POLITECNICO
MILANO 1863

Variational Inference for Dirichlet Process Mixtures and Beyond

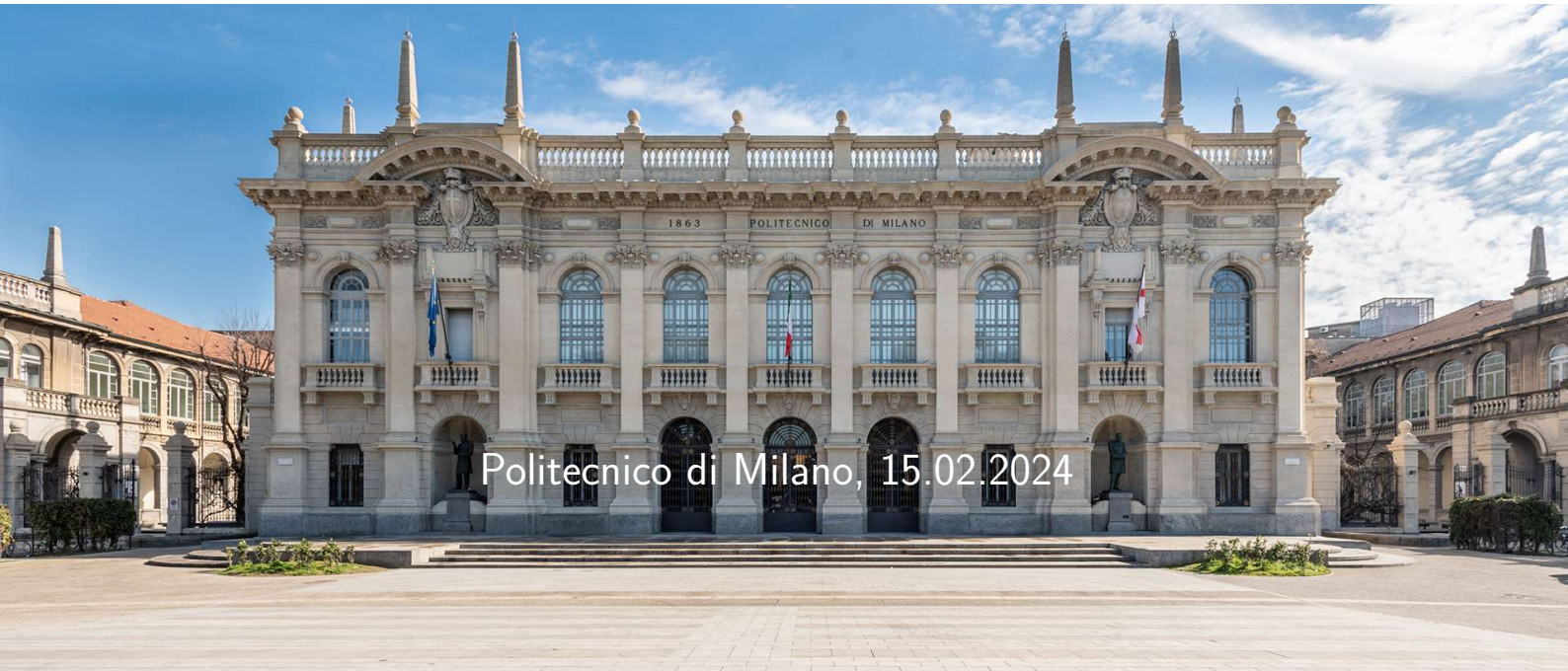
Bayesian statistics

Group R8

Group members

Alessandro Ciceri
Michele Caputo
Angelo Curti
Ange Dakouri
Elena Di Liberatore
Giulia Frigerio

Tutor: Mario Beraha



Politecnico di Milano, 15.02.2024

Contents

1 | Introduction on Variational Inference

1.1 | What is variational inference

One of the core problems in Bayesian statistics is approximating difficult-to-compute probability densities. Variational Inference (VI) is a deterministic method that uses optimization techniques to estimate complex probability densities. This method converges faster than classical methods, such as Markov Chain Monte Carlo sampling. ([3], [5])
The idea behind VI is to first posit a family of densities $\mathcal{Q}$ over the lantent varibles $\mathbf{z}$. Each q in $\mathcal{Q}$ is a possible approximation of the real density. The final goal is to find the candidate that is closer to the target density by minimizing the KL divergence:

$$\mathrm{KL}\left(q(z)\|p(z|x)\right) = \mathbb{E}_q[\log q(z)] - \mathbb{E}_q[\log p(z,x)] + \log p(x).$$

It is a measure of information that quantifies how similar a probability distribution p(x) is to a candidate distribution q(x). Inference now amounts to solving the following optimization problem:

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

The KL divergence is not directly computable, so we rely on a different quantity, the evidence lower bound (ELBO). This object is equivalent to KL up to an added constant:

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

The first term is an expected likelihood; it encourages densities that place their mass on configurations of the latent variables that explain the observed data. The second term is the negative divergence between the variational density and the prior; it encourages densities close to the prior.
Another property of the ELBO is that it lower-bounds the (log) evidence, log p(x) $\geq$ ELBO(q) for any q(z). This explains the name. To see this look the following expression:

$$logp(x) = \mathrm{KL}\left(q(z)\|p(z|x)\right) + \mathrm{ELBO}(q)$$

The bound then follows from the fact that KL ($\cdot$) ≥ 0.
A very crucial point in Variational Inference is the choice of the family of distributions $\mathcal{Q}$: in fact this introduces a trade-off between precision and efficiency of the algorithm. We want to solve the problem with a small error but we want to do it fast, so we need a family that is flexible enough to allow a good result, but not too complex to make the computational cost too high. A usual proposal for $\mathcal{Q}$ is the exponential distribution family, which gives both good efficiency and good flexibility.

1.2 | Mean Field Variational Family

A particular choice of the family $\mathcal{Q}$ stands in the mean field family, In this field the latent variables are mutually independent and each governed by a distinct factor in the variational density. A generic member of the mean-field variational family is:

$$q(z) = \prod_{j=1}^{m} q_j(z_j)$$

Each latent variable z_j is governed by its own variational factor, the density $q_j(z_j)$. We emphasize that the variational family is not a model of the observed data, indeed, the data do not appear in the previous equation. Instead, it is the ELBO, and the corresponding KL minimization problem, that connects the fitted variational density to the data and model.

One of the most commonly used algorithms for solving this optimization problem is coordinate ascent variational inference (CAVI) [1]. CAVI iteratively optimizes each factor of the mean-field variational density, while holding the others fixed. It climbs the ELBO to a local optimum [3]. This is the algorithm we will use in our next implementations.

1.3 | MCMC vs VI

The main advantage of the VI respect to MCMC is the capability to assess the solution faster in terms of the time. This creates the possibility explore an important amount of possible different solutions. In other terms this methods is perfectly suitable for large datasets and scenarios where it is needed to to explore many models. But the price that we pay for this is that the solution is not guaranteed to be exact.
On the other hand MCMC algorithms are very slow in terms of time and very heavy in terms of computations, but asymptotically give a precise solution. This methos is perfect for small datasets and more precise results are needed.

So, there is not a right or wrong method. The choice depends on the goal we want to achieve. But one thing is clear: we are going through a *computational cost/exactness of the solution* trade-off.

2 | Gaussian Mixture model

2.1 | The model

In our project we implemented an algorithm for Bayesian mixture of Gaussians as the first application of variational inference.
Consider a Bayesian mixture of multivariate Gaussians which has K components, corresponding to K Gaussian distributions with means vectors $\{\mu_1, .., \mu_K\}$ and variance equal to the identity matrix $\mathbb{I}_d$, with d the dimension of our data. The mean parameters are independent and we assume them to be distributed as a Gaussian $\mathcal{N}_d(0,\ \sigma^2\ \mathbb{I}_d)$, where the prior variance σ^2 is a hyperparameter.
Note that c_i indicates which latent cluster x_i belongs to and it is distributed as a categorical distribution over $\{1, \dots, K\}$. It is a K-dimensional vector with all zeros except in the position of x_i's cluster.
Finally the datum x_i has distribution $\mathcal{N}_d\left(c_i^T\mu,\ \mathbb{I}_d\right)$.
Here the complete hierarchical model:

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

In the following discussion we'll use p to indicate the distributions of the model, depending on the argument it could be the joint or one of the ones above. In particular, the full joint distribution has the following form:

$$p(\mu, c, x) = \prod_{k=1}^{K} p(\mu_k) \prod_{i=1}^{N} p(c_i)p(x_i|c_i, \mu)$$

2.2 | Variational inference setting

What we want to approximate is the posterior distribution of μ_k for each k in $1, ..., K$ and the cluster assignment probabilities for each sample. First of all it's important to say that we work in a mean-field framework, so the μ_k will be mutually independent in our family $\mathcal{Q}$ as well as the cluster assignment probabilities. The choice of $\mathcal{Q}$ has been simple since we know that the posterior distribution of μ_k is still a Gaussian:

$$q(\mu, \mathbf{c}) = \prod_{k=1}^{K} q(\mu_k; m_k, s_k^2) \prod_{i=1}^{n} q(c_i; \varphi_i)$$

