; \mu_c, \lambda_c^{-1}) \, \mathrm{d}\mu_c \, \mathrm{d}\lambda_c + \int q(\boldsymbol{\pi}) \ln \mathrm{Cat}(z_n = c|\boldsymbol{\pi}) \, \mathrm{d}\boldsymbol{\pi}\right) \\
&\propto \exp\left(\psi(\alpha_c^*) - \psi\left(\sum_c \alpha_c^*\right) + \frac{\psi(a_c^*) - \ln b_c^*}{2} - \frac{1}{2\kappa_c^*} - \frac{a_c^*}{2b_c^*}(x_n - m_c^*)^2\right) \\
&= \rho_{nc}
\end{aligned}
$$

$$
q(z_n = c)^* = \gamma_{nc} = \frac{\rho_{nc}}{\sum_k \rho_{nk}}
$$

where $\psi(.)$ is digamma function

Updating distributions $\mathrm{q}(z_n)$ means computing responsibilities γ_{nc}

Summary of VB-GMM updates

- Update distributions $\mathrm{q}(z_n)$ (i.e. the responsibilities γ_{nc}):

$$\rho_{nc} = \exp\left(\psi(\alpha_c^*) - \psi\left(\sum_c \alpha_c^*\right) + \frac{\psi(a_c^*) - \ln b_c^*}{2} - \frac{1}{2\kappa_c^*} - \frac{a_c^*}{2b_c^*}(x_n - m_c^*)^2\right)$$

$$\gamma_{nc} = \frac{\rho_{nc}}{\sum_k \rho_{nk}}$$

- For all $c = 1..C$, update parameters of $\mathrm{q}(\mu_c, \lambda_c)$ and $\mathrm{q}(\boldsymbol{\pi})$:

$$m_c^* = \frac{\kappa m + N_c \bar{x}_c}{\kappa + N_c}$$

$$\kappa_c^* = \kappa + N_c$$

$$a_c^* = a + \frac{N_c}{2}$$

$$b_c^* = b + \frac{N_c}{2}\left(s_c + \frac{\kappa(\bar{x}_c - m)^2}{\kappa + N_c}\right)$$

$$N_c = \sum_n \gamma_{nc}$$

$$\bar{x}_c = \frac{\sum_n \gamma_{nc} x_n}{\sum_n \gamma_{nc}}$$

$$s_c = \frac{\sum_n \gamma_{nc}(x_n - \bar{x}_c)^2}{\sum_n \gamma_{nc}}$$

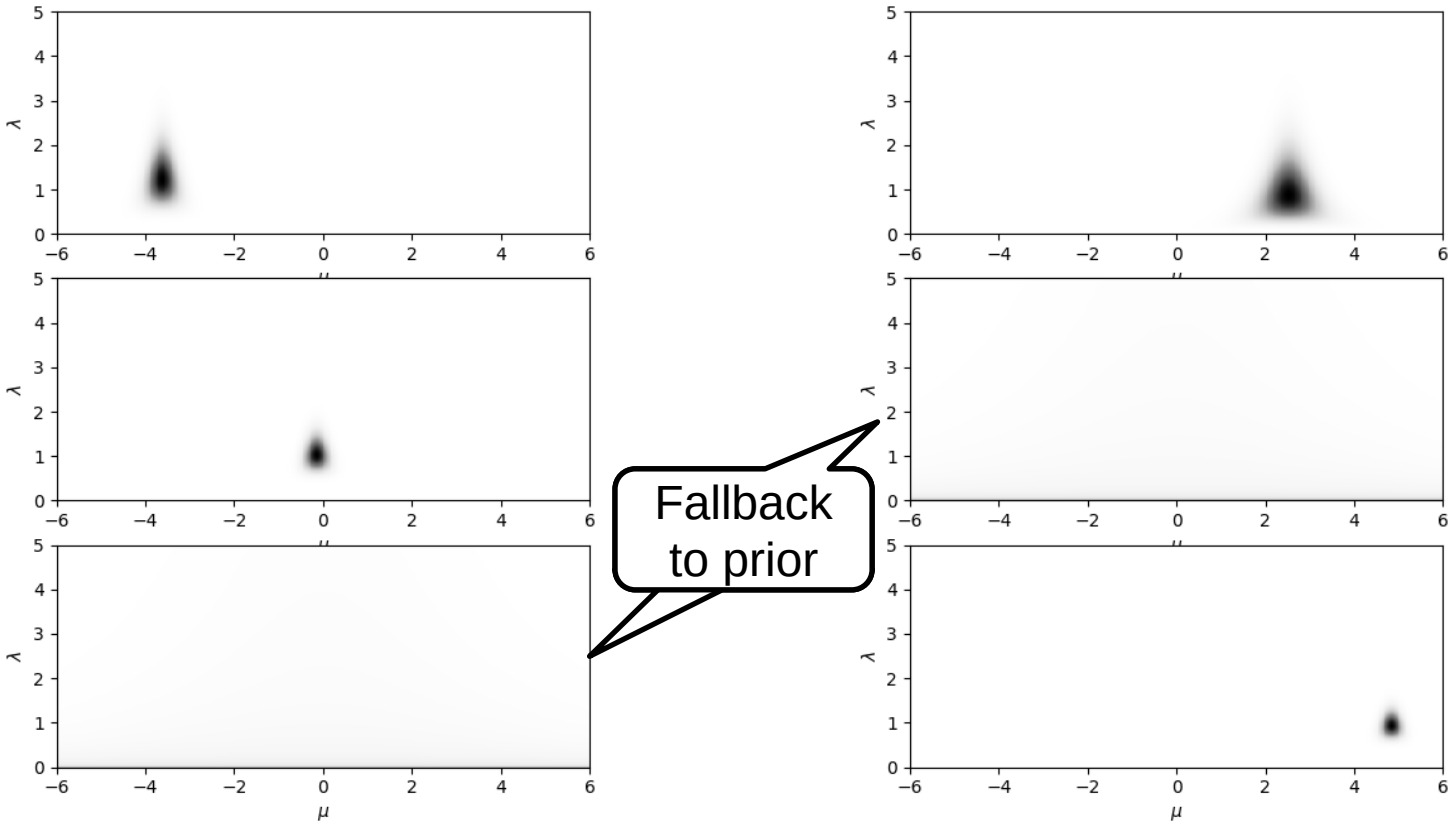
$$\alpha_c^* = \alpha_c + N_c$$

- Iterate until convergence

VB parameter posteriors

- Priors:
  - $p(\mu_c, \lambda_c), = \text{NormalGamma}(\mu_c, \lambda_c | 0.0,\ 0.05,\ 0.05,\ 0.05),\ c = 1..C$
  - $p(\boldsymbol{\pi}), = \text{Dir}(\boldsymbol{\pi} | [1, 1, 1, 1, 1, 1])$
- Posteriors:
  - $\boldsymbol{\alpha}_N = [17.1\ \ 8.3\ \ 32.2\ \ 1.0\ \ 1.0\ \ 46.4]$
  - $q(\mu_c, \lambda_c)$ for the 6 Gaussian components

Fallback to prior



Evaluating VB-GMM

- Lower bound $\mathcal{L}\big(q(\mathbf{Y})\big)$ can be evaluated to check for the convergence
  - Formula not shown here

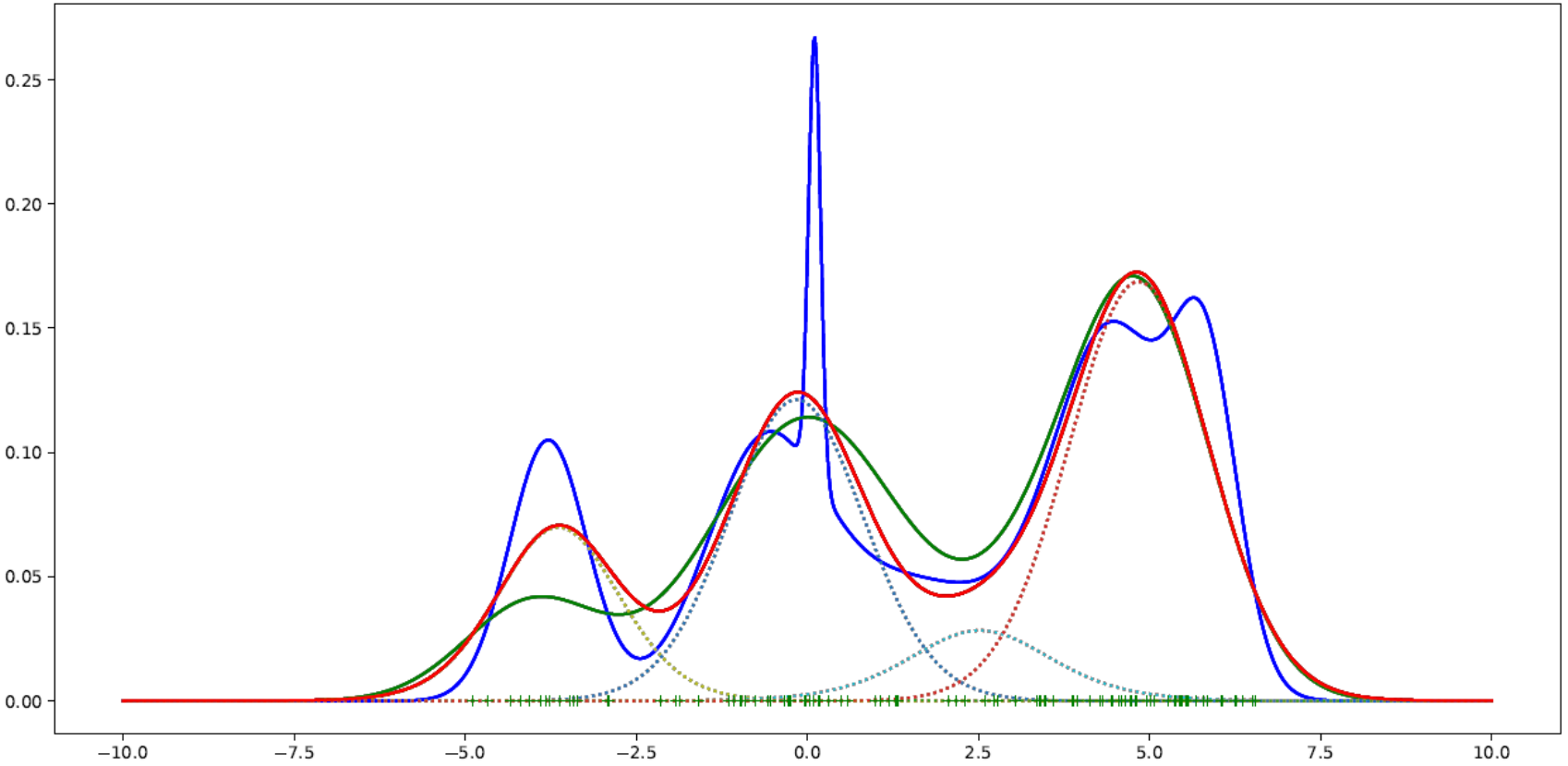
- Posterior predictive distribution is a mixture component specific posterior predictive of Student's t-distributions

