Variational Inference for Dirichlet Process Mixtures and Beyond

A. Ciceri, M. Caputo, A. Curti, A. Dakouri, E. Di Liberatore, G. Frigerio
Tutored by Mario Beraha

15 February 2024

Table of contents

Variational Inference

Variational inference - introduction

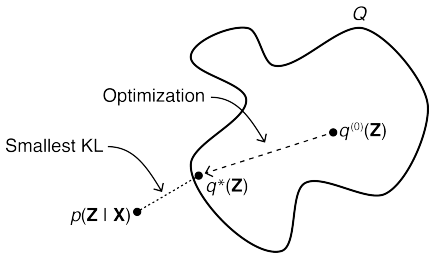

Figure 1: Visualization of VI optimization

Minimization of the Kullback-Leibler (KL) divergence:

$$q^*(z) = \underset{q(z) \in \mathcal{Q}}{\operatorname{argmin}} \, \mathrm{KL}\left(q(z) \| p(z|x)\right)$$

Since we cannot compute the KL, we optimize an alternative objective that is equivalent to the KL up to an added constant, which is called ELBO.

$$\mathrm{ELBO}(q) = \mathbb{E}_q[\log p(z, x)] - \mathbb{E}_q[\log q(z)]$$

Mixture of Gaussians

Bayesian Multivariate Gaussian Mixture Model

Hierarchical model:

$$
\begin{aligned}
x_i \mid c_i, \mu &\overset{\text{ind}}{\sim} \mathcal{N}_d\left(c_i^T \mu,\ \mathbb{I}_d\right) && i = 1, ..., n \\
\mu_k &\overset{\text{iid}}{\sim} \mathcal{N}_d\left(0,\ \sigma^2\ \mathbb{I}_d\right) && k = 1, ..., K \\
c_i &\overset{\text{iid}}{\sim} Cat(\frac{1}{K}, ..., \frac{1}{K}) && i = 1, ..., n
\end{aligned}
$$

i. K is the number of clusters
ii. c_i is the cluster assignment, it indicates which latent cluster x_i comes from
iii. n is the size of the sample
iv. d is the data dimension
v. σ^2 is a fixed hyperparameter

Variational densities

We want to approximate the posterior of μ_k, which is still a Gaussian, so we consider the following variational densities q:

$$q(\mu_k) \overset{\text{iid}}{\sim} \mathcal{N}_d\left(m_k,\ s_k^2\ \mathbb{I}_d\right) \qquad\qquad k = 1, ..., K$$

While to approximate the cluster allocation parameters we have:

$$P(c_i = j) = \varphi_{ij} \qquad\qquad k = 1, ..., K; \quad i = 1, ..., n$$

To define our variational densities we have to compute the following parameters:

- m: K vectors of means
- s^2: vector of variances
- φ: NxK matrix of cluster assignments probabilities

Updates

Variational update for the cluster assignment c_i:

$$\varphi_{ik} \propto exp[\ \mathbb{E}[\mu_k;\ m_k,\ s_k^2]x_i - \mathbb{E}[\mu_k^2;\ m_k,\ s_k^2]/2\ ]$$

Variational density of the kth mixture component expressed in terms of the variational mean and variance:

$$m_k = \frac{\sum_i \varphi_{ik} x_i}{\frac{1}{\sigma^2} + \sum_i \varphi_{ik}} \qquad s_k^2 = \frac{1}{\frac{1}{\sigma^2} + \sum_i \varphi_{ik}}$$

Optimization

Since VI can be considered an optimization problem, here we have the function to be optimized. In particular, the goal is to maximize the ELBO.

$$ELBO(m, s^2, \varphi) = \sum_{k=1}^{K} \mathbb{E}[\log p(\mu_k);\ m_k,\ s_k^2] +$$

$$\sum_{i=1}^{n} (\ \mathbb{E}[\log p(c_i);\ \varphi_i] + \mathbb{E}[\log p(x_i|c_i, \mu);\ \varphi_i,\ \hat{m},\ s^2]\ ) +$$

$$- \sum_{i=1}^{n} \mathbb{E}[\log q(c_i;\ \varphi_i)] - \sum_{k=1}^{K} \mathbb{E}[\log q(\mu_k;\ m_k,\ s_k^2)]$$