Then we consider the following variational densities:

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

While to approximate the cluster allocation parameters we have:

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

The mixture components are Gaussian with variational parameters (mean and variance) specific to the kth cluster; the cluster assignments are categorical with variational parameters (cluster probabilities) specific to the ith data point. In fact, these are the optimal forms of the mean-field variational density for the mixture of Gaussians [3].
We can derive the variational update for the ith cluster assignment:

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

Moreover we can derive the variational density of the kth mixture component expressed in terms of the variational mean and variance:

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

After the definition of the variational family, we need to specify the ELBO to completely specify the variational inference problem for the mixture of Gaussians. The ELBO is a function of the variational parameters $\mathbf{m}$, $\mathbf{s}$ and φ and is defined as follows:

$$\begin{aligned}\text{ELBO}(\mathbf{m}, \mathbf{s^2}, \varphi) = \sum_{k=1}^{K} \mathbb{E}_q[\log p(\mu_k);\ m_k,\ s_k^2] + \\ \sum_{i=1}^{n}(\ \mathbb{E}_q[\log p(c_i);\ \varphi_i] + \mathbb{E}_q[\log p(x_i|c_i,\mu);\ \varphi_i,\ \mathbf{m},\ \mathbf{s^2}]\ ) + \\ - \sum_{i=1}^{n} \mathbb{E}_q[\log q(c_i;\ \varphi_i)] - \sum_{k=1}^{K} \mathbb{E}_q[\log q(\mu_k;\ m_k,\ s_k^2)]\end{aligned}$$

Each expectation can be computed in close form [6], the full computation can be found in appendix A:

- first term:

$$\sum_{k=1}^{K} \mathbb{E}_q[\log p(\mu_k);\ m_k,\ s_k^2] = -\frac{1}{2\sigma^2}\sum_{k=1}^{K}(ds_k^2 + m_k^T m_k) - \frac{Kd}{2}\log(2\pi\sigma^2)$$

- second term:

$$\sum_{i=1}^{n}(\ \mathbb{E}_q[\log p(c_i);\ \varphi_i]) = \sum_{i=1}^{n}(\ \mathbb{E}_{q_\mu, q_c}[\log \frac{1}{K}]) = -n\log K$$

- third term:

$$\sum_{i=1}^{n}(\mathbb{E}_q[\log p(x_i|c_i,\mu)) = \sum_{i=1}^{n}\sum_{k=1}^{K} -\frac{p}{2}\log(2\pi)\varphi_{i,k} - \frac{1}{2}\sum_{i=1}^{n}\sum_{k=1}^{K}\phi_{i,k}(x_i^T x_i - 2x_i^T m_k + ds_k^2 + m_k^T m_k)$$

- fourth term:

$$\sum_{i=1}^{n} \mathbb{E}_q[\log q(c_i;\ \varphi_i)] = \sum_{i=1}^{n}\sum_{k=1}^{K} \varphi_{i,k}\log\varphi_{i,k}$$

- fifth term:

$$\sum_{k=1}^{K} \mathbb{E}_q[\log q(\mu_k;\ m_k,\ s_k^2)] = \sum_{k=1}^{K} -\frac{d}{2}\log(2\pi s_k^2) - \frac{Kd}{2}$$

2.3 | Results

2.3.1 | algorithm results

To test the algorithm we generate some data after having defined the dimention d, the total number of clusters K, the number of samples N.
To generate the data we have followed the hierarchical model: we first sampled the N cluster assignment over K clusters and the K means from their respective distributions, then we sampled the proper data from the conditional distribution given the cluster means and assignments above. In this way we have the exact hierarchy described in the model.

Example
Considering bi-dimensional data, 5 clusters and 1000 samples we obtained the following result:

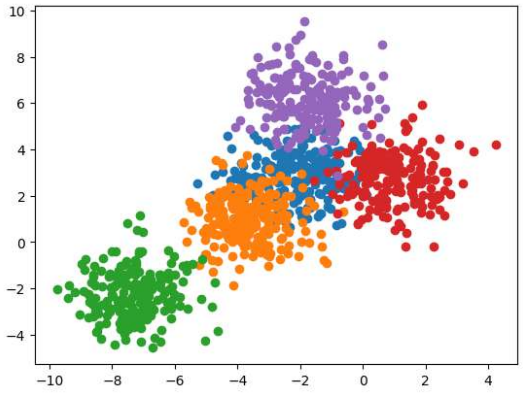


Figure 2.1: Our generated data

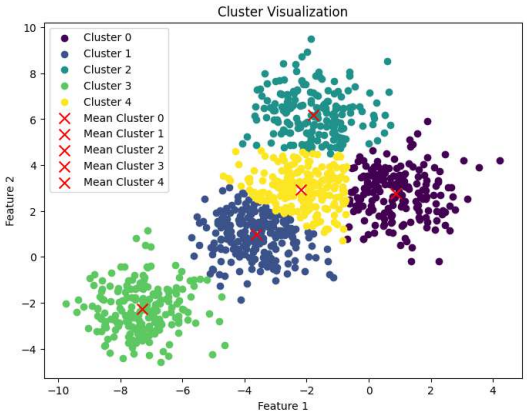


Figure 2.2: Clusters from VI

We can see from the figures that our algorithm captures the clusters as well as their posterior mean in a quite precise way. It struggles in predicting the cluster assignments in the overlapping regions between two clusters.

2.3.2 | Comparison with MCMC

We compared the results obtained with VI with those obtained by implementing a Gibbs sampler algorithm using STAN. To do so we had first to find the real posterior distribution of μ_k, which we can find in a closed form since we are generating the data hence we have everything we need.

$$\mathcal{L}(\mu_k|rest) \propto \mathcal{L}(y, c, \mu, \sigma^2) \propto \mathcal{L}(y|c, \mu)\pi(c)\pi(\mu) \propto \prod_{c_i=k} \mathcal{L}(y_i|\mu_k)\pi(\mu_k) =$$

$$= exp\{ \sum_{c_i=k} (y_i - \mu_k)^T(y_i - \mu_k) + \frac{\mu_k^T\mu_k}{\sigma^2}\} \propto exp\{ (N_k + \frac{1}{\sigma^2})(\mu_k^T\mu_k - 2\frac{1}{N_k + \frac{1}{\sigma^2}}\mu_k^T \sum_{c_i=k} y_i)\}$$

with N_k the number of samples belonging to cluster k.
From the last line we can obtain the kernel of a Gaussian distribution, with following

parameters:

$$m_{k_{post}} = \frac{1}{N_k + \frac{1}{\sigma^2}} \sum_{c_i=k} y_i \qquad k = 1, ..., K$$

$$s^2_{k_{post}} = \frac{1}{N_k + \frac{1}{\sigma^2}} \qquad k = 1, ..., K$$

$$\mu_k | rest \overset{\text{ind}}{\sim} \mathcal{N}_d \left(m_{k_{post}},\ s^2_{k_{post}}\ \mathbb{I}_d \right) \qquad k = 1, ..., K$$

Now we can comprehend the goodness of our approximations.
Here the dimension of our data and the number of clusters are fixed, d=2 and K=5.

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

K 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

It is important to say that the hyperparameters of the algorithms are fixed: the tolerance of the VI as well as the number of iterations of MCMC. In fact we know that MCMC guarantees convergence but the error in the distribution when N=10000 is not so low: this happens because it would have needed more iterations to converge.
The errors estimates are described as follows:

- The cluster error refers to sklearn.metrics.adjusted_rand_score command in phyton. It is able to couple the predicted clusters and the real ones even if at the same cluster is assigned a different index in the two cluster assignments, and then returns a metric of the precision of the assignments. In particular, it returns 1 if there is no misassignment, 0 if the cluster prediction is equivalent to the one of a random classifier, and up to -0.5 if it performs worse than the random classifier. We use 1- the return value so that we have positive but low error for good assignment and error up to 1.5 for bad assignments.
- The distribution error is an approximation of the L^2 norm of the difference between the real distribution and the predicted one, given that they are both gaussian densities, so both distributions are uniquely identified by the predicted parameters. As posterior probability for the predicted distribution we use the posterior approximation of the cluster assignments probabilities and the predicted density is a weighted sum over the single clusters distributions:

$$w_k = \frac{1}{N} \sum_{i=1}^{N} \varphi_{ik}$$

$$\hat{f} = \sum_{k=1}^{K} w_k \mathcal{N}_d \left(m_k,\ s^2_k\ \mathbb{I}_d \right)$$