$$p(x'|\mathbf{x}) = \sum_c \mathrm{St}\left(x' \middle| m_c^*, 2\mathrm{a}_\mathrm{c}^*, \frac{a_c^* \kappa_c^*}{b_c^*(\kappa_c^* + 1)}\right) \pi_c^*$$

where mixture weights are given by categorical posterior predictive:

$$\pi_c^* = \frac{\alpha_c^*}{\sum_c \alpha_c^*}$$

VB predictive vs. ML solution



- VB was initialized from ML solution – first update of $q(\mu_c, \lambda_c)$ and $q(\boldsymbol{\pi})$ uses the responsibilities from last ML iteration
- VB recovers from ML overfitting and more robust solution closer to the true distribution for generating the training data

Approximate inference (for Bayesian GMM)

- Variational Bayes
  - Approximate intractable $p(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\lambda}, \mathbf{z}|\mathbf{X})$ with tractable $q(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\lambda}, \mathbf{z}|\mathbf{X})$
  - Iteratively tune parameters of $q(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\lambda}, \boldsymbol{z})$ minimize $D_{KL}\left(q(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\lambda}, \boldsymbol{z})||p(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\lambda}, \boldsymbol{z}|\mathbf{X})\right)$
- Gibbs sampling
  - Instead of obtaining $p(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\lambda}, \boldsymbol{z}|\mathbf{X})$, we only generate samples from this distribution
  - Integrating over $p(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\lambda}, \boldsymbol{z}|\mathbf{X})$ (e.g. for predictive distribution) can be approximated with *empirical expectations*
- …

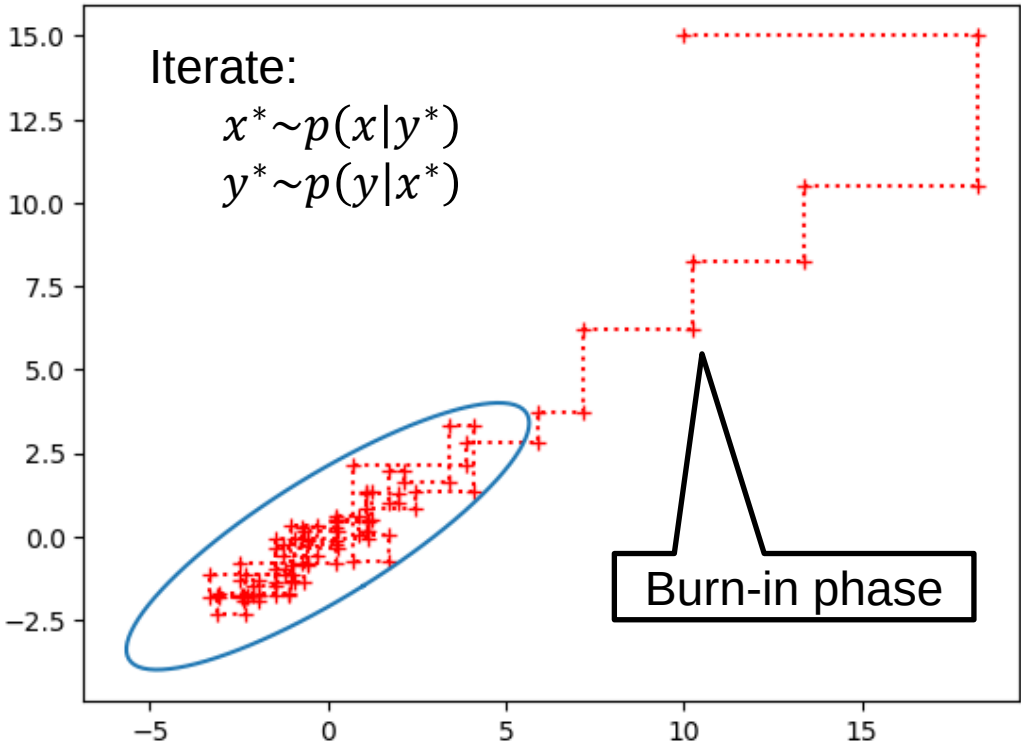
Gibbs Sampling

- Assume we cannot sample from the complex joint distribution $p(z_1, z_2)$ but it is possible to sample from the conditional distributions $p(z_1|z_2)$ and $p(z_2|z_1)$
  1. Initialize z_1^* to any value (i.e., chosen constant)
  2. Given current sample z_1^* generate $z_2^* \sim p(z_2|z_1)$
  3. Given current sample z_2^* generate $z_1^* \sim p(z_1|z_2)$
  4. Go to steps 2.

- In theory, after infinite number of iteration the final values z_1^*, z_2^* is a sample from $p(z_1, z_2)$
- Or, with increasing number of iterations, z_1^*, z_2^* converges to a valid sample from $p(z_1, z_2)$
- In practice, after several initial iterations (burn-in phase) take z_1^*, z_2^* from every N^{th} iteration and consider them samples from $p(z_1, z_2)$
  - Often $N = 1$ is used
  - Starting from a likely value of z_1^* requires less burn-in iterations
- This can be extended to any number variables
  - always sample one given current values for others
- Works for any random variables (discrete, continuous; scalars, vectors)

Gibbs sampling for 2D Gaussian

Of course, it is possible to efficiently and exactly sample directly from a 2D Gaussian distribution. We use this toy example only to demonstrate how Gibbs sampling works.



For 2D gaussian distribution

$$p\left(\begin{bmatrix} x \\ y \end{bmatrix}\right) = \mathcal{N}\left(\begin{bmatrix} x \\ y \end{bmatrix} \middle| \begin{bmatrix} \mu_x \\ \mu_y \end{bmatrix}, \begin{bmatrix} \Sigma_{xx} & \Sigma_{xy} \\ \Sigma_{yx} & \Sigma_{xx} \end{bmatrix}\right)$$

the conditional probability

$$p(x|y) = \mathcal{N}\left(x \middle| \mu_{x|y}, \Lambda_{xx}^{-1}\right)$$

where

$$\mu_{x|y} = \mu_x - \Lambda_{xx}^{-1}\Lambda_{xy}(y - \mu_y)$$

and

$$\begin{bmatrix} \Sigma_{xx} & \Sigma_{xy} \\ \Sigma_{yx} & \Sigma_{xx} \end{bmatrix}^{-1} = \begin{bmatrix} \Lambda_{xx} & \Lambda_{xy} \\ \Lambda_{yx} & \Lambda_{xx} \end{bmatrix}$$

Gibbs Sampling for Bayesian GMM

- Using sampled values of $\{\mu_c^*, \lambda_c^*\}$ and $\boldsymbol{\pi}^*$, generate new samples (hard assignments of observations to GMM components) from posterior over z_n^*
  - The distribution is just like the responsibilities from EM:

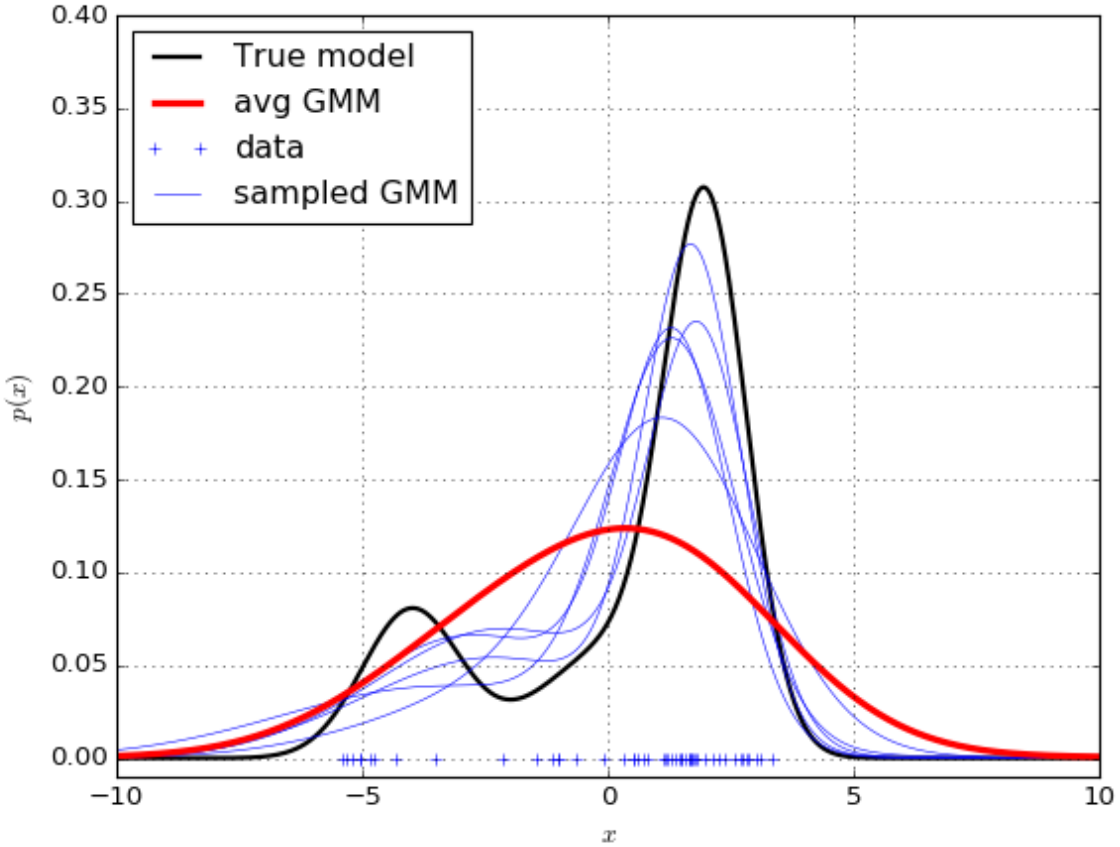
$$P(z_n = c|\mathbf{x}_n) = \frac{p(x_n|z_n = c)P(z_n = c)}{\sum_k p(x_n|k)P(k)} = \frac{\mathcal{N}(x_n|\mu_c^*, \lambda_c^{*-1})\pi_c^*}{\sum_k \mathcal{N}(\mathrm{x}_i|\mu_k^*, \lambda_k^{*-1})\pi_k^*}$$

- Using the sampled values z_n^* , for each component c, generate new samples of GMM parameters μ_c^*, λ_c^* from posteriors $p(\mu_c, \lambda_c|\mathbf{x}, \mathbf{z}^*)$
  - Estimate sufficient statistics $N_c^*, \bar{x}_c^*, s_c^*$ using the observations $\{\mathrm{x}_n : z_n = c\}$ (i.e., those hard assigned to the component c) and calculated the posterior as:

$$p(\mu_c, \lambda_c|\mathbf{x}) = \mathrm{NormalGamma}\left(\mu_c, \lambda_c \middle| \frac{\kappa m + N_c\bar{x}_c}{\kappa + N_c}, \kappa + N_c, a + \frac{N_c}{2}, b + \frac{N_c}{2}\left(s_c + \frac{\kappa(\bar{x}_c - m)^2}{\kappa + N_c}\right)\right)$$

- Sample $\boldsymbol{\pi}^*$ from posterior $p(\boldsymbol{\pi}|\mathbf{z}^*) = \mathrm{Dir}(\boldsymbol{\pi}|\boldsymbol{\alpha} + \mathbf{N}^*)$ where the vector of component occupation counts $\mathbf{N}^* = [N_1^*, N_2^*, \ldots, N_C^*]$ is given by $\mathbf{z}^*$

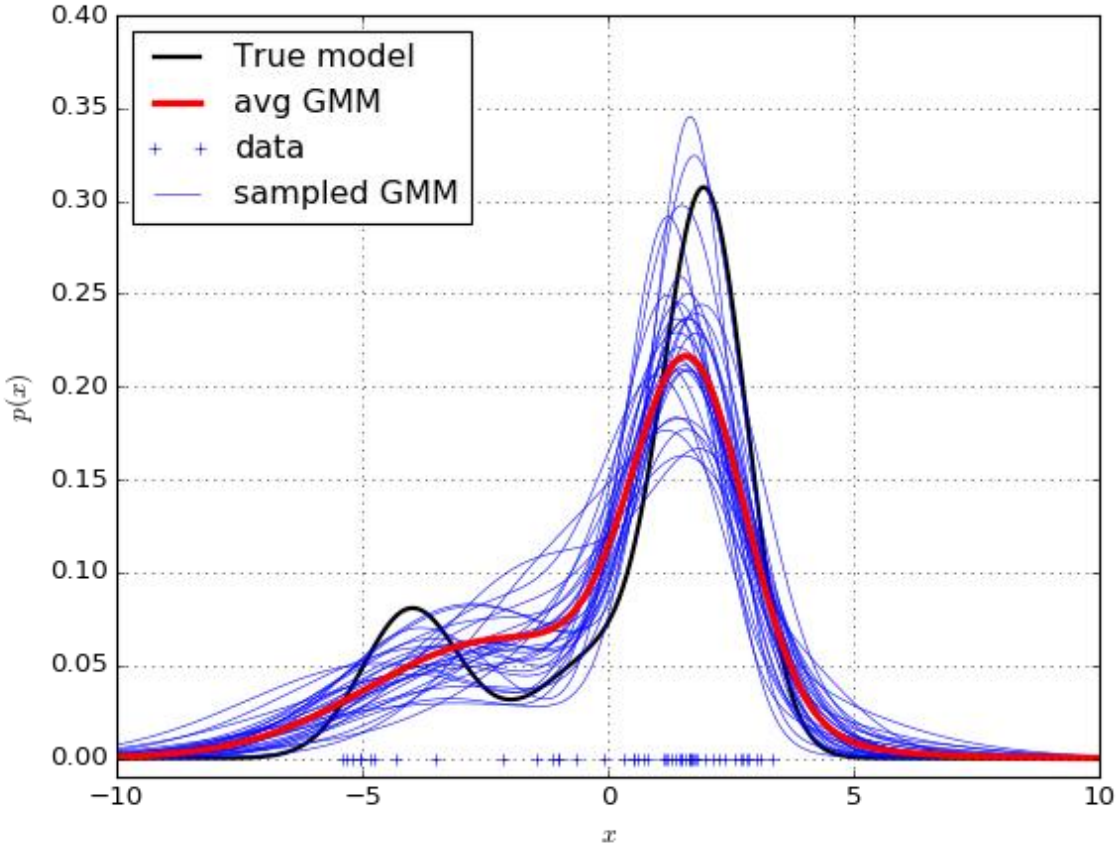
First 5-iterations of GS



Predictive distributions can be approximated by empirical expectations using the samples from the posterior distribution $\hat{\boldsymbol{\eta}}_l$:

$$p(x'|\mathbf{X}) = \int p(x'|\boldsymbol{\eta})p(\boldsymbol{\eta}|\mathbf{X})\mathrm{d}\,\boldsymbol{\eta} \approx \frac{1}{L}\sum_{l} p(x'|\hat{\boldsymbol{\eta}}_l)$$

First 30-iterations of GS



Predictive distributions can be approximated by empirical expectations using the samples from the posterior distribution $\hat{\boldsymbol{\eta}}_l$:

$$p(x'|\mathbf{X}) = \int p(x'|\boldsymbol{\eta})p(\boldsymbol{\eta}|\mathbf{X})\mathrm{d}\,\boldsymbol{\eta} \approx \frac{1}{L}\sum_{l} p(x'|\hat{\boldsymbol{\eta}}_l)$$

Collapsed GS for Bayesian GMM

- Sampling discrete latent variables like z_n is fine as they have limited number of possible values
- For continuous latent variables like $\boldsymbol{\pi}, \mu_c, \lambda_c$, however, we might need too many samples to get a reasonable representation of their posterior distributions (especially for multivariate higher dimensional variables).
- Collapsed Gibbs Sampling
  - Iterates over (and samples only from) a subset of the latent variables in the model (e.g. the discrete ones)
  - integrates (marginalizes) over the remaining (continuous) variables
- CBS for Bayesian GMM:

$$\text{for } i = 1..N$$

$$z_i^* \sim P(z_i | \mathbf{x}, \mathbf{z}_{\backslash i})$$

where

$\mathbf{z}_{\backslash \mathrm{i}}$ is $\mathbf{z}$ with z_i removed

$\mathbf{x}_{\backslash \mathrm{i}}$ is $\mathbf{x}$ with x_i removed

CGS for BGMM - $P(z_i|\mathbf{z}_{\backslash i})$

- How do we obtain $p(z_i|\mathbf{x}, \mathbf{z}_{\backslash i})$?
- Let's first introduce some useful distributions
- Posterior distribution of weights $\boldsymbol{\pi}$ given $\mathbf{z}_{\backslash i}$ (or corresponding vector of component occupation counts $\mathbf{N}_{\backslash i}$)

$$p(\boldsymbol{\pi}|\mathbf{z}_{\backslash i}) \propto \prod_{n \neq i} P(z_n|\boldsymbol{\pi}) p(\boldsymbol{\pi}) = \prod_{n \neq i} \mathrm{Cat}(z_n|\boldsymbol{\pi}) \mathrm{Dir}(\boldsymbol{\pi}|\boldsymbol{\alpha}) \propto \mathrm{Dir}(\boldsymbol{\pi}|\boldsymbol{\alpha} + \mathbf{N}_{\backslash i})$$