Each expectation can be computed in closed form. [6]

Results

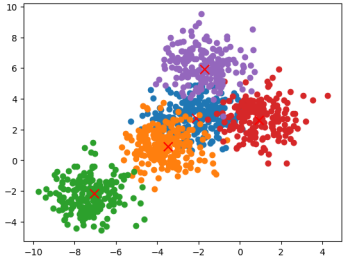

Figure 2: Our generated data

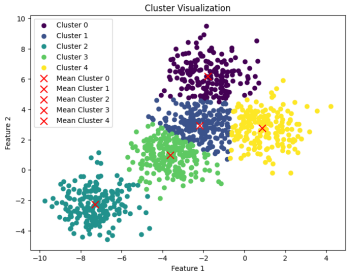

Figure 3: VI clustering

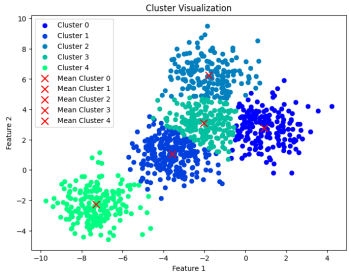

Figure 4: MCMC clustering

Comparison with MCMC (K=5, d=2)

TIME	N = 100	N = 1000	N = 10000
VI	508 ms ± 6.78 ms	2.31 s ± 304 ms	1min 16s ± 778 ms
MCMC	19 s ± 2 s	1min 47 s ± 7 s	21 min ± 1 min

CLUSTER ERROR	N = 100	N = 1000	N = 10000
VI	0.292078797	0.240227441	0.195773545
MCMC	0.326697788	0.246452642	0.197457232

DISTR. ERROR	N = 100	N = 1000	N = 10000
VI	33.17399	34.65825	64.832405
MCMC	0.24917	4.21288	33.59156

Dirichlet Process Mixture Model

Dirichlet process Mixture Model

Hierarchical model:

$$
\begin{aligned}
G \mid \alpha, G_0 &\sim DP(\alpha,\ G_0) && \\
\eta_n \mid G &\overset{\text{iid}}{\sim} G && n = 1, ..., N \\
X_n \mid \eta_n &\overset{\text{ind}}{\sim} p(x_n,\ \eta_n) && n = 1, ..., N
\end{aligned}
$$

i. α is a positive scaling parameter

ii. G_0 is a non-atomic probability distribution

iii. η_n follows an Pòlya urn distribution

iv. N is the total number of drawn η_n

VI adaptation

We consider this factorized family of variational distributions for mean-field variational inference:

$$q(v, \hat{\eta}, z) = \prod_{t=1}^{T-1} q_{\gamma_t}(v_t) \prod_{t=1}^{T} q_{\tau_t}(\hat{\eta_t}) \prod_{n=1}^{N} q_{\phi_n}(z_n)$$

Definition of the ELBO, the function to maximize:

$$ELBO(V, \hat{\eta}, Z) = \mathbb{E}_q[\log p(V \mid \alpha)] \ + \mathbb{E}_q[\log p(\hat{\eta} \mid \lambda)] +$$

$$\sum_{n=1}^{N} (\mathbb{E}_q[\log p(Z_n \mid V)] \ + \mathbb{E}_q[\log p(x_n \mid Z_n)]) - \mathbb{E}_q[\log q(V, \ \hat{\eta}, \ Z)]$$

Mean-field coordinate ascent algorithm

We can define the updates to be implemented:

$$\begin{aligned}
\gamma_{t,1} &= 1 + \textstyle\sum_{n=1}^{N} \phi_{n,t} \\
\gamma_{t,2} &= \alpha + \textstyle\sum_{n=1}^{N} \sum_{j=t+1}^{T} \phi_{n,j} \phi_{n,t} \\
\tau_{t,1} &= \lambda_1 + \textstyle\sum_{n=1}^{N} \phi_{n,t} x_n \\
\tau_{t,2} &= \lambda_2 + \textstyle\sum_{n=1}^{N} \phi_{n,t} \\
\phi_{n,t} &\propto \exp(S_t)
\end{aligned}$$

Where $t \in \{1, ..., T\}$ and $n \in \{1, ..., N\}$ and

$$S_t = \mathbb{E}_q[\log V_t] + \sum_{i=1}^{t-1} \mathbb{E}_q[\log(1 - V_t)] + \mathbb{E}_q[\hat{\eta}_t]^T X_n - \mathbb{E}_q[a(\hat{\eta})]$$

Iterating these updates optimizes the ELBO with respect to the variational parameters.