3 | Dirichelet Process Mixture model

3.1 | The model

Dirichlet process (DP) mixture models serve as the foundation for nonparametric Bayesian statistics. The advancement of Monte-Carlo Markov chain (MCMC) sampling techniques for DP mixtures has facilitated the implementation of nonparametric Bayesian approaches in addressing a range of practical data analysis challenges.
The Dirichlet process (DP), is a measure function of other measures. The DP is parameterized by a base distribution G_0 and a positive scaling parameter α. Suppose we draw a random measure G from a Dirichlet process and independently draw N random variables η_n from G:

$$\begin{aligned} G|G_0, \alpha &\sim \mathrm{DP}(G_0, \alpha) \\ \eta_n &\overset{iid}{\sim} G, \qquad n \in \{1, \dots, N\} \end{aligned}$$

By marginalizing the random measure G, the collective distribution of $\{\eta_1, \dots, \eta_N\}$ adheres to a Pólya urn scheme. The idea is to consider for example a urn with blue and white balls and for each ball extracted, we reinsert the ball and an additional one identical to the other. In this way the probability of extracting a ball of that color will increase over time. In these configurations, where distinct η_n assume identical values, they receive positive probability. More precisely, the variables $\{\eta_1, \dots, \eta_{n-1}\}$ are randomly partitioned according to which variables are equal to the same value, with the distribution of the partition obtained from that scheme. Let $\{\eta_1^*, \dots, \eta_{|c|}^*\}$ denote the distinct values of $\{\eta_1, \dots, \eta_{n-1}\}$, let $c = \{c_1, \dots, c_{n-1}\}$ be assignment variables such that $\eta_i = \eta_{c_i}^*$, and let $|c|$ denote the number of cells in the partition. The distribution of η_n follows the urn distribution:

$$\eta_n = \begin{cases} \eta_i^* & \text{with probability } \frac{|\{j:c_j=i\}|}{n-1+\alpha} \\ \eta, \eta \sim G_0 & \text{with probability } \frac{\alpha}{n-1+\alpha} \end{cases}$$

where $|\{j : c_j = i\}|$ is the number of times the value η_i^* occurs in $\{\eta_1, \dots, \eta_{n-1}\}$.
We are going to work in stick-breaking setting in which the random probability measure G is discrete with probability one and is explicitly expressed as an infinite sum of atomic measures. A more explicit characterization of the DP in terms of a stick-breaking construction is obtained considering two infinite collections of independent random variables:

$$\begin{aligned} V_i &\overset{iid}{\sim} Beta\,(1,\ \alpha) && i = 1, 2, \dots \\ \eta_i^* &\overset{iid}{\sim} G_0 && i = 1, 2, \dots \end{aligned}$$

The stick-breaking representation of G is as follows:

$$\begin{aligned} \pi_i(v) &= v_i \prod_{j=1}^{i-1} (1 - v_j) \\ G &= \sum_{i=1}^{\infty} \pi_i(v) \delta_{\eta_i^*} \end{aligned}$$

The support of G consists of a countably infinite set of atoms, drawn independently from G_0. The mixing proportions $\pi_i(v)$ are given by successively breaking a unit length "stick" into an infinite number of pieces. The size of each successive piece, proportional to the rest of the stick, is given by an independent draw from a Beta(1,α) distribution. Let Z_n be an assignment variable of the mixture component with which the data point x_n is associated. The data can be described as arising from the following process:

1. Draw $V_i|\alpha \overset{iid}{\sim} Beta(1, \alpha), \quad \mathrm{i} = \{1, 2, \dots\}$
2. Draw $\eta_i^*|G_0 \overset{iid}{\sim} G_0\ , \quad \mathrm{i} = \{1, 2, \dots\}$
3. For the n^{th} data point:
   [a] Draw $Z_n|\{v_1, v_2, \dots\} \overset{iid}{\sim} \mathrm{Mult}(\pi(v))$
   [b] Draw $X_n|z_n \overset{ind}{\sim} p(x_n|\eta_{z_n}^*)$

Leveraging the discrete nature of G, the Dirichlet process mixture model can be interpreted as a mixture model characterized by an infinite number of potential components. In the Dirichlet process mixture model, the DP is used as a nonparametric prior in a hierarchical Bayesian specification:

$$G|\alpha, G_0 \overset{iid}{\sim} DP(\alpha, G_0)$$
$$\eta_n|G \overset{iid}{\sim} G$$
$$X_n|\eta_n \overset{ind}{\sim} p(x_n|\eta_n)$$

It is extremely important to highlight that η_n create a clustering effect over the data.

3.2 | Variational inference setting

Variational inference provides an alternative, deterministic methodology for approximating likelihoods and posteriors. Consider a model with hyperparameters θ, latent variables $W = \{W_1, \dots, W_M\}$, and observations $x = \{x_1, \dots, x_N\}$. The posterior distribution of the latent variables is:

$$p(w|x, \theta) = \exp\{\log p(x, w|\theta) - \log p(x|\theta)\}$$

These methods rely on optimizing the Kullback-Leibler (KL) divergence concerning a variational distribution. Specifically, consider $q_\nu(w)$ as a family of distributions parameterized by a variational parameter ν . The objective is to minimize the KL divergence between $q_\nu(w)$ and $p(w|x, \theta)$:

$$D(q_\nu(w)||p(w|x, \theta)) = \mathbb{E}_q[\log q_\nu(W)] - \mathbb{E}_q[\log p(W, x|\theta)] + \log p(x|\theta)$$

As in the first application, it is not possible to compute the KL expression due to the last term, so we are going to maximise its lower bound, which is the ELBO function:

$$\mathbb{E}_q[\log p(W, x|\theta)] - \mathbb{E}_q[\log q_\nu(W)]$$

For each latent variable, let us assume that the conditional distribution $p(w_i|w_{-i}, x, \theta)$ is a member of the exponential family:

$$p(w_i|w_{-i}, x, \theta) = h(w_i)exp\{g_i(w_{-i}, x, \theta)^T w_i - a(g_i(w_{-i}, x, \theta))\}$$

is the natural parameter for W_i when conditioning on the remaining latent variables and the observations. Also in this context we are going to work with meanfield variational families, in which each component is expressed as:

$$q_\nu(w) = \prod_{i=1}^{M} \exp\{\nu_i^T w_i - a(w_i)\}$$

where $\nu = \{\nu_1, \nu_2, \ldots, \nu_M\}$ are variational parameters. The optimization of KL divergence with respect to a single variational parameter ν_i is achieved by computing the following expectation:

$$\nu_i = \mathbb{E}_q[g_i(w_{-i}, x, \theta)]$$

In this representation the latent variables are the stick lengths, the atoms, and the cluster assignments:$W = \{V, \nu^*, Z\}$. The hyperparameters are the scaling parameter and the parameter of the conjugate base distribution: $\theta = \{\alpha, \lambda\}$. Following the general idea seen before, we write the variational bound on the log marginal probability of the data:

$$\begin{aligned}\log p(x|\alpha, \lambda) \geq & \mathbb{E}_q[\log p(V|\alpha)] + \mathbb{E}_q[\log p(\eta^*|\lambda)] \\ & + \sum_{n=1}^{N} (\mathbb{E}_q[\log p(Z_n|V)] + \mathbb{E}_q[\log p(x_n|Z_n)]) \\ & - \mathbb{E}_q[\log q(V, \eta^*, Z_n)]\end{aligned}$$

The main difference with respect to the first application is that the components can not be written in the closed form. Then, since the random measure is infinite dimensional, we are going to consider the truncated stick-breaking representation. Thus, we fix the truncation level T and let $q(\nu^T = 1) = 1$; this implies that the mixture proportions $\pi_t(\nu)$ are equal to zero for $t > T$. Note that T is a variational parameter which can be freely set; it is not a part of the prior model specification. The factorized family of variational distributions for meanfield variational inference that we are going to use is exactly this one:

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

where $q_{\gamma_t}(v_t)$ are beta distributions, $q_{\tau_t}(\eta_t^*)$ are exponential family distributions with natural parameters τ_t, and $q_{\phi_n}(z_n)$ are multinomial distributions. We point out that the free variational parameters are $\nu = \{\gamma_1, \ldots, \gamma_{T-1}, \tau_1, \ldots, \tau_T, \phi_1, \ldots, \phi_N\}$.
We aim to develop a mean-field coordinate ascent algorithm. This results in:

$$\begin{aligned}\gamma_{t,1} &= 1 + \sum_n \phi_{n,t} \\ \gamma_{t,2} &= \alpha + \sum_n \sum_{j=t+1}^{T} \phi_{n,j} \\ \varphi_{n,t} &\propto \exp(S_t),\end{aligned}$$

To obtain $\tau_{t,1}$ and $\tau_{t,2}$ we proceeded calculating firstly these two quantities:

$$\begin{aligned}\tau_{t,1}^* &= \lambda_1 + \sum_n \phi_{n,t} X_n \\ \tau_{t,2}^* &= \lambda_2 + \sum_n \phi_{n,t}\end{aligned}$$

And we then performed a normalization to obtain $\tau_{t,1}$ and a reciprocal to get $\tau_{t,2}$:

$$\tau_{t,1} = \frac{\tau^*_{t,1}}{\sum_t \tau^*_{t,2}}$$
$$\tau_{t,2} = \frac{1}{\tau^*_{t,2}}$$

for $t \in \{1, \dots, T\}$ and $n \in \{1, \dots, N\}$, where

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

Practical applications of variational methods must address initialization of the variational distribution. While the algorithm yields a bound for any starting values of the variational parameters, poor choices of initialization can lead to local maxima that yield poor bounds.