- Posterior predictive distribution for z_i given $\mathbf{z}_{\backslash i}$

$$P(z_i|\mathbf{z}_{\backslash i}) = \int P(z_i|\boldsymbol{\pi}) p(\boldsymbol{\pi}|\mathbf{z}_{\backslash i}) \, \mathrm{d}\boldsymbol{\pi} = \int \mathrm{Cat}(z_i|\boldsymbol{\pi}) \mathrm{Dir}(\boldsymbol{\pi}|\boldsymbol{\alpha} + \mathbf{N}_{\backslash i}) \, \mathrm{d}\boldsymbol{\pi}$$

$$= \mathrm{Cat}\left(z_i \middle| \frac{\boldsymbol{\alpha} + \mathbf{N}_{\backslash i}}{\sum_c \alpha_c + N - 1}\right)$$

CGS for BGMM - $p\left(x_i \middle| z_i, \mathbf{x}_{\backslash \mathrm{i}}, \mathbf{z}_{\backslash \mathrm{i}}\right)$

- Let $S_{c\backslash i}$ define the subset of observations assigned by $\mathbf{z}_{\backslash \mathrm{i}}$ to component c
- Posterior distribution of μ_c, λ_c given $\mathbf{x}_{\backslash \mathrm{i}}, \mathbf{z}_{\backslash \mathrm{i}}$ is estimated in the usual way using only the observations $S_{c\backslash i}$

$$p(\mu_c, \lambda_c | \mathbf{x}_{\backslash i}, \mathbf{z}_{\backslash i}) \propto \prod_{n \in S_{c\backslash i}} p(x_n | z_n, \boldsymbol{\mu}, \boldsymbol{\lambda}) p(\boldsymbol{\mu}, \boldsymbol{\lambda})$$

$$= \prod_{n \in S_{c\backslash i}} \mathcal{N}(x_n | \mu_c, \lambda_c^{-1}) \mathrm{NormalGamma}(\mu_c, \lambda_c | m, \kappa, a, b)$$

$$\propto \prod_{n \in S_{c\backslash i}} \mathrm{NormalGamma}(\mu_k, \lambda_k | m^*_{c\backslash i}, \kappa^*_{c\backslash i}, a^*_{c\backslash i}, b^*_{c\backslash i})$$

- Posterior predictive distrib. of x_i for component c given observations $S_{c\backslash i}$

$$p(x_i | z_i = c, \mathbf{x}_{\backslash i}, \mathbf{z}_{\backslash i}) = \int p(x_i | z_i = c, \mu_c, \lambda_c) p(\mu_c, \lambda_c | \mathbf{x}_{\backslash i}, \mathbf{x}_{\backslash i}) \, \mathrm{d}\mu_c, \, \mathrm{d}\lambda_c$$

$$= \int \mathcal{N}(x_i | \mu_c, \lambda_c^{-1}) \mathrm{NormalGamma}(\mu_c, \lambda_c | m^*_{c\backslash i}, \kappa^*_{c\backslash i}, a^*_{c\backslash i}, b^*_{c\backslash i}) \, \mathrm{d}\mu_c \, \mathrm{d}\lambda_c$$

$$= \mathrm{St}\left(x_i | m^*_{c\backslash i}, 2a^*_{c\backslash i}, \frac{a^*_{c\backslash i} \kappa^*_{c\backslash i}}{b^*_{c\backslash i}(\kappa^*_{c\backslash i} + 1)}\right)$$

CGS for BGMM - $p(x_i|\mathbf{x}, \mathbf{z}_{\backslash \mathbf{i}})$

- Finally, using Bayes rule

$$P(z_i|\mathbf{x}, \mathbf{z}_{\backslash i}) = P(z_i|x_i, \mathbf{x}_{\backslash i}, \mathbf{z}_{\backslash i}) = \frac{p(x_i|z_i, \mathbf{x}_{\backslash i}, \mathbf{z}_{\backslash i})P(z_i|\mathbf{x}_{\backslash i}, \mathbf{z}_{\backslash i})}{\sum_c p(x_i|z_i = c, \mathbf{x}_{\backslash i}, \mathbf{z}_{\backslash i})P(z_i = c|\mathbf{z}_{\backslash i})}$$

- The Collapsed Gibbs sampling iterations

$$\text{for } i = 1..N$$
$$z_i^* \sim P(z_i|\mathbf{x}, \mathbf{z}_{\backslash i})$$

gives us samples from $\mathbf{z}^* \sim p(\mathbf{z}|\mathbf{x})$. What can we do with that?

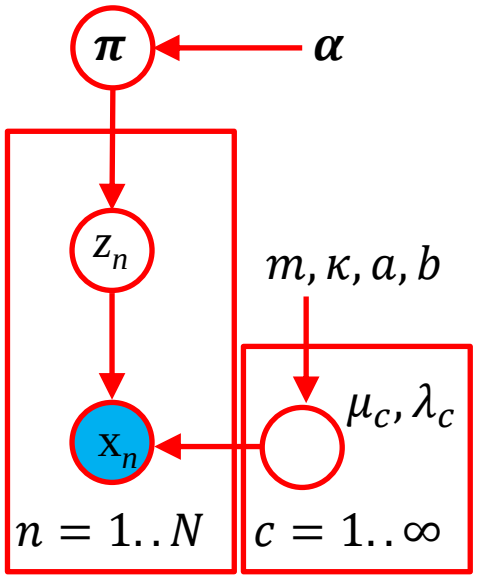
- GMM posterior predictive distribution for new x' given $\mathbf{x}$ and (sampled) $\mathbf{z}$

$$p(x'|\mathbf{x}, \mathbf{z}) = \sum_c p(x'|z = c, \mathbf{x}, \mathbf{z})P(z = c|\mathbf{z})$$

- Full predictive distribution can be approximated using the samples $\mathbf{z}_l^*$ as

$$p(x|\mathbf{x}) = \sum_{\mathbf{z}} p(x'|\mathbf{x}, \mathbf{z})p(\mathbf{z}|\mathbf{x}) \approx \frac{1}{L}\sum_l p(x'|\mathbf{x}, \mathbf{z}_l^*)$$

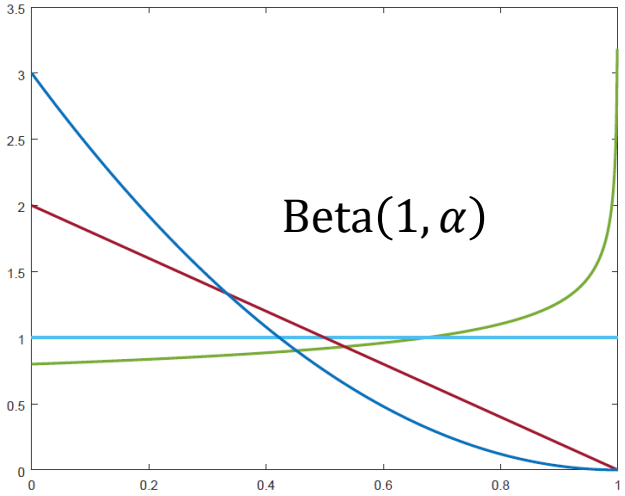
Infinite Bayesian GMM



- Let's consider Bayesian GMM with an infinite number of Gaussian components $c = 1..\infty$
- The priors for μ_c, λ_c for Gaussian component $c = 1 \ldots \infty$ can be defined as before:
  - $p(\mu_c, \lambda_c) \sim \mathrm{NormalGamma}(\mu_c, \lambda_c | \, m, \kappa, a, \mathrm{b})$
- However, we need an infinite number of mixture weights $\boldsymbol{\pi} = [\pi_1, \pi_2, \ldots\ ]$ so that $\sum_{c=1}^{\infty} \pi_c = 1$
- We also need a suitable prior distribution for $\boldsymbol{\pi}$