Applications

We explored two models:

- First model: α normally distributed, variance of each cluster constant;
- Second model: α follows a Normal distribution multiplied by an Inverse Gamma which incorporates a prior for the variance of the clusters.

First model: α normally distributed and σ^2 constant

Hierarchical model:

$$
\begin{aligned}
X_i|\mu_i &\overset{\text{ind}}{\sim} \mathcal{N}_d\left(\mu_i,\ \mathbb{I}_d\right) && i = 1, ..., N \\
\mu_i|P &\overset{\text{iid}}{\sim} P && i = 1, ..., N \\
P &\sim \mathcal{D}_\alpha \\
\alpha &= \mathcal{N}_d\left(0_d,\ \sigma^2\mathbb{I}_d\right)
\end{aligned}
$$

Second model: α as multiplication of Normal and Inverse Gamma distributions with a prior for custers' variance

Hierarchical model:

$$
\begin{aligned}
X_i|\mu_i, \tau_i^2 &\overset{\text{ind}}{\sim} \mathcal{N}_d\left(\mu_i,\ \tau_i^2\mathbb{I}_d\right) && i = 1, ..., N \\
\mu_i, \tau_i^2|P &\overset{\text{iid}}{\sim} P && i = 1, ..., N \\
P &\sim \mathcal{D}_\alpha \\
\alpha &= \mathcal{N}_d\left(0_d,\ \sigma^2\mathbb{I}_d\right) \times IG(a, b)
\end{aligned}
$$

Results

Using the second model with N=1000, d=2, α=5 we get:

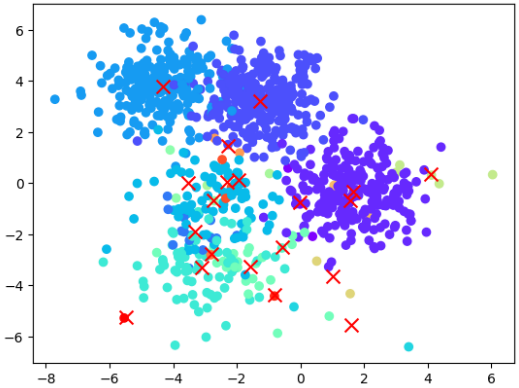

Figure 5: Our generated data

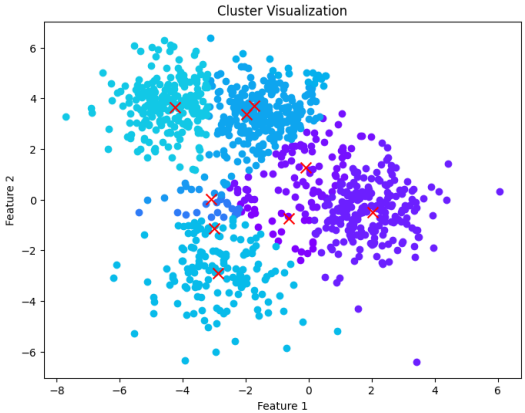

Figure 6: VI clustering

Results First Model, VI vs MCMC(bayesmix)

TIME	N = 100	N = 1000	N = 10000
VI	6 s	32 s	7 min
MCMC	18.3 s	1 min 3 s	1 hr 4 min

K ERROR	N = 100	N = 1000	N = 10000
VI	0.3879607	0.3750703	0.3926690
MCMC	0.4954638	0.3623457	0.3411489

REL. ERROR	N = 100	N = 1000	N = 10000
VI/MCMC	-0.21133067	-0.39421669	0.52850597

N CLUSTERS	N = 100	N = 1000	N = 10000
VI	8 on 15	13 on 26	15 on 40
MCMC	12 on 15	17 on 26	19 on 40