3.3 | Application

In our implementation the variational distribution is initialized by incrementally updating the parameters according to a random permutation of the data points. The algorithm is runned multiple times and we choose the final parameter settings that give the best bound on the marginal likelihood.

In this setting we worked with two different models, in the first we considered the parameter α to follow a Normal distribution and the variance of each cluster, to be constant; while in the second one α was defined in order to follow a Normal distribution multiplied to an Inverse Gamma and we considered a prior for the variance of the clusters. In this second model, we look for a posterior distribution not only for the mean, but also for the variance. This differences that distinguish the models imply also a difference in the expression of the ELBO because we have to consider additional quantities to update the variational parameters.

3.3.1 | First model

$$\mu_k \overset{\text{iid}}{\sim} \mathcal{N}_d\left(\mu_0,\ \sigma^2 \mathbb{I}_d\right) \qquad\qquad k = 1, ..., K$$

with K the number of clusters, not known a priori, defined during data generation.

Data model:

$$X_i|\eta, z_i \overset{\text{ind}}{\sim} \mathcal{N}_d\left(\mu_{z_i},\ \mathbb{I}_d\right) \qquad i = 1, ..., N$$

with η such that η_k is equal to the mean and variance for each cluster $k = 1, ..., K$
In this case the variance is fixed.

So the final DP mixture model is:

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

Here there the computations of the ELBO terms for the first model:

$$\begin{aligned} ELBO = &\mathbb{E}_q[\log p(V|\alpha)] + \mathbb{E}_q[\log p(\eta^*|\lambda)] \\ &+ \sum_{n=1}^{N}(\mathbb{E}_q[\log p(Z_n|V)] + \mathbb{E}_q[\log p(x_n|Z_n)]) \\ &- \mathbb{E}_q[\log q(V, \eta^*, Z)] \end{aligned}$$

In the following lines we report the computation of the ELBO. Each term has been evaluated separately in order to have a better understanding.
For the values of λ we have concluded that, given the notation on the paper, $\lambda_1 = 0_d$ and $\lambda_2 = \frac{1}{\sigma^2}$. Moreover, q is the variational ditribution, so we have:

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

where $q_{\gamma_t}(v_t)$ are beta distributions, $q_{\tau_t}(\eta_t^*)$ are normal distributions with parameters τ_t, and $q_{\phi_n}(z_n)$ are multinomial distributions.
Let's define ψ as the digamma function, the full computation is reported in Appendix B.

- First term:

  $$\mathbb{E}_q[\log p(V|\alpha)] = -T\log(B(1,\alpha)) + (\alpha - 1)\sum_{t=1}^{T-1}(\psi(\gamma_{t,2}) - \psi(\gamma_{t,1} + \gamma_{t,2}))$$

  The last sum stops at T-1 because v_T has been force to be 1. The first part has been omitted in the code since it is constant every iteration.

- Second term:

  $$\mathbb{E}_q[\log p(\eta^*|\lambda)] = T\log(\frac{1}{(2\pi\sigma^2)^{d/2}}) - \frac{1}{2\sigma^2}\sum_{t=1}^{T}(d\tau_{t,2} + \tau_{t,1}^T\tau_{t,1})$$

  Again, the first part is constant through the iterations so it will be omitted in the code.

- Third term: it has already been computed in the paper [2], we report the final result.

  $$\mathbb{E}_q[\log p(Z_n|V)] = \sum_{i=1}^{T-1}((\sum_{j=i+1}^{T}\phi_{n,j})(\psi(\gamma_{i,2}) - \psi(\gamma_{i,1} + \gamma_{i,2})) + \phi_{n,i}(\psi(\gamma_{i,1}) - \psi(\gamma_{i,1} + \gamma_{i,2})))$$

  The first sum stops at T-1 for the reason we explained above.

- Fourth term:

$$\mathbb{E}_q[\log p(x_n|Z_n)] = \log(\frac{1}{(2\pi)^{d/2}}) - \frac{1}{2}(x_n^T x_n - 2x_n^T \sum_{t=1}^{T} \phi_{n,t}\tau_{t,1} + \sum_{t=1}^{T} \phi_{n,t}(d\tau_{t,2} + \tau_{t,1}^T \tau_{t,1}))$$

  The first two terms are constant along the iterations so we can omit them.

- Fifth term:

$$\mathbb{E}_q[\log q(V, \eta^*, Z)] = \sum_{t=1}^{T-1} \mathbb{E}_q[\log q_{\gamma_t}(v_t)] + \sum_{t=1}^{T} \mathbb{E}_q[\log q_{\tau_t}(\eta_t^*)] + \sum_{n=1}^{N} \mathbb{E}_q[\log q_{\phi_n}(z_n)]$$

  Where the first term is:

$$\begin{aligned}\mathbb{E}_q[\log q_{\gamma_t}(v_t)] = &-\log(B(\gamma_{t,1}, \gamma_{t,2})) + (\gamma_{t,1} - 1)(\psi(\gamma_{t,1}) - \psi(\gamma_{t,1} + \gamma_{t,2})) + \\ &+ (\gamma_{t,2} - 1)(\psi(\gamma_{t,2}) - \psi(\gamma_{t,1} + \gamma_{t,2}))\end{aligned}$$

  The second therm is:

$$\mathbb{E}_q[\log q_{\tau_t}(\eta_t^*)] = -\frac{d}{2}\log(2\pi) - \frac{d}{2}\log(\tau_{t,2}) - \frac{1}{2}$$

  And the last one is given by:

$$\mathbb{E}_q[\log q_{\phi_n}(z_n)] = \sum_{t=1}^{T} \log(\phi_{n,t})\phi_{n,t}$$

  In the code we omit the constant terms.

Finally, we report the computation of the update of the ϕ according to our model.
In particular the function $a(.)$ in our model, since we are in a multidimensional framework, is given by $a(\eta) = \eta^T \eta$.

$$\phi_{n,t} \propto exp(S_t)$$

Where S_t is given by:

$$\begin{aligned}S_t &= \mathbb{E}_q[\log V_t] + \sum_{i=1}^{t-1} \mathbb{E}_q[\log(1 - V_i)] + \mathbb{E}_q[\eta_t^*]^T x_n - \mathbb{E}_q[\eta_t^{*T} \eta_t^*] \\ &= \psi(\gamma_{t,1}) - \psi(\gamma_{t,1} + \gamma_{t,2}) + \sum_{i=1}^{t-1}(\psi(\gamma_{i,1}) - \psi(\gamma_{i,1} + \gamma_{i,2})) + \tau_{t,1}^T x_n - (\tau_{t,2} + \tau_{t,1}^T \tau_{t,1})\end{aligned}$$

3.3.2 | Second model

We generalized a little bit the previous model by assuming a different variance for the clusters still mantaining the diagonal structure of the variance matrix. We did so adding a parameter for the cluster specific variance and changing the measure α of the DP.

Data model:

$$X_i|\eta, z_i \overset{\text{ind}}{\sim} \mathcal{N}_d\left(\mu_{z_i},\ \tau_{z_i}^2 \mathbb{I}_d\right) \qquad i = 1, ..., N$$

with η such that η_k is equal to the mean and variance for each cluster $k = 1, ..., K$.

So the final DP mixture model is:

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

Regarding the computation of the ELBO, everything is the same as before, apart for the terms that depends on τ^2 and the added quantity $\mathbb{E}_q[\log(q(\tau^2))]$, but it's just a trivial computation.

3.4 | Results

For both model there's one thing to take into consideration, which is that our algorithm doesn't detect clusters which are too small with respect to N. Anyway, since our goal is to approximate the posterior distribution, even if those cluster allocations are not predicted correctly, in our application is enough that the points belonging to them are assigned to a nearby cluster with a very similar mean, which is what happens.

We tested our first algorithm with different number of points, always considering 2 dimentions, 100 iterations and $\alpha = 5$. Note also that the number of clusters in this case is not a priori known.
Using the second model with N=1000, d=2, α=5 we get:

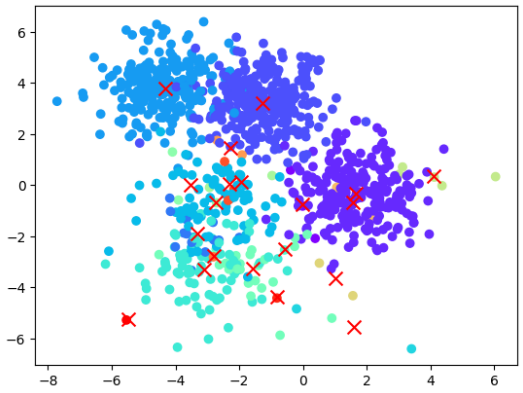


Figure 3.1: Our generated data

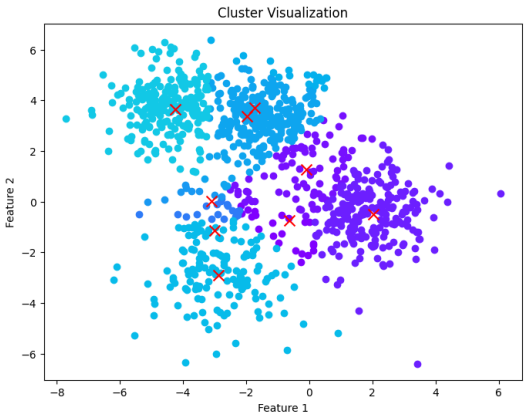


Figure 3.2: Clusters from VI

The clusters we obtain with the first model are almost the same so we reported just one plot.