Stick breaking process - GEM

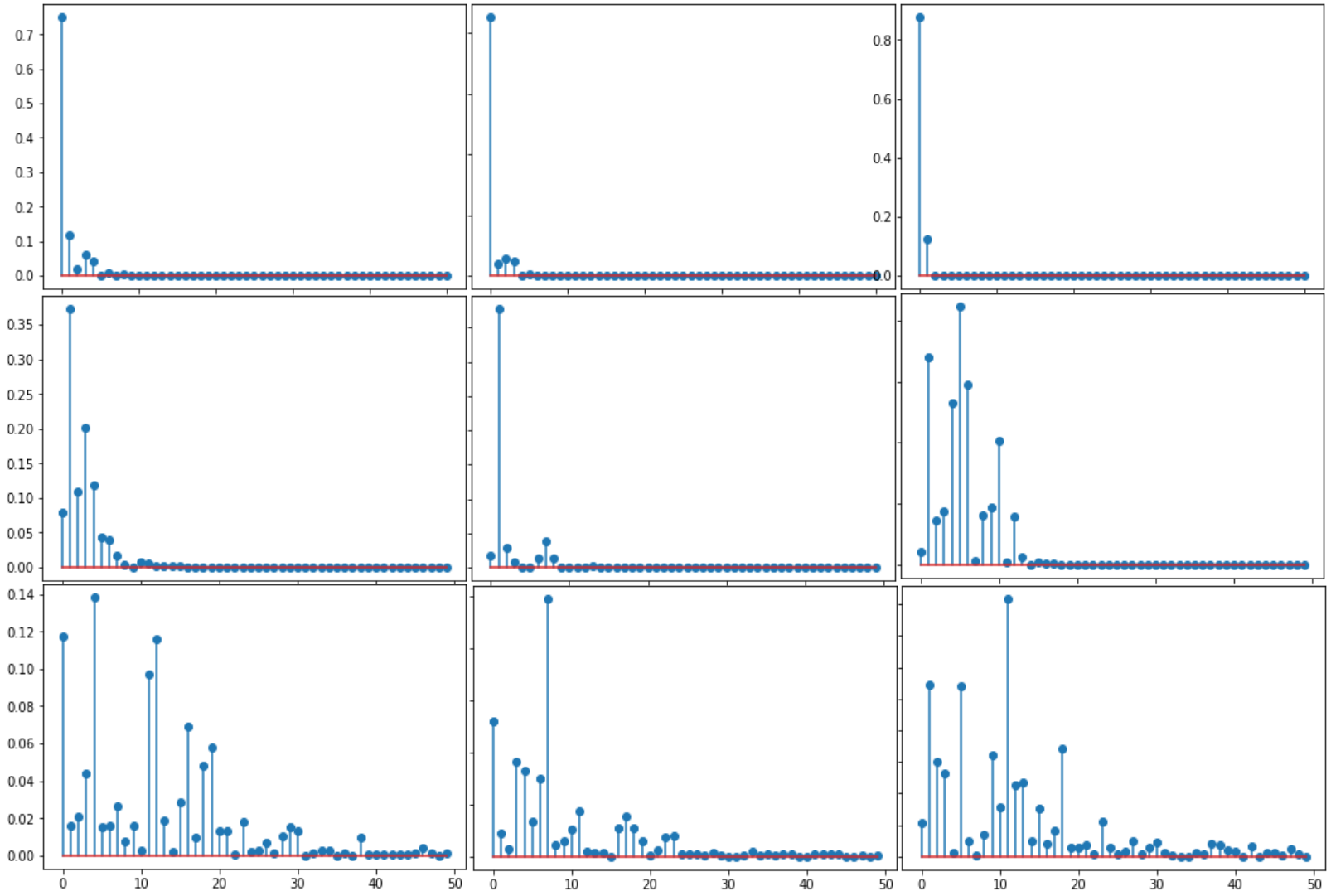
- for $c = 1,2,\ldots,\infty$
  $v_c \sim \mathrm{Beta}(1,\alpha)$
  $\pi_c = v_c \prod_{k=1}^{c-1} (1 - v_k)$



- Take a unit length stick
  For $c = 1,2,\ldots,\infty$
  - Generate v_c in range $(0,1)$ from $\mathrm{Beta}(1,\alpha)$
  - Break the stick into two pieces with proportions $v_c : 1 - v_c$
  - The length of the first piece corresponds to π_c
  - The second piece is the stick to be broken in further iterations

- The resulting infinite dimensional vector of weights is a sample from the stick breaking process $\boldsymbol{\pi} \sim \mathrm{GEM}(\alpha)$ (Griffiths, Engen and McCloskey)
- $\mathrm{GEM}(\alpha)$ can be used as a prior for infinite number of component weights
- With small concentration parameter α, only few weights will be non-negligable

Samples from GEM

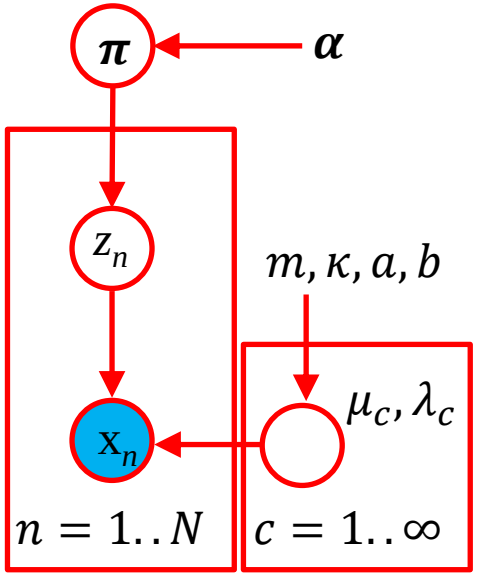


$\alpha = 0.5$

$\alpha = 2$

$\alpha = 10$

Infinite Bayesian GMM



- We assume that the observed data were generated as follows:
  - $\boldsymbol{\pi} = [\pi_1, \pi_2, \ldots] \sim \mathrm{GEM}(\alpha)$
  - For Gaussian component $c = 1 \ldots \infty$
    - $\mu_c, \lambda_c \sim \mathrm{NormalGamma}(\mu_c, \lambda_c | m, \kappa, a, \mathrm{b})$
  - For each observation $i = 1 \ldots N$
    - $z_n \sim \mathrm{P}(z_n | \boldsymbol{\pi}) = \mathrm{Cat}(z_n | \boldsymbol{\pi})$
    - $x_n \sim p(x_n | z_n, \boldsymbol{\mu}, \boldsymbol{\lambda}) = \mathcal{N}\left(x_n | \mu_{z_n}, \ \lambda_{z_n}^{-1}\right)$
- Obviously, the observed data can be generated from at most N Gaussian components.
- Again, the task is to infer the posterior distribution of parameters $\mathrm{p}(\boldsymbol{\pi}, \mu_1, \lambda_1, \mu_2, \lambda_2 \ldots | \mathbf{x})$ given some observed data $\mathbf{x} = [x_1, x_2, \ldots, x_N]$

CGS for infinite Bayesian GMM

- We can use the same Collapsed Gibbs sampling iterations that we used in the case of the BGMM with fixed number of Gaussian

for $i = 1..N$

$$z_i^* \sim P(z_i|\mathbf{x}, \mathbf{z}_{\backslash i})$$

where again $$P(z_i|\mathbf{x}, \mathbf{z}_{\backslash i}) = \frac{p(x_i|z_i, \mathbf{x}_{\backslash i}, \mathbf{z}_{\backslash i})P(z_i|\mathbf{x}_{\backslash i}, \mathbf{z}_{\backslash i})}{\sum_c p(x_i|z_i = c, \mathbf{x}_{\backslash i}, \mathbf{z}_{\backslash i})P(z_i = c|\mathbf{z}_{\backslash i})}$$

and the component posterior predictive

$$p(x_i|z_i = c, \mathbf{x}_{\backslash i}, \mathbf{z}_{\backslash i}) = \mathrm{St}\left(x_i|m^*_{c\backslash i}, 2a^*_{c\backslash i}, \frac{a^*_{c\backslash i}\kappa^*_{c\backslash i}}{b^*_{c\backslash i}(\kappa^*_{c\backslash i} + 1)}\right)$$

- The only difference will be in $P(z_i|\mathbf{z}_{\backslash \mathrm{i}})$, which is evaluated using Chinese Restaurant Process (CRP)

Chinese Restaurant Process

- Let the prior on the infinite weight vector be $p(\boldsymbol{\pi}) = \mathrm{GEM}(\pi|\alpha)$
- Let $z_n,\ \ n = 1..N$ be samples generated from an (unknown) "infinite categorical distribution" $\mathrm{Cat}(z_n|\boldsymbol{\pi})$
- The posterior $p(\boldsymbol{\pi}|\mathbf{z}) \propto \ \prod_n p(z_n|\boldsymbol{\pi})\, p(\boldsymbol{\pi})$ is intractable
  - We cannot even easily sample from it as the sample would be infinite vector of weights
- However, the predictive posterior $P(z'|\mathbf{z}) = \int P(z'|\boldsymbol{\pi}) p(\boldsymbol{\pi}|\mathbf{z}) \mathrm{d}\boldsymbol{\pi}$ can be evaluated as

$$P(z' = c|\mathbf{z}) = \frac{N_c}{\alpha + N}$$

$$P(z' = C + 1|\mathbf{z}) = \frac{\alpha}{\alpha + N}$$

where N_c is the number of observations assigned by $\mathbf{z}$ to category c and $C + 1$ is a new so far not seen category.