Results Second Model, VI vs MCMC(Stan)

TIME	N = 100	N = 1000	N = 10000
VI	4 s	12 s	5 min
MCMC	9 min 23 s	1 hr 3 min	x

K ERROR	N = 100	N = 1000	N = 10000
VI	0.292883244	0.4133986977	0.253740856
MCMC	0.7537454079	0.702642917	x

REL. ERROR	N = 100	N = 1000	N = 10000
VI/MCMC	-0.08534313	0.02627501	x

N CLUSTERS	N = 100	N = 1000	N = 10000
VI	8 on 11	8 on 20	11 on 34
MCMC	6 on 11	5 on 20	x

Indian Buffet Process - IBP

What is the IBP

The Indian Buffet Process (IBP) is a nonparametric prior for latent feature models in which observations are influenced by a combination of hidden features. Is based on infinite binary matrices that allows us to simultaneously infer which features influence a set of observations and how many features there are.

IBP model - culinary metaphor

IBP model - culinary metaphor

The first customer takes the first Poisson(α) dishes

IBP model - culinary metaphor

The first customer takes the first Poisson(α) dishes

The second customer then takes dishes that have been previously sampled with probability 1/2. He also takes Poisson(α/2) new dishes.

IBP model - culinary metaphor

The first customer takes the first Poisson(α) dishes

The second customer then takes dishes that have been previously sampled with probability 1/2. He also takes Poisson(α/2) new dishes.

⋮

The i-th customer then takes dishes that have been previously sampled with probability m_k/i, where m_k is the number of people who have already sampled dish k. He also takes Poisson(α/i) new dishes.

IBP model

Let's consider the general model, we'll introduce:

- X an N×D matrix where each of the N rows contains a D-dimensional observation. In our analysis, as done in our reference paper, X can be approximated by ZA.
- Z is an N × K binary matrix, each column corresponds to the presence of a latent feature.

$$\begin{cases} z_{nk} = Z(n,k) = 1 & \text{if feature k is present in observation n} \\ z_{nk} = Z(n,k) = 0 & \text{otherwise} \end{cases}$$

- A is a K × D matrix. The values for feature k are stored in row k of A.
- We also introduce the measurement noise ϵ which we assume to be independent form A and Z and uncorrelated across observations.
- The observed data X is given by $X = ZA + \epsilon$

IBP model

Given X, we wish to find the posterior distribution of Z and A. We do this using Bayes rule:

$$p(Z, A|X) \propto p(X|Z, A)p(Z)p(A)$$

where we have assumed that Z and A are a priori independent. Regarding the prior for Z, since often K is unknown, the idea is to place a flexible prior on Z that allows K to be determined at inference time.

IBP model

The Indian Buffet Process places the following prior on [Z], a canonical form of Z that is invariant to the ordering of the features.

$$p([Z]) = \frac{\alpha^K}{\prod_{h \in \{0,1\}^N \setminus \mathbf{0}} K_h!} \exp\{-\alpha H_N\} \prod_{k=1}^{K} \frac{(N - m_k)!(m_k - 1)!}{N!}$$

where:

- K is the number of nonzero columns in Z;
- m_k is the number of ones in the k^{th} column of Z;
- H_N is the N^{th} harmonic number;
- α controls the expected number of features present in each observation;
- K_h is the number of occurrences of the non-zero binary vector h among the columns in Z.

IBP model - Stick breaking construction

To generate a matrix Z from the IBP prior using the stick-breaking construction, we begin by assigning a parameter $\pi_k \in (0, 1)$ to each column of Z, such that

$$z_{nk}|\pi_k \overset{ind}{\sim} \text{Bernoulli}(\pi_k)$$

The π_k themselves are generated by a stick-breaking process. We first draw a sequence of iid random variables $v_1, v_2, ...$ from a Beta$(\alpha, 1)$. Then, we let $\pi_1 = v_1$, and for each k we have:

$$\pi_k = v_k \pi_{k-1} = \prod_{i=1}^{k} v_i$$

The expression for π_k shows that, in a set of N observations, the probability of seeing feature k decreases exponentially with k. We also see that larger values of α mean that we expect to see more features in the data.