3.4.1 | First Model

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

3.4.2 | Second Model

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

3.4.3 | Considerations

Notice that the times of the MCMC are very different in the two models, that's because we were able to implement the first model using bayesmix, which is an optimized library for MCMC models. Obtaining times lower even than bayesmix is a very good result since it has been optimized with more advanced tecniques. Instead, the second model was implemented using Stan, which doesn't rely on a very good optimization.
For the second model we did not run the MCMC algorithm using 10000 iterations, because we already saw that the time taken to run 1000 iterations was very high compared to VI. What can we see from these tables is not just that Variational Inferece is clearly faster, but also that the error it makes it's not high enough to justify the usage of MCMC.
Note that we computed the relative error between the distribution error of VI and MCMC. Obtaining a negative value means that the error made by VI is lower that the one made by MCMC.

4 | Indian buffet process

The Indian Buffet Process (IBP) is a nonparametric prior for latent feature models in which observations are influenced by a combination of hidden features. In unsupervised learning, the objective is to discover hidden patterns or features that exist simultaneously within a given set of observations. For example, given images composed of various objects, we may wish to identify the set of unique objects and determine which images contain which objects.
Using non-parametric Bayesian approaches allow us to treat the number of features as a random quantity to be determined as part of the posterior inference procedure.
The most common nonparametric prior for latent feature models is the Indian Buffet Process (IBP). The IBP is a prior on infinite binary matrices that allows us to simultaneously infer which features influence a set of observations and how many features there are. The form of the prior ensures that only a finite number of features will be present in any finite set of observations, but more features may appear as more observations are received.
The implementation of a mean field variational method provide a deterministic alternative to sampling-based approaches. Which have the problem to be less flexible, because they allocate specific values to the feature-assignment variables and also tends to be slow.
Of all the nonparametric Bayesian models studied so far - Gaussian Processes, Dirichlet Processes - the IBP is the most combinatorial and is therefore in the most need of a more efficient inference algorithm. [4]

4.1 | The model

For the IBP, the approximating distribution maintains a separate probability for each feature-observation assignment. Optimising these probability values is also fraught with local optima, but the soft variable assignments give the variational method flexibility lacking in the samplers.
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 from A and Z and uncorrelated across observations.
- The observed data X is given by $X = ZA + \epsilon$

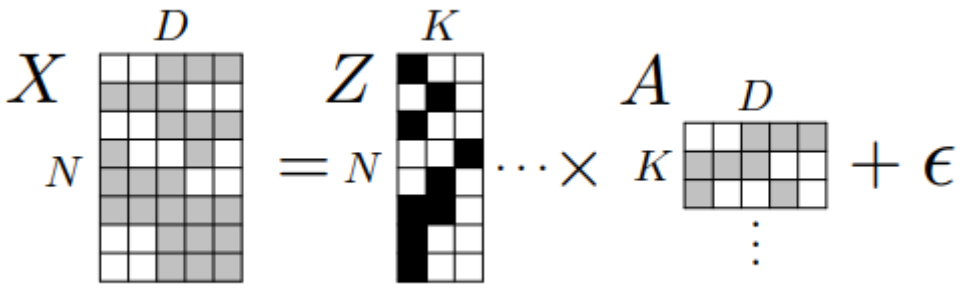


Figure 4.1: The latent feature model proposes the data X is the product of Z and A with some noise.

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. The application will determine the likelihood function $p(X|Z, A)$ and the feature prior $p(A)$. 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.

4.1.1 | The IBP prior

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.

We can introduce a culinary metaphor which represents one way to sample a matrix Z from the prior described in the previous equation.
Imagine the rows of Z correspond to customers and the columns correspond to dishes in an infinitely long (Indian) buffet.

i) The first customer takes the first Poisson(α) dishes;

ii) 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.

Then

$$\begin{cases} z_{nk} = 1 & \text{if customer n tried the } k^{th} \text{ dish} \\ z_{nk} = 0 & \text{otherwise} \end{cases}$$

This process is infinitely exchangeable, which means that the order in which the customers attend the buffet has no impact on the distribution of Z.
While the restaurant construction of the IBP directly lends itself to a Gibbs sampler, we will consider the stick-breaking construction of [7] for the implementation of the variational approach.

4.1.2 | 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

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

The π_k themselves are generated by a stick-breaking process. We first draw a sequence of independent random variables $v_1, v_2, ... \overset{ind}{\sim} \text{Beta}(\alpha, 1)$. 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.

4.2 | Variational Inference setting

We focus on variational inference procedures for the linear-Gaussian likelihood model, as in our reference paper [4], where:

$$\begin{aligned} A_k &\overset{iid}{\sim} N_d(0,\ \sigma_A^2 \mathbb{I}_d) && k = 1, 2, \dots, K \\ \epsilon_n &\overset{iid}{\sim} N_d(0,\ \sigma_X^2 \mathbb{I}_d) && n = 1, 2, \dots, N \end{aligned}$$

Set of hidden variables in the IBP $\mathbf{W} = \{\pi, \mathbf{Z}, \mathbf{A}\}$;
Set of parameters $\theta = \{\alpha, \sigma_A^2, \sigma_X^2\}$.

Since the computation of the true posterior is difficult, 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, which is given by:

$$\arg\max_{\tau,\phi,\nu} \ln p(\mathbf{X} - \theta) - KL(q||p) = \arg\max_{\tau,\phi,\nu} H[q] + \mathbb{E}_q[\ln(p(\mathbf{X}, \mathbf{W}|\theta)]$$

where $H[q]$ is the entropy of distribution q.

In our analysis we'll employ two different methods, which are termed *finite variational approach* and *infinite variational approach* by our reference paper [4]. In both approaches the idea is to apply a truncation level K to the maximum number of features in the variational distribution.

4.2.1 | Finite variational approach

The finite variational approach considers a finite Beta-Bernoulli approximation to the IBP. If we consider a large K, the finite Beta-Bernoulli approximation is equivalent to the IBP.

Finite Beta-Bernoulli model with K features:

$$\begin{aligned} z_{nk}|\pi_k &\overset{ind}{\sim} \text{Bernoulli}(\pi_k) && \forall n = 1, ..., N \\ \pi_k &\overset{iid}{\sim} \text{Beta}(\alpha/K, 1) && \forall k = 1, ..., K \end{aligned}$$

The finite variational approach approximates the true IBP model $p(W, X|\theta)$ with $p_K(W, X|\theta)$ in the previous equation where p_K uses the prior on Z defined by the finite Beta-Bernoulli model. We use a fully factorised variational distribution, where:

$$\begin{aligned} q_{\tau_k}(\pi_k) &= \text{Beta}(\pi_k; \tau_{k1}, \tau_{k2}) && \forall k = 1, ..., K \\ q_{\phi_k}(A_k.) &= N(A_k.; \phi_k, \Phi_k) && \forall k = 1, ..., K \\ q_{\nu_{nk}}(z_{nk}) &= \text{Bernoulli}(z_{nk}; \nu_{nk}) && \forall k = 1, ..., K \text{ and } \forall n = 1, ..., N \end{aligned}$$

In order to obtain the updates we consider:

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

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

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

To retrieve the updates of the parameters we first reconstructed the kernels and then we found their expression. In the following lines we only report the parameter expression since the computation is kind of heavy.

$$\begin{aligned} \tau_{k1} &= \sum_{n=1}^{N} \nu_{nk} + \frac{\alpha}{K}; \\ \tau_{k2} &= N - \sum_{n=1}^{N} \nu_{nk} + 1; \\ \phi_k &= \frac{1}{\sum_{n=1}^{N} \nu_{nk}/\sigma_X^2 + 1/\sigma_A^2} \left(\sum_{n=1}^{N} \nu_{nk} x_n - \sum_{n=1}^{N} \nu_{nk} \sum_{j \neq k} \nu_{nj} \phi_k \right) \frac{1}{\sigma_X^2}; \\ \Phi_k &= \frac{1}{\sum_{n=1}^{N} \nu_{nk}/\sigma_X^2 + 1/\sigma_A^2} \mathbb{I}_d \end{aligned}$$

which is a diagonal matrix, this means we can easily find $\det(\Phi_k)$ and $\mathrm{tr}(\Phi_k)$, quantities that we will use in the computation of the ELBO. Moreover it allows us to define the variational parameters Φ_k as K scalars, knowing that the variance matrix will be $\Phi_k \mathbb{I}_d$, making all the computations simpler and less time and memory consuming.

$$\nu_{nk} = \frac{1}{1 + e^{-\theta}}$$

where

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

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. We split the expectation into terms depending on each of the latent variable:

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

Where H is the entropy of the distribution, i.e. $-\mathbb{E}_q[\log q]$.
Let's consider each term one at a time, the full computation is reported in Appendix C.