Posterior predictive: Dirichlet vs GEM prior

- Posterior predictive for Categorical distribution and Dirichlet prior (with single concentration parameter) converges to CRP as the number of categories increases
  - Prior: $\mathrm{Dir}(\boldsymbol{\pi}|\alpha)$
  - Observation distribution: $\mathrm{Cat}(z|\boldsymbol{\pi})$
  - Posterior $\mathrm{Dir}(\boldsymbol{\pi}|\mathbf{m}+\boldsymbol{\alpha})$, where $\mathbf{m} = [N_1, N_2, \ldots, N_{\mathrm{C}}]$
  - Posterior predictive $p(\mathrm{z}'|\mathbf{z}) = \int \mathrm{Cat}(\mathrm{z}'|\boldsymbol{\pi})\mathrm{Dir}(\boldsymbol{\pi}|\boldsymbol{\alpha}_N)\,\mathrm{d}\boldsymbol{\pi} = \mathrm{Cat}\left(\mathrm{z}'\middle|\frac{\boldsymbol{\alpha}+\mathbf{m}}{\sum_c \alpha_c + N_c}\right)$
- For single concentration parameter $\alpha_c = \alpha = \frac{\gamma}{C}$

$$p(\mathrm{z}' = k|\mathbf{z}) = \frac{\alpha + N_k}{C\alpha + N} = \frac{\frac{\gamma}{C} + N_k}{\gamma + N}$$

$$p(\mathrm{z}' = k|\mathbf{z}) = \frac{N_k}{\gamma + N} \text{ for } \mathrm{C} \to \infty$$

- γ is number of prior observations that we keep constant with increasing C ➔ $\alpha_c$gets smaller with increasing C

Chinese Restaurant Process

- Imagine Chinese Restaurant with an infinite number of tables, each with infinite capacity
- The first customer sits at the first table
- Every new customer:
  - Joins already occupied table with probability proportional to the number of customers sitting at that table

$$P(z' = c|\mathbf{z}) = \frac{N_c}{\alpha + N}$$

  - or starts a new table with probability proportional to concentrarion parameter α

$$P(z' = C + 1|\mathbf{z}) = \frac{\alpha}{\alpha + N}$$

Dirichlet Process

We have defined Infinite BGMM as (for simplicity assuming the same σ for all Gaussian component variances σ and conjugate prior $p(\mu_c) = \mathcal{N}(\mu_c|\mu_0, \sigma_0)$):

$$
\begin{aligned}
&\boldsymbol{\pi} = [\pi_1, \pi_2, \dots] \sim \mathrm{GEM}(\alpha) \\
&\mu_c \sim \mathcal{N}(\mu_c|\mu_0, \sigma_0), && c = 1..\infty \\
&z_i \sim \boldsymbol{\pi}, && i = 1..N \\
&x_i \sim \mathcal{N}\left(x_i\middle|\mu_{z_i}, \sigma\right), && i = 1..N
\end{aligned}
$$

Alternative definition using $\delta_\mu(\tilde{\mu}) = \begin{cases} 1, & \mu = \tilde{\mu} \\ 0, & \mu \neq \tilde{\mu} \end{cases}$

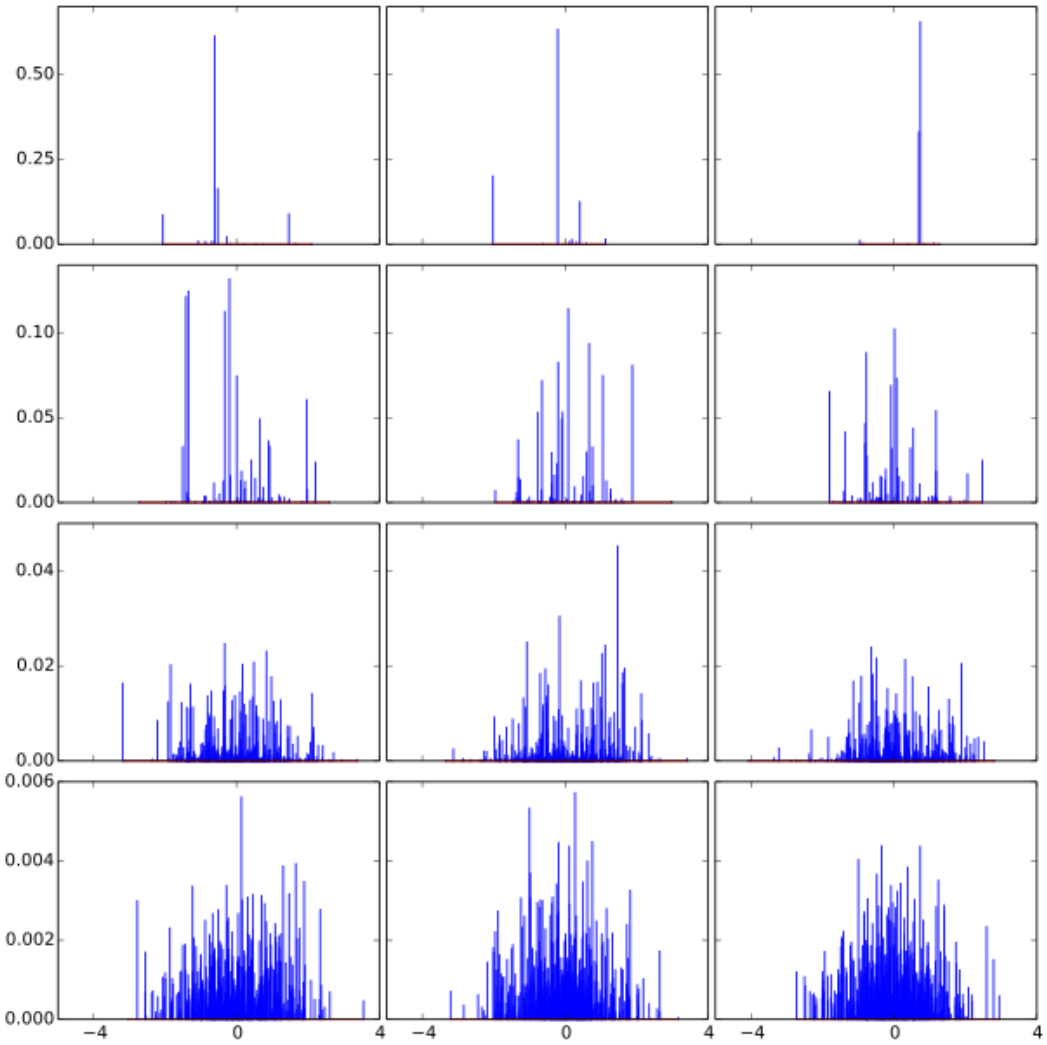
$$
\begin{aligned}
&\boldsymbol{\pi} = [\pi_1, \pi_2, \dots] \sim \mathrm{GEM}(\alpha) \\
&\mu_c \sim \mathcal{N}(\mu_c|\mu_0, \sigma_0), && c = 1..\infty \\
&\tilde{\mu}_i \sim G = \sum_{c=1}^{\infty} \pi_c \delta_{\mu_c}(\tilde{\mu}_i), && i = 1..N \\
&x_i \sim \mathcal{N}(x_i|\tilde{\mu}_i), && i = 1..N
\end{aligned}
$$

or using Dirichlet Process with base distribution $H = \mathcal{N}(\mu_0, \sigma_0)$ and concentration parameter α

$$
\begin{aligned}
&G \sim DP(H, \alpha) \\
&\tilde{\mu}_i \sim G, && i = 1..N \\
&x_i \sim \mathcal{N}(x_i|\tilde{\mu}_i), && i = 1..N
\end{aligned}
$$

Dirichlet process

Samples $\mathrm{G} \sim \mathrm{DP}(\mathcal{N}(0,1), \alpha)$



$\alpha = 1$

$\alpha = 10$

$\alpha = 100$

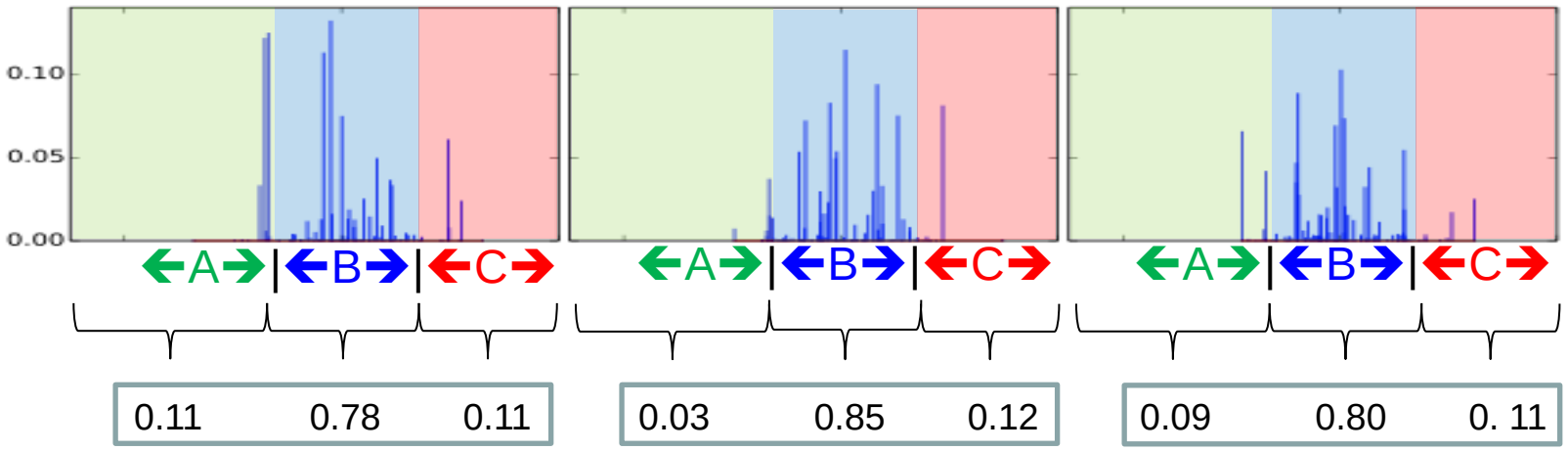
$\alpha = 1000$

G is discrete distribution with continuous support

$\mathrm{DP}(\mathcal{N}(0,1), \alpha)$ is distribution over discrete distributions with continuous support

Dirichlet process

Samples $\mathrm{G} \sim \mathrm{DP}(\mathcal{N}(0,1), \alpha)$



$\alpha = 10$

Let's decide on arbitrary partitioning of the support (regions A, B, C, ...). Now, for each sample from the DP, let's integrate the probability mass in each partition. The resulting vectors of probabilities are samples from Dirichlet distribution.