Variational Inference setting

Following the approach of the paper [4], we have the following:

$$A_k \overset{iid}{\sim} N_d(0,\ \sigma_A^2 \mathbb{I}_d) \qquad k = 1, 2, \ldots, K$$

$$\epsilon_n \overset{iid}{\sim} N_d(0,\ \sigma_X^2 \mathbb{I}_d) \qquad n = 1, 2, \ldots, N$$

Set of hidden variables in the IBP $\mathbf{W} = \{\pi, \mathbf{Z}, \mathbf{A}\}$;
Set of parameters $\theta = \{\alpha, \sigma_A^2, \sigma_X^2\}$.
As always, we use the mean field variational method to approximate it with a variational distribution $q(\mathbf{W})$ from some tractable family of distributions Q. For the IBP, we will let Q be the factorised family:

$$q(\mathbf{W}) = q_\tau(\pi) q_\phi(\mathbf{A}) q_\nu(\mathbf{Z})$$

where τ , ϕ, and ν are the variational parameters that we want to optimise in order to minimize the ELBO function.

Variational Inference setting

We employed two different methods:

- finite variational approach;
- infinite variational approach.

In both approaches the idea is to apply a truncation level K to the maximum number of features in the variational distribution.

Finite dimensional approach

The finite variational approach considers a finite *Beta-Bernoulli approximation* to the IBP.

Finite Beta-Bernoulli model with K features:

$$z_{nk}|\pi_k \overset{ind}{\sim} \text{Bernoulli}(\pi_k) \qquad \forall n = 1, ..., N$$

$$\pi_k \overset{iid}{\sim} \text{Beta}(\alpha/K, 1) \qquad \forall k = 1, ..., K$$

The finite variational approach approximates the true IBP model $p(W, X|\theta)$ with $p_K(W, X|\theta)$. We use a fully factorised variational distribution, where:

$$q_{\tau_k}(\pi_k) = \text{Beta}(\pi_k; \tau_{k1}, \tau_{k2}) \qquad \forall k = 1, ..., K$$

$$q_{\phi_k}(A_{k\cdot}) = N(A_{k\cdot}; \phi_k, \Phi_k) \qquad \forall k = 1, ..., K$$

$$q_{\nu_{nk}}(z_{nk}) = \text{Bernoulli}(z_{nk}; \nu_{nk}) \qquad \forall k = 1, ..., K \text{ and } \forall n = 1, ..., N$$

Finite dimensional approach

We consider:

$$q_J^*(z_J) \propto \exp\left\{\mathbb{E}_{-J}[\log p(z_J|z_{-J}, x)]\right\}$$

where z are the latent variables, in this case $z = (z_{nk}, A_k, \pi_k)$.

To retrieve the updates of the parameters we first reconstructed the kernels and then we found their expression.

Finite dimensional approach: updates of the parameters

In the following lines we only report the parameter expression:

$$\tau_{k1} = \sum_{n=1}^{N} \nu_{nk} + \frac{\alpha}{K}; \qquad \tau_{k2} = N - \sum_{n=1}^{N} \nu_{nk} + 1;$$

$$\phi_k = \frac{1}{\sum_{n=1}^{N} \nu_{nk}/\sigma_X^2 + 1/\sigma_A^2} \left(\sum_{n=1}^{N} \nu_{nk} x_n - \sum_{n=1}^{N} \nu_{nk} \sum_{j \neq k} \nu_{nj} \phi_k \right) \frac{1}{\sigma_X^2};$$

$$\Phi_k = \frac{1}{\sum_{n=1}^{N} \nu_{nk}/\sigma_X^2 + 1/\sigma_A^2} \mathbb{I}_d;$$

$$\nu_{nk} \propto \exp\left(-\frac{1}{2\sigma_X^2} (-2 x_n^T \phi_k + 2\phi_k^T (\sum_{j \neq k} \nu_{nj} \phi_j) + d\Phi_k + \phi_k^T \phi_k) + \psi(\tau_{k,1}) - \psi(\tau_{k,2}) \right)$$