- First term:

  $$H[q] = \sum_{k=1}^{K} \left(H(q_{\tau_k}) + H(q_{\phi_k}) + \sum_{n=1}^{N} H(\nu_{nk}) \right)$$
  $$H(q_{\tau_k}) = \ln B(\tau_{k,1}, \tau_{k,2}) - (\tau_{k,1} - 1)\psi(\tau_{k,1}) - (\tau_{k,2} - 1)\psi(\tau_{k,2}) + (\tau_{k,1} + \tau_{k,2} - 2)\psi(\tau_{k,1} + \tau_{k,2})$$
  $$H(q_{\phi_k}) = \frac{d}{2}(1 + \ln 2\pi + \ln \Phi_k)$$
  $$H(q_{\nu_{nk}}) = -\nu_{nk} \log_2 \nu_{nk} - (1 - \nu_{nk}) \log_2 1 - \nu_{nk}$$

- Second term:

  $$\mathbb{E}_{\pi}[\ln p(\pi_k|\alpha)] = -\ln\left(\frac{\alpha}{K}\right) + \left(\frac{\alpha}{K} - 1\right)(\psi(\tau_{k1}) - \psi(\tau_{k1} + \tau_{k2}))$$

  where $\ln \frac{\alpha}{k}$ is constant and won't be included in the code.

- Third term:

  $$\mathbb{E}_{\mathbf{A}}[\ln p(\mathbf{A}_{k\cdot}|\sigma_A^2)] = -\frac{d}{2}\ln(2\pi\sigma_A^2) - \frac{1}{2\sigma_A^2}(d\Phi_k + \phi_k^T \phi_k)$$

  where the first term is constant.

- Fourth term:

  $$\mathbb{E}_{\pi,\mathbf{Z}}[\ln p(z_{nk}|\pi_k)] = \nu_{nk}(\psi(\tau_{k1}) - \psi(\tau_{k1} + \tau_{k2}) + (1 - \nu_{nk})(\psi(\tau_{k2}) - \psi(\tau_{k1} + \tau_{k2})]$$

- Fifth term:

$$\mathbb{E}_{\mathbf{Z},\mathbf{A}}[\ln p(\mathbf{X}_{n.}|\mathbf{Z},\mathbf{A},\sigma_X^2)] =$$
$$= -\frac{d}{2}ln(2\pi\sigma_X^2) - \frac{1}{2\sigma_X^2}\left(X_n^T X_n - 2X_n^T\sum_{k=1}^{K}\nu_{nk}\phi_k + \sum_{k=1}^{K}\sum_{l\neq k}\nu_{nk}\nu_{nl}\phi_k^T\phi_l + \sum_{k=1}^{K}\nu_{nk}(d\Phi_k + \phi_k^T\phi_k)\right)$$

  where the first term and $\frac{1}{2\sigma_X^2}X_n^T X_n$ are constant.

All the expected values are computed with respect to the variational distribution.

4.2.2 | Infinite variational approach

The infinite variational approach uses a truncated version of the stick-breaking construction for the IBP as the approximating variational model q. 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:

$$\pi_k = \begin{cases} \prod_{i=1}^{k} v_i & \text{if } k < K \\ 0 & \text{otherwise} \end{cases}$$

The advantage of using v as our hidden variable is that under the IBP prior, the $\{v_1, ..., v_K\}$ are independent draws from the Beta distribution, 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

$$\begin{aligned} q_{\tau_k}(v_k) &= \text{Beta}(v_k;\tau_{k1},\tau_{k2}) && \forall k = 1,...,K \\ q_{\phi_k}(A_{k.}) &= N(A_{k.};\phi_k,\Phi_k) && \forall k = 1,...,K \\ q_{\nu_{nk}}(z_{nk}) &= \text{Bernoulli}(z_{nk};\nu_{nk}) && \forall k = 1,...,K \text{ and } \forall n = 1,...,N \end{aligned}$$

With respect to the previous case the only update that changes is the one of the ν_{nk} where we add $\mathbb{E}_v[\ln p(z_{nk}|v)]$.
We can now consider the equation of the ELBO, which is very similar to the one of the previous model:

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

The computations are very similar to the ones of the previous model, but for the third therm, that becomes:

$$\mathbb{E}_{v,Z}[\ln p(z_{nk}|v)]$$
$$= \mathbb{E}_{v,Z}[\ln p(z_{nk}=1|v)^{\mathbb{I}(z_{nk}=1)}\ln p(z_{nk}=|v)^{\mathbb{I}(z_{nk}=)}]$$
$$= \nu_{nk}(\sum_{m=1}^{k}\psi(\tau_{m1}) - \psi(\tau_{m1}+\tau_{m2})) + (1-\nu_{nk})\mathbb{E}_v[\ln(1-\prod_{m=1}^{k}v_m)]$$

The last term hasn't a closed form solution. The paper proposes 2 possible approximation: a Taylor expansion or a lower bound given by the introduction of a multinomial distribution. We decided to follow the second approach since it is more flexible with respect to the changing of the number of clusters or data and it follows quite the same update approach we've applied so far.
Hence it is like we have introduced a new variational parameter with its updates for the approximation of the ELBO and of the updates of ν as well, where we have the exact same quantity to approximate.
In particular the lower bound applying the multinomial approach is the following:

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

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

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

Where the proportional is meant to make every q_k a proper distribution and each q_k is a multinomial with parameters 1 and the values found above.

4.3 | Results

We have generated the data by following the model hierarchical definition, then we have applied the algorithms to the generated data.
Firstly, we observed that both algorithms perform well even if the number of features requested is higher then the real one: if more are requested, it tends to allocate the features to the samples, making them more and more negligible.
This is actually not a bad result, it means that you don't have to worry too much about the number of feature you request to the algorithm because the final result won't have a big change. Anyway we need to take into consideration that this would influence the computational cost.
Another important observation is that can happen that more than one feature is allocated to almost every sample. Usually in the data generation this happens with the first one.
To have a better interpretation of the output, it could be useful to obtain the sum of the features allocated to almost every sample, that should be the best approximation you could have of the real first feature.
Apart from the first, we don't have a method for coupling the real and approximated feature, but this is not a problem since our aim is not feature approximation but distribution approximation, which could be found using a weighted sum of the feature using as weights the feature allocation probabilities.
An important observation is that both these algorithms are very sensitive with respect

to the initialization of the parameters: with a wrong formula for initialization, you get trapped in local optima, obtaining unrealistic results.
For example, the feature allocation probabilities are independent probabilities in the variational distribution. If you sample their starting values from an iid uniform distribution you can higher the number of different initialization, changing the seed for sampling, but you won't easily get a good result.
Instead, if you follow the hierarchical model structure, sampling the starting values from an iid beta distribution and computing the cumulative product as for the π_k in the parametrization of the IBP, you will likely find a good local optima that gives you realistic and acceptable results.
We also noticed that changing the hyperparameters of the distribution from which you sample them can change the result.
This happens because, as in every optimization algortihm, there not only the global optima. Moreover, we are making the optimization of a very high number of parameters.
In fact, the number of parameters scales linearly with respect to both the number of features and the number of samples, which can be very high, mainly due to the feature allocation probabilities that alone are NxK parameters.
Taking this into consideration, it is not a surprise to have a lot of local optima. A good suggestion could be to make more initialization as the number of parameters increase.
However, taking into consideration what is said above, the algorithms work fine.

5 | Conclusions

To sum up, we have applied variational inference algorithm to five different models: a Gaussian Mixture Model, two different Dirichlet Processes and two different parametrizations for Indian Buffet Processes.
In every application, we have seen that our code has always been fast, but always less accurate than the employment of MCMC algorithm.
But one important mention that has to be done is that the application of our algorithms wasn't made to find the perfect cluster allocations and the perfect posterior parameter, it was thought to find a good approximation of the distribution function of a sample.
In this framework, we approximate the distribution of a sample with the sum of the cluster assignment probabilities times their respective distributions. In the IBP it could be done as weighted sum on the means with a fixed normal distribution with given variance for our models.
In this particular framework variational inference has shown some good results, good enough to accept the trade off introduced by its application, especially if you need some fast approximation.
What we have experienced to be a 'problem' in the Variational Inference implementation is that every different model has different ELBO, different updates and possibly also different parameters.
While it has been easy to generalize the code with respect to the dimension of the data, it hasn't been the same for the distinct models: you always have to start pretty much from scratch and compute the ELBO and the updates, only at this point you can move to the implementation.
This means that when you have the code it is fast and pretty accurate, but the implementation for different models is quite time-consuming. Nowadays implementing the same models with a MCMC algorithm is quite easier since, for example, with Stan you provide a description of the model and you don't have to perform any additional computation or coding.
So we can see that another trade-off is introduced: while you decide if you want to apply MCMC or variational inference, you have to consider how much you need to use it. If you need it for a high number of runs, then it could be convenient to implement VI and run it, otherwise the time invested in the implementation might outweigh the time saved during each execution.
This could be the real key to decide wether to use VI or MCMC.