When Categorical distribution is used as the base distribution DP degrades to Dirichlet distribution

$$\mathrm{DP}(\mathrm{Cat}(\boldsymbol{\pi}), \alpha) = \mathrm{Dir}(\alpha\boldsymbol{\pi})$$

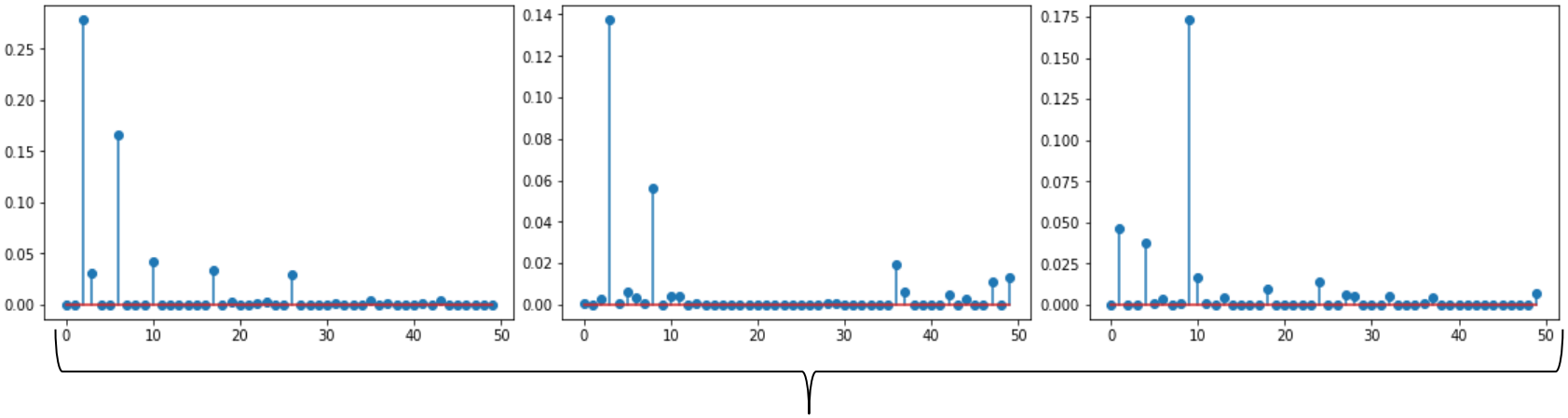
Pitman-Yor process

- Generalization of Dirichlet Process based on stick breaking process $\mathrm{GEM}(\alpha, d)$ with two parameters
  - discount parameter $0 \leq d < 1$
  - concentration parameter $\alpha > -d$

$$\text{for } c = 1,2,\ldots,\infty$$
$$v_c \sim \mathrm{Beta}(1-\mathrm{d}, \alpha + cd)$$
$$\pi_c = v_c \prod_{k=1}^{c-1} (1 - v_k)$$

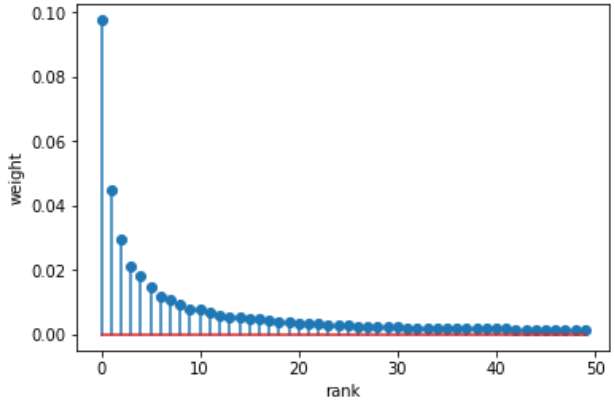
- For $d = 0$, PY process degrades to DP
- With d close to one, distribution of weights has long tail following Zipf's law: first weights is (in average) twice the second one, tree times the third one, ...
  - In any language, the most frequent word is about 2x more frequent than the second most frequent and 3x more frequent than the third most frequent and …
  - In English: 7% “THE”, 3.5% “OF”, 2.8%, “AND”, …
  - Largest city in a country has about twice the population of the second largest …
  - Same for: corporation sizes, income rankings, ranks of number of people watching the same TV channel

Samples from $\mathrm{GEM}(\alpha = 0, d = 0.9)$

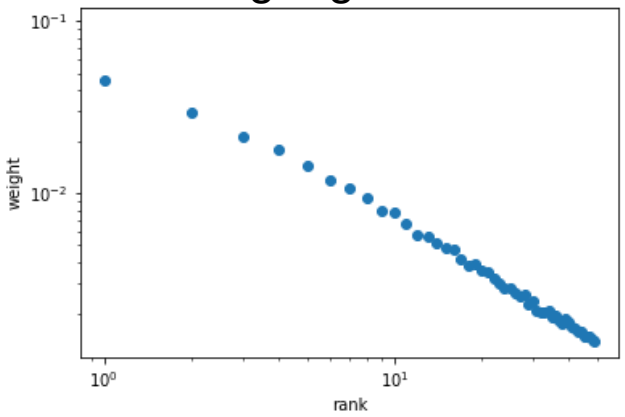


average over many samples

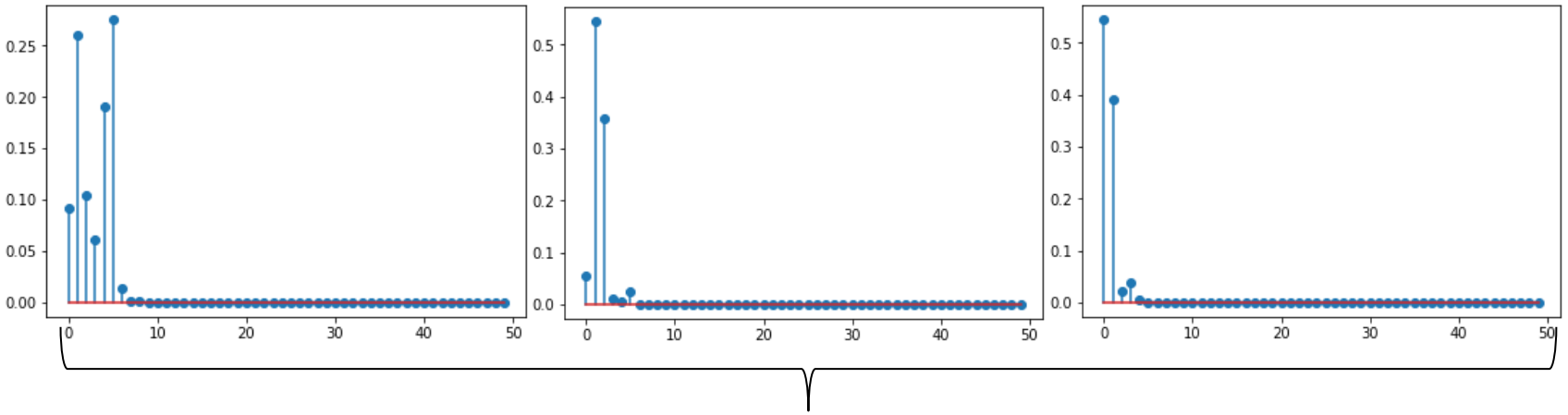
linear scale



log-log scale

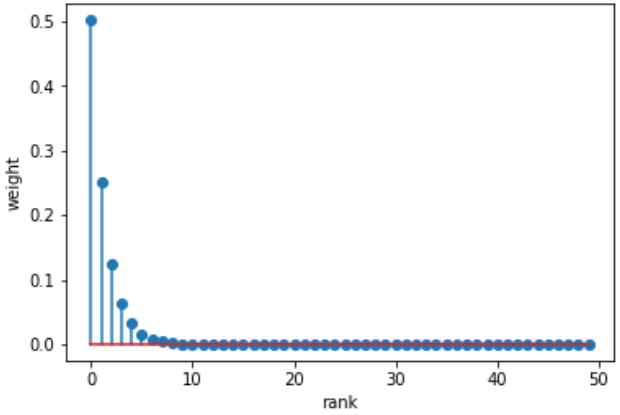


Samples from $\mathrm{GEM}(\alpha = 1, d = 0)$

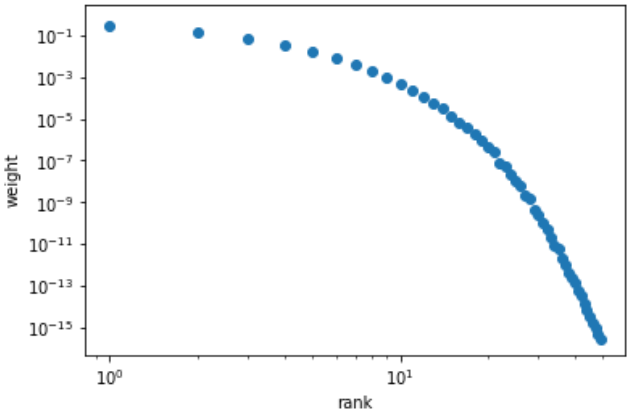


average over many samples

linear scale



log-log scale



CRP for $\mathrm{GEM}(\alpha, d)$

- Imagine Chinese Restaurant with an infinite number of tables, each with infinite capacity
- The first customer sits at the first table
- Every new customer:
  - Joins one of C already occupied table with probability proportional to the number of customers sitting at that table minus discount d

$$P(z' = c|\mathbf{z}) = \frac{N_c - d}{\alpha + N}$$

  - or starts a new $C + 1$ table with probability

$$P(z' = C + 1|\mathbf{z}) = \frac{\alpha + Cd}{\alpha + N}$$

Pitman-Yor Process

Sample from Pitman-Yor Process with base distribution H and concentration parameter α and discount d

$$G \sim \mathrm{PY}(H, \alpha, d)$$

can be obtained as

$$\boldsymbol{\pi} \sim \mathrm{GEM}(\alpha, d)$$
$$\mu_c \sim H, \qquad c = 1..\infty$$

$$G = \sum_{c=1}^{\infty} \pi_c \delta_{\mu_c}$$

- In CRP analogy, μ_c is a meal served at table c.
- For $H = \mathrm{NormalGamma}$ (or Normal) G again corresponds to (parameters of) infinite Gaussian Mixture model and $\mathrm{PY}(H, \alpha, d)$ can be seen and prior for GMM parameters.
- However, for PY process, it is interesting to consider $H = \mathrm{Cat}$