Finite dimensional approach - ELBO

After the definition of the finite variational approach and the definition of the updates, we can consider the computation of the ELBO in this setting.

$$ELBO = H[q] + \sum_{k=1}^{K} \mathbb{E}_{\pi}[\ln p(\pi_k|\alpha)] + \sum_{k=1}^{K} \mathbb{E}_{\mathbf{A}}[\ln p(\mathbf{A}_{k\cdot}|\sigma_A^2)] + \\ + \sum_{k=1}^{K}\sum_{n=1}^{N} \mathbb{E}_{\pi,\mathbf{Z}}[\ln p(z_{nk}|\pi)] + \sum_{n=1}^{N} \mathbb{E}_{\mathbf{Z},\mathbf{A}}[\ln p(\mathbf{X}_{n\cdot}|\mathbf{Z},\mathbf{A},\sigma_X^2)]$$

Where H is the entropy of the distribution, i.e. $-\mathbb{E}_q[\log q]$.

Infinite dimensional approach

Using this second approach we will work with the distribution of the stick-breaking variables $v = \{v_1, ..., v_K\}$ instead of directly approximating the distribution of π_k. In our truncated model with truncation level K, we have:

$$\pi_k = \prod_{i=1}^{k} v_i \times \mathbb{I}_{\{k<K\}}$$

The advantage of using v as our hidden variable is that under the IBP prior, the $\{v_1, ..., v_K\}$ are independent draws from a $Beta(\alpha, 1)$, whereas the $\{\pi_1 ... \pi_K\}$ are dependent. Here the factorized variational distribution is given by:

$$q(\mathbf{W}) = q_\tau(v) q_\phi(\mathbf{A}) q_\nu(\mathbf{Z})$$

where $\forall k = 1, ..., K$: $q_{\tau_k}(v_k) = \text{Beta}(v_k; \tau_{k1}, \tau_{k2})$; $q_{\phi_k}(A_k.) = N(A_k.; \phi_k, \Phi_k)$;
and $\forall k = 1, ..., K$, $\forall n = 1, ..., N$: $q_{\nu_{nk}}(z_{nk}) = \text{Bernoulli}(z_{nk}; \nu_{nk})$

Infinite dimensional approach

The only parameter update that changes from before is the update of ν, in particular we add the term $\mathbb{E}_v\left[\ln p(z_{nk}|v)\right]$

Also the ELBO expression is very similar to before, now it depends on v and not π:

$$ELBO = H[q] + \sum_{k=1}^{K} \mathbb{E}_v[\ln p(v_k|\alpha)] + \sum_{k=1}^{K} \mathbb{E}_{\mathbf{A}}[\ln p(\mathbf{A}_{k\cdot}|\sigma_A^2)] + \\ + \sum_{k=1}^{K}\sum_{n=1}^{N} \mathbb{E}_{v,\mathbf{Z}}[\ln p(z_{nk}|v)] + \sum_{n=1}^{N} \mathbb{E}_{\mathbf{Z},\mathbf{A}}[\ln p(\mathbf{X}_{n\cdot}|\mathbf{Z},\mathbf{A},\sigma_X^2)]$$

ELBO and updates approximation

For this model in the computations it appears $\mathbb{E}_q[\log(1 - \prod_{i=1}^{k} v_i)]$, which has not a closed form. Two possible approximations were proposed, a Taylor expansion approximation and a multinomial lower bound, we used the second one.

$$\begin{aligned}
\mathbb{E}_q[\log(1 - \prod_{i=1}^{k} v_i)] &= \mathbb{E}_{v,Z}[\ln(\sum_{y=1}^{k} q_k(y) \frac{(1 - v_y)\prod_{m=1}^{y-1} v_m}{q_k(y)})] \\
&\geq \mathbb{E}_v \mathbb{E}_y[\ln((1 - v_y)\prod_{m=1}^{y-1} v_m) - \ln q_k(y)] \\
&= \mathbb{E}_y[\psi(\tau_{y2}) + \sum_{m=1}^{y-1} \psi(\tau_{m1}) - \sum_{m=1}^{y} \psi(\tau_{m1} + \tau_{m2})] + H(q_k)
\end{aligned}$$

where the update for $q_k(y)$ is found by taking the derivative and maximise this lower bounds, finding out:

$$q_k(y) \propto exp\{\psi(\tau_{y2}) + \sum_{m=1}^{y-1} \psi(\tau_{m1}) - \sum_{m=1}^{y} \psi(\tau_{m1} + \tau_{m2})\}$$