Our codes can be found here: https://github.com/angelocurti/Bayesian_Statistics_2324

6 | References

[1] Christopher M. Bishop. *Pattern recognition and machine learning*. New York : Springer, [2006] ©2006, 2006.

[2] David M. Blei and Michael I. Jordan. Variational inference for Dirichlet process mixtures. *Bayesian Analysis*, 1(1):121 – 143, 2006.

[3] David M. Blei, Alp Kucukelbir, and Jon D. McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, apr 2017.

[4] Finale Doshi, Kurt Miller, Jurgen Van Gael, and Yee Whye Teh. Variational inference for the indian buffet process. In David van Dyk and Max Welling, editors, *Proceedings of the Twelth International Conference on Artificial Intelligence and Statistics*, volume 5 of *Proceedings of Machine Learning Research*, pages 137–144, Hilton Clearwater Beach Resort, Clearwater Beach, Florida USA, 16–18 Apr 2009. PMLR.

[5] Ankush Ganguly and Samuel W. F. Earp. An introduction to variational inference, 2021.

[6] Haoliang. Variational Inference in Bayesian Multivariate Gaussian Mixture Model. *Bayesian Analysis*, 2020.

[7] Yee Whye Teh, Dilan Grür, and Zoubin Ghahramani. Stick-breaking construction for the indian buffet process. In Marina Meila and Xiaotong Shen, editors, *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics*, volume 2 of *Proceedings of Machine Learning Research*, pages 556–563, San Juan, Puerto Rico, 21–24 Mar 2007. PMLR.

A | ELBO computation for GMM

- first term:

$$\sum_{k=1}^{K} \mathbb{E}_q[\log p(\mu_k);\ m_k,\ s_k^2] = \sum_{k=1}^{K} \mathbb{E}_{q_\mu,q_c}[\log \frac{1}{(2\pi\sigma^2)^{\frac{d}{2}}} \exp(\frac{-\mu_k^T\mu_k}{2\sigma^2})] =$$
$$= \sum_{k=1}^{K} \mathbb{E}_{q_\mu,q_c}[-\frac{p}{2}\log(2\pi\sigma^2) - \frac{\mu_k^T\mu_k}{2\sigma^2}] = \sum_{k=1}^{K} \mathbb{E}_{q_\mu}[-\frac{\mu_k^T\mu_k}{2\sigma^2}] - \frac{Kd}{2}\log(2\pi\sigma^2) =$$
$$= \frac{1}{2\sigma^2}\sum_{k=1}^{K}(\int_{q_{\mu_k}} (\mu_k^T\mu_k q_{\mu_k} d\mu_k)) - \frac{Kd}{2}\log(2\pi\sigma^2)$$

We assumed $q(\mu_k; m_k, s_k^2) \overset{\text{iid}}{\sim} \mathcal{N}_d(m_k,\ s_k^2\ \mathbb{I}_d)$, then:

$$\sum_{k=1}^{K} \mathbb{E}_q[\log p(\mu_k);\ m_k,\ s_k^2] = -\frac{1}{2\sigma^2}\sum_{k=1}^{K}(ds_k^2 + m_k^T m_k) - \frac{Kd}{2}\log(2\pi\sigma^2)$$

- second term:

$$\sum_{i=1}^{n}(\ \mathbb{E}_q[\log p(c_i);\ \varphi_i]) = \sum_{i=1}^{n}(\ \mathbb{E}_{q_\mu,q_c}[\log \frac{1}{K}]) = -n\log K$$

- third term: since $p(x_i|c_i,\mu) = \sum_{k=1}^{K} p(x_i|\mu_k)^{c_{i,k}}$

$$\sum_{i=1}^{n}(\mathbb{E}_q[\log p(x_i|c_i,\mu)) = \sum_{i=1}^{n}\sum_{k=1}^{K} -\frac{p}{2}\log(2\pi)\varphi_{i,k} -$$
$$-\frac{1}{2}\sum_{i=1}^{n}\sum_{k=1}^{K}\phi_{i,k}(x_i^T x_i - 2x_i^T m_k + ds_k^2 + m_k^T m_k)$$

- fourth term:

$$\sum_{i=1}^{n} \mathbb{E}_q[\log q(c_i;\ \varphi_i)] = \sum_{i=1}^{n}\sum_{k=1}^{K} \varphi_{i,k}\log\varphi_{i,k}$$

- fifth term:

$$\sum_{k=1}^{K} \mathbb{E}_q[\log q(\mu_k;\ m_k,\ s_k^2)] = \sum_{k=1}^{K} \mathbb{E}_q[\log \frac{1}{(2\pi s_k^2)^{\frac{d}{2}}} \exp(\frac{-(\mu_k - m_k)^T(\mu_k - m_k)}{2s_k^2})] =$$
$$= \sum_{k=1}^{K}(-\frac{d}{2}\log(2\pi s_k^2) - \frac{1}{2s_k^2}(\mathbb{E}_q[\mu_k^T\mu_k] - \mathbb{E}_q[\mu_k^T m_k] - \mathbb{E}_q[m_k^T\mu_k] + \mathbb{E}_q[m_k^T m_k])) =$$
$$= \sum_{k=1}^{K} -\frac{d}{2}\log(2\pi s_k^2) - \sum_{k=1}^{K}\frac{1}{2s_k^2}(m_k^T m_k + ds_k^2 - m_k^T m_k - m_k m_k^T + m_k^T m_k) =$$
$$= \sum_{k=1}^{K} -\frac{d}{2}\log(2\pi s_k^2) - \frac{Kd}{2}$$

B | ELBO computation for DMM

In the following computation we'll consider ψ as the digamma function.

- First term:

$$\begin{aligned}\mathbb{E}_q[\log p(V|\alpha)] &= \mathbb{E}_q[\log(\prod_{t=1}^{T}\frac{1}{B(1,\alpha)}(1-v_t)^{\alpha-1})] \\ &= \sum_{t=1}^{T}\mathbb{E}_q[-\log(B(1,\alpha)) + (\alpha-1)\log(1-v_t)] \\ &= -T\log(B(1,\alpha)) + \sum_{t=1}^{T}(\alpha-1)\mathbb{E}_q[\log(1-v_t)] \\ &= -T\log(B(1,\alpha)) + (\alpha-1)\sum_{t=1}^{T-1}(\psi(\gamma_{t,2}) - \psi(\gamma_{t,1}+\gamma_{t,2}))\end{aligned}$$

  The last sum stops at T-1 because v_T has been force to be 1, moreover the paper recalls us that $\mathbb{E}_q[\log(1-v_T)] = 0$.

- Second term:

$$\begin{aligned}\mathbb{E}_q[\log p(\eta^*|\lambda)] &= \mathbb{E}_q[\log\prod_{t=1}^{T}p(\eta_t^*|\lambda)] \\ &= \sum_{t=1}^{T}\mathbb{E}_q[\log(\frac{1}{(2\pi\sigma^2)^{d/2}}e^{-\frac{1}{2\sigma^2}(\eta_t^{*T}\eta_t^*)})] \\ &= T\log(\frac{1}{(2\pi\sigma^2)^{d/2}}) - \frac{1}{2\sigma^2}\sum_{t=1}^{T}\mathbb{E}_q[\eta_t^{*T}\eta_t^*] \\ &= T\log(\frac{1}{(2\pi\sigma^2)^{d/2}}) - \frac{1}{2\sigma^2}\sum_{t=1}^{T}(d\tau_{t,2} + \tau_{t,1}^T\tau_{t,1})\end{aligned}$$

  Again, the first part is constant through the iterations so it will be omitted in the code.

- Third term: it has already been computed in the paper [2], we report the final result.

$$\mathbb{E}_q[\log p(Z_n|V)] = \sum_{i=1}^{T-1}((\sum_{j=i+1}^{T}\phi_{n,j})(\psi(\gamma_{i,2}) - \psi(\gamma_{i,1}+\gamma_{i,2})) + \phi_{n,i}(\psi(\gamma_{i,1}) - \psi(\gamma_{i,1}+\gamma_{i,2})))$$

  Since $v_T = 1$, we have assumed $\mathbb{E}_q[\log v_T]$ equal to 0, even if in the paper doesn't say that explicitly. Moreover the first sum stops at T-1 for the reason we explained above.