PY Process for $H = \mathrm{Cat}(\boldsymbol{r})$

- $\boldsymbol{r} = [r_1, r_2, \ldots, r_K]$ are probabilities of K categories
- μ_c corresponds to category associated with cluster c
- $\delta_{\mu_c}$is distribution where $\mathrm{P}(\mu_c) = 1$ and $\mathrm{P}(\mu \neq \mu_c) = 0$
- When sampling from G, we pick δ_{μ_c} with probability π_c and generated corresponding category μ_c
- $\Rightarrow G$ is (finite) Categorical distribution where

$$P(\mu_k) = \sum_{c:\mu_c=\mu_\mathrm{k}} \pi_c$$

PY Process for $H = \mathrm{Cat}(\boldsymbol{r})$

$$G \sim \mathrm{PY}(\mathrm{Cat}(\boldsymbol{r}), \alpha, d)$$

- G is Categorical Distribution
- We have seen that

$$\mathrm{PY}(\mathrm{Cat}(\boldsymbol{r}), \alpha, d = 0) = \mathrm{DP}(\mathrm{Cat}(\boldsymbol{r}), \alpha) = \mathrm{Dir}(\alpha \boldsymbol{r})$$

- But for $d \neq 0$

$$\mathrm{PY}(\mathrm{Cat}(\boldsymbol{r}), \alpha, d) \neq \mathrm{Dir}(\,)$$

- Prior for (finite) Categorical distributions imposing Zipf's law
- Useful for modeling different phenomena e.g. in Natural Language Processing (NLP)

$\mathrm{PY}(\mathrm{Cat}(\boldsymbol{r}), \alpha, d)$ prior

- We assume (*unigram*) generative model, where sequence of categories (e.g. words $\boldsymbol{x} = [x_1, x_2, \ldots x_N]$ are generated as

$$
\begin{aligned}
&G \sim \mathrm{PY}(\mathrm{Cat}(\boldsymbol{r}), \alpha, d) \\
&\text{for } i = 1..N \\
&\quad x_i \sim G
\end{aligned}
$$

or equivalently

$$
\begin{aligned}
&\boldsymbol{\pi} = [\pi_1, \pi_2, \ldots] \sim \mathrm{GEM}(\alpha, d) \\
&\mu_c \sim \mathrm{Cat}(\boldsymbol{r}), \qquad c = 1..\infty \\
&z_i \sim \boldsymbol{\pi}, \qquad i = 1..N \\
&x_i = \mu_{z_i}, \qquad i = 1..N
\end{aligned}
$$

CGS with $\mathrm{PY}(\mathrm{Cat}(\boldsymbol{r}), \alpha, d)$ prior

- We use Collapsed Gibbs sampling (similar to infinite BGMM)

  for $i = 1..N$

  $$z_i^* \sim P(z_i | \mathbf{x}, \mathbf{z}_{\backslash i})$$

- $$P(z_i | \mathbf{x}, \mathbf{z}_{\backslash i}) = \frac{p(x_i | z_i, \mathbf{x}_{\backslash i}, \mathbf{z}_{\backslash i}) P(z_i | \mathbf{x}_{\backslash i}, \mathbf{z}_{\backslash i})}{\sum_c p(x_i | z_i = c, \mathbf{x}_{\backslash i}, \mathbf{z}_{\backslash i}) P(z_i = c | \mathbf{z}_{\backslash i})}$$

- $P(z_i | \mathbf{z}_{\backslash \mathrm{i}})$ is evaluated using Chinese Restaurant Process (CRP)

  $$P(z_i = c | \mathbf{z}_{\backslash \mathrm{i}}) = \frac{N_{\backslash \mathrm{i}}^c - d}{\alpha + N - 1} \qquad P(z_i = C + 1 | \mathbf{z}_{\backslash \mathrm{i}}) = \frac{\alpha + Cd}{\alpha + N - 1}$$

- $P(x_i | z_i = C + 1, \mathbf{x}_{\backslash \mathrm{i}}, \mathbf{z}_{\backslash \mathrm{i}}) = \mathrm{Cat}(x_i | \boldsymbol{r})$ when starting new table.
- Each table c, serves only one "meal" μ_c, which is shared by all customers $z_j = c$ sitting at that table ⇒ For already occupied table c, we can look at any customer sitting at the table (i.e. $z_j \in \mathbf{z}_{\backslash \mathrm{i}} \wedge z_j = c$) and his meal $x_j = \mu_c$ and we know that a new customer $z_i = c$ joining the same table will eat the same meal with probability one
- $P(x_i = \mu_c | z_i = c, \mathbf{x}_{\backslash \mathrm{i}}, \mathbf{z}_{\backslash \mathrm{i}}) = 1$ and $P(x_i \neq \mu_c | z_i = c, \mathbf{x}_{\backslash \mathrm{i}}, \mathbf{z}_{\backslash \mathrm{i}}) = 0$

CGS with $\mathrm{PY}(\mathrm{Cat}(\boldsymbol{r}), \alpha, d)$ prior-II

- Approximate posterior predictive

$$
\begin{aligned}
&\mathrm{P}(x|\mathbf{x}) \\
&\approx \frac{1}{L}\sum_{l} P(x|\mathbf{x}, \mathbf{z}_{\mathrm{l}}^{*}) = \frac{1}{L}\sum_{l}\sum_{c=1}^{C_l+1} P(x|z=c, \mathbf{x}, \mathbf{z}_{l}^{*}) P(z=c|\mathbf{z}_{\mathrm{l}}^{*}) \\
&= \frac{1}{L}\sum_{l}\left(\sum_{c:\mu_c=x} P(z=c|\mathbf{z}_{\mathrm{l}}^{*}) + \mathrm{Cat}(x|\boldsymbol{r}) P(z=C_l+1|\mathbf{z}_{\mathrm{l}}^{*})\right) \\
&= \frac{1}{L}\sum_{l}\left(\sum_{c:\mu_c=x} \frac{N_l^c - d}{\alpha + N} + \mathrm{Cat}(x|\boldsymbol{r})\frac{\alpha + C_l d}{\alpha + N}\right)
\end{aligned}
$$

Language Modeling

- In NLP or speech processing, we often need to model distribution of word sequences
  $P(x_1, x_2, \dots, x_N) = P(x_1)P(x_2|x_1)P(x_3|x_1, x_2) \dots P(x_N|x_1, x_2, \dots, x_{N-1})$
- This can be approximated by N-gram language model (LM)
  - uni-gram:
    $P(x_1, x_2, \dots, x_N) \approx P(x_1)P(x_2)P(x_3) \dots P(x_N)$
  - bi-gram:
    $P(x_1, x_2, \dots, x_N) \approx P(x_1)P(x_2|x_1)P(x_3|x_2) \dots P(x_N|x_{N-1})$
  - ML estimation of $P(x)$ or $P(x|h)$ is not robust – not seeing certain word or word pair in training text $\mathbf{x}$ does not mean that it has zero probability in new data.
  - Smoothing techniques (Good–Turing discounting, Kneser–Ney smoothing, …) are typically used to get better estimates.
- Let's use Bayesian approach …

Bayesian Language Modeling

- For unigram
  - We assume that individual words are i.i.d. from $x_n \sim \mathrm{Cat}(x_n|\boldsymbol{r}_1)$
  - Let's use $\boldsymbol{r}_1 \sim \mathrm{PY}(H_0, \alpha, d)$ as a prior for the parameters $\boldsymbol{r}_1$
  - For H_0, will be flat categorical distribution $\mathrm{Cat}(x|\boldsymbol{r}_0) = \frac{1}{K}$
  - We can use the CGS inference described before
  - We can use the approximate posterior predictive $\mathrm{P}(x|\mathbf{x})$ as the unigram probabilities, where $\mathbf{x}$ is the training text.
- For bigram
  - One Categorical distribution for each bigram history $\mathrm{P}(x|h) = \mathrm{Cat}(x|\boldsymbol{r}_h)$
  - $r_\mathrm{h} \sim \mathrm{PY}(H, \alpha, d)$ as a prior for parameter of each bigram distribution
  - $H = \mathrm{Cat}(\boldsymbol{r}_1)$ is categorical distribution, but its parameters $\boldsymbol{r}_1$ are treated as random variable with prior $\boldsymbol{r}_1 \sim \mathrm{PY}(H_0, \alpha, d)$ (i.e. as for the unigram).

Y. Teh. A hierarchical Bayesian language model based on Pitman-Yor process.
In Proceedings of ACL International Conference, 2006.