Optimization Problem

On these models we have particularly experienced a problem: the starting initialization distributions heavily influenced the final result. All the initializations are random and within a single run they are all iid but the seed of the sampling changes. However, changing the type of distributions or even only the hyperparameters changes heavily the final result because of the presence of local optima.
For example:

$$\nu_{nk} \overset{iid}{\sim} U([0,1]) \qquad \Rightarrow \text{bad results}$$

$$\tilde{\nu}_{nk} \overset{iid}{\sim} Beta(a,b), \qquad \nu_{nk} = \prod_{j=1}^{k} \tilde{\nu}_{nj} \qquad \Rightarrow \text{good results}$$

Results

Once we took into account the initializations problem, the algorithms worked better. Again we don't look for the perfect feature-sample allocation, but for a distribution approximation. Here we can interpret more features allocated to almost every sample as the approximation of only one real feature allocated to almost every sample.

Conclusions

Conclusions

In summary, the we applied variational inference algorithms to five different models. Although the code consistently demonstrated speed, it consistently fell short in accuracy compared to the MCMC algorithm. However, the purpose wasn't to find perfect cluster allocations or posterior parameters but rather to approximate the distribution function of a sample.
Variational inference yielded satisfactory results within this framework, despite the introduced trade-offs, particularly for obtaining quick approximations.

Conclusions

Another trade-off arises when deciding between MCMC and variational inference: the frequency of usage. If you require frequent runs, implementing variational inference might be beneficial despite the initial implementation time. However, if usage is not frequent, the time saved during each execution might not outweigh the implementation time, favoring MCMC instead. This consideration could be the crucial factor in determining whether to use variational inference or MCMC.

References

[1] Christopher M. Bishop. **Pattern recognition and machine learning.** New York : Springer, [2006] ©2006, 2006. URL: https://search.library.wisc.edu/catalog/9910032530902121.

[2] David M. Blei and Michael I. Jordan. **"Variational inference for Dirichlet process mixtures".** In: *Bayesian Analysis* 1.1 (2006), pp. 121–143. DOI: 10.1214/06-BA104. URL: https://doi.org/10.1214/06-BA104.

[3] David M. Blei, Alp Kucukelbir, and Jon D. McAuliffe. **"Variational Inference: A Review for Statisticians".** In: *Journal of the American Statistical Association* 112.518 (Apr. 2017), pp. 859–877. DOI: 10.1080/01621459.2017.1285773. URL: https://doi.org/10.1080%2F01621459.2017.1285773.

[4] Finale Doshi et al. **"Variational Inference for the Indian Buffet Process".** In: *Proceedings of the Twelth International Conference on Artificial Intelligence and Statistics*. Ed. by David van Dyk and Max Welling. Vol. 5. Proceedings of Machine Learning Research. Hilton Clearwater Beach Resort, Clearwater Beach, Florida USA: PMLR, 16–18 Apr 2009, pp. 137–144. URL: https://proceedings.mlr.press/v5/doshi09a.html.

[5] Ankush Ganguly and Samuel W. F. Earp. **An Introduction to Variational Inference.** 2021. arXiv: 2108.13083 [cs.LG].

[6] Haoliang. **"Variational Inference in Bayesian Multivariate Gaussian Mixture Model".** In: *Bayesian Analysis* (2020). DOI: Hao-2020.

[7] Github repository. **"Github repository".** Here we have our codes.

[8] Yee Whye Teh, Dilan Grür, and Zoubin Ghahramani. **"Stick-breaking Construction for the Indian Buffet Process".** In: *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics*. Ed. by Marina Meila and Xiaotong Shen. Vol. 2. Proceedings of Machine Learning Research. San Juan, Puerto Rico: PMLR, 21–24 Mar 2007, pp. 556–563. URL: https://proceedings.mlr.press/v2/teh07a.html.