- Fourth term:

$$\begin{aligned}\mathbb{E}_q[\log p(x_n|Z_n)] &= \mathbb{E}_q[\log(\frac{1}{(2\pi)^{d/2}}e^{-\frac{1}{2}((x_n-\eta^*_{Z_n})^T(x_n-\eta^*_{Z_n}))})] \\ &= \log(\frac{1}{(2\pi)^{d/2}}) - \frac{1}{2}(x_n^T x_n - 2x_n^T\mathbb{E}_q[\eta^*_{Z_n}] + \mathbb{E}_q[{\eta^*_{Z_n}}^T\eta^*_{Z_n}]) \\ &= \log(\frac{1}{(2\pi)^{d/2}}) - \frac{1}{2}(x_n^T x_n - 2x_n^T\sum_{t=1}^{T}\phi_{n,t}\mathbb{E}_q[\eta^*_t] + \sum_{t=1}^{T}\phi_{n,t}\mathbb{E}_q[{\eta^*_t}^T\eta^*_t]) \\ &= \log(\frac{1}{(2\pi)^{d/2}}) - \frac{1}{2}(x_n^T x_n - 2x_n^T\sum_{t=1}^{T}\phi_{n,t}\tau_{t,1} + \sum_{t=1}^{T}\phi_{n,t}(d\tau_{t,2} + \tau_{t,1}^T\tau_{t,1}))\end{aligned}$$

- Fifth term:

$$\begin{aligned}\mathbb{E}_q[\log q(V,\eta^*,Z)] &= \mathbb{E}_q[\log\prod_{t=1}^{T-1}q_{\gamma_t}(v_t)\prod_{t=1}^{T}q_{\tau_t}(\eta^*_t)\prod_{n=1}^{N}q_{\phi_n}(z_n)] \\ &= \sum_{t=1}^{T-1}\mathbb{E}_q[\log q_{\gamma_t}(v_t)] + \sum_{t=1}^{T}\mathbb{E}_q[\log q_{\tau_t}(\eta^*_t)] + \sum_{n=1}^{N}\mathbb{E}_q[\log q_{\phi_n}(z_n)]\end{aligned}$$

Where the first term is:

$$\begin{aligned}\mathbb{E}_q[\log q_{\gamma_t}(v_t)] &= \mathbb{E}_q[-\log(B(\gamma_{t,1},\gamma_{t,2})) + (\gamma_{t,1}-1)\log v_t + (\gamma_{t,2}-1)\log(1-v_t)] \\ &= -\log(B(\gamma_{t,1},\gamma_{t,2})) + (\gamma_{t,1}-1)(\psi(\gamma_{t,1}) - \psi(\gamma_{t,1}+\gamma_{t,2})) \\ &+ (\gamma_{t,2}-1)(\psi(\gamma_{t,2}) - \psi(\gamma_{t,1}+\gamma_{t,2}))\end{aligned}$$

The second therm is:

$$\begin{aligned}\mathbb{E}_q[\log q_{\tau_t}(\eta^*_t)] &= \mathbb{E}_q[\log(\frac{1}{(2\pi\tau_{t,2})^{d/2}}e^{-\frac{1}{2\tau_{t,2}}((\eta^*_t-\tau_{t,1})^T(\eta^*_t-\tau_{t,1}))})] \\ &= -\frac{d}{2}\log(2\pi) - \frac{d}{2}\log(\tau_{t,2}) - \frac{1}{2\tau_{t,2}}\mathbb{E}_q[(\eta^*_t-\tau_{t,1})^T(\eta^*_t-\tau_{t,1})] \\ &= -\frac{d}{2}\log(2\pi) - \frac{d}{2}\log(\tau_{t,2}) - \frac{1}{2\tau_{t,2}}\tau_{t,2} \\ &= -\frac{d}{2}\log(2\pi) - \frac{d}{2}\log(\tau_{t,2}) - \frac{1}{2}\end{aligned}$$

And the last one is given by:

$$\mathbb{E}_q[\log q_{\phi_n}(z_n)] = \sum_{t=1}^{T}\log(\phi_{n,t})\phi_{n,t}$$

C | ELBO computation for IBP model

Computation for the finite varational approach:

- First term:

$$H[q] = \sum_{k=1}^{K} \left(H(q_{\tau_k}) + H(q_{\phi_k}) + \sum_{n=1}^{N} H(\nu_{nk}) \right)$$
$$H(q_{\tau_k}) = \ln B(\tau_{k,1}, \tau_{k,2}) - (\tau_{k,1} - 1)\psi(\tau_{k,1}) - (\tau_{k,2} - 1)\psi(\tau_{k,2}) + (\tau_{k,1} + \tau_{k,2} - 2)\psi(\tau_{k,1} + \tau_{k,2})$$
$$H(q_{\phi_k}) = \frac{d}{2}(1 + \ln 2\pi + \ln \Phi_k)$$
$$H(q_{\nu_{nk}}) = -\nu_{nk} \log_2 \nu_{nk} - (1 - \nu_{nk}) \log_2 1 - \nu_{nk}$$

- Second term:

$$\mathbb{E}_{\pi}[\ln p(\pi_k|\alpha)] = \mathbb{E}_{\pi}\left[-\ln B\left(\frac{\alpha}{K}, 1\right) + \left(\frac{\alpha}{K} - 1\right) \ln(\pi_k) \right] =$$
$$= -\ln B\left(\frac{\alpha}{K}, 1\right) + \left(\frac{\alpha}{K} - 1\right) (\psi(\tau_{k1}) - \psi(\tau_{k1} + \tau_{k2})) =$$
$$= -\ln\left(\frac{\alpha}{K}\right) + \left(\frac{\alpha}{K} - 1\right) (\psi(\tau_{k1}) - \psi(\tau_{k1} + \tau_{k2}))$$

- Third term:

$$\mathbb{E}_{\mathbf{A}}[\ln p(\mathbf{A}_{k\cdot}|\sigma_A^2)] = \mathbb{E}_{\mathbf{A}}\left[-\frac{d}{2} \ln(2\pi\sigma_A^2) - \frac{1}{2\sigma_A^2}(A_k^T A_k) \right] =$$
$$= -\frac{d}{2} \ln(2\pi\sigma_A^2) - \frac{1}{2\sigma_A^2} \int_{\mathbb{R}^d} \frac{1}{(2\pi\Phi_k)^{d/2}} A_k^T A_k \exp(-\frac{1}{2}(A_k - \phi_k)^T \Phi_k^{-1}(A_k - \phi_k) dA_k =$$
$$= -\frac{d}{2} \ln(2\pi\sigma_A^2) - \frac{1}{2\sigma_A^2}(d\Phi_k + \phi_k^T \phi_k)$$

- Fourth term:

$$\mathbb{E}_{\pi,\mathbf{Z}}[\ln p(z_{nk}|\pi_k)] = \mathbb{E}_{\pi,\mathbf{Z}}\left[\ln\left((\pi_k)^{\mathbb{I}\{z_{nk}=1\}} (1 - \pi_k)^{\mathbb{I}\{z_{nk}=0\}} \right) \right] =$$
$$= \nu_{nk}\mathbb{E}_{\pi_k}[\ln \pi_k] + (1 - \nu_{nk})\mathbb{E}_{\pi_k}[\ln(1 - \pi_k)] =$$
$$= \nu_{nk}(\psi(\tau_{k1}) - \psi(\tau_{k1} + \tau_{k2}) + (1 - \nu_{nk})(\psi(\tau_{k2}) - \psi(\tau_{k1} + \tau_{k2})]$$

- Fifth term:

$$\mathbb{E}_{\mathbf{Z},\mathbf{A}}[\ln p(\mathbf{X}_{n\cdot}|\mathbf{Z}, \mathbf{A}, \sigma_X^2)] = \mathbb{E}_{\mathbf{Z},\mathbf{A}}\left[\ln\left(\frac{1}{(2\pi\sigma_X^2)^{d/2}} \right) \exp\left(-\frac{1}{2\sigma_X^2}(X_n - Z^T A)^T (X_n - Z^T A) \right) \right] =$$
$$= -\frac{d}{2} ln(2\pi\sigma_X^2) - \frac{1}{2\sigma_X^2}\mathbb{E}_{\mathbf{Z},\mathbf{A}}[(X_n - A_{Z_n})^T (X_n - A_{Z_n})] =$$
$$= -\frac{d}{2} ln(2\pi\sigma_X^2) - \frac{1}{2\sigma_X^2}(X_n^T X_n - 2X_n^T \mathbb{E}_{\mathbf{Z},\mathbf{A}}[A_{Z_n}] + \sum + \mathbb{E}_{\mathbf{Z},\mathbf{A}}[(\sum_{k=1}^{K} z_{nk} A_k)^T (\sum_{k=1}^{K} z_{nk} A_k)]) =$$
$$= -\frac{d}{2} ln(2\pi\sigma_X^2) - \frac{1}{2\sigma_X^2}\left(X_n^T X_n - 2X_n^T \sum_{k=1}^{K} \nu_{nk}\phi_k + \sum_{k=1}^{K}\sum_{l\neq k} \nu_{nk}\nu_{nl}\phi_k^T \phi_l + \sum_{k=1}^{K} \nu_{nk}(d\Phi_k + \phi_k^T \phi_k) \right)$$

All the expected values are computed with respect to the variational distribution.