DD2434 - Machine Learning, Advanced Course
Project 1.5 - Variational Inference
Group 3

Tristan Perrot	Étienne Riguet	Romain Darous	Anthony Jones
tristanp@kth.se	riguet@kth.se	darous@kth.se	arjjones@kth.se

December 2023

Abstract

In DD2432 variational inference methods for parametric models were explored. This project discusses and recreates key findings found in *Blei, David M., and Michael I. Jordan. "Variational methods for the Dirichlet process."*, a paper which describes a method of variational inference for the Dirichlet Process. The Dirichlet Process is a distribution on distributions used to describe non-parametric models. Before the findings of this paper, other techniques such as Monte Carlo Markov Chain (MCMC) algorithms like Gibbs sampling were used to approximate the posterior of non-parametric models with a Dirichlet Process Prior, but the paper describes this new method and compares the method with the existing Gibbs sampling method. In this project we recreate the variational inference algorithm described, and test this algorithm on data in the same way as done as in the paper - by simulating a Dirichlet Process Gaussian Mixture model and by using the dataset of a robotic arm. We then compare our results to the ones of the original paper, and discuss and evaluate the paper's approach, methodology, and conclusions.

Tristan Perrot
Étienne Riguet
Romain Darous
Anthony Jones

1 Introduction

The aim of this project is to implement a mean-field variational approach to approximately inference the Gaussian Dirichlet Process mixture model, following the variational method described in *Blei, David M., and Michael I. Jordan. "Variational methods for the Dirichlet process."*.

Throughout the DD2434 course, methods to derive a mean-field variational approach for parametric models, such as the Gaussian mixture model with a known number of clusters, have been explored. The work in this paper extends variational methods to deal with non-parametric statistics. These are models where the number of parameters can grow as the number of data points increases. As with parametric models, a prior probability distribution is used when using non-parametric models. There are various priors used for models of this sort, but the Dirichlet Process is one of most important.

The Dirichlet Process can be understood as a distribution on a distribution. Given a base distribution and a scaling parameter, the Dirichlet Process then outputs a distribution. Observable data in a model can then be obtained by sampling this outputted distribution. The Dirichlet Process Gaussian mixture model is therefore a Gaussian mixture model with *infinitely* many clusters, where the Gaussian distribution supplying a data point is sampled itself from an unknown distribution (the Dirichlet Process).

The posterior distribution of Dirichlet Process mixture models is intractable to compute, and so it is necessary to approximate. Before Blei & Jordan's paper, Markov Chain Monte Carlo algorithms such as the Gibbs sampler existed to approximate the posterior. Their paper reviews this algorithm, as well as describes a variational approach to approximate the posterior. It then implements the Gibbs sampling algorithm, as well as the variational algorithm derived in the paper, on a simulated 2D Dirichlet Process Gaussian Mixture model, as well as on a dataset of simulations on a robot arm. The paper compares and evaluates the two approximation techniques, comparing the amount of time it takes for each algorithm to converge, as well as the log likelihood of further data obtained in the same way, which wasn't used to train the models.

In this project, the aim is to recreate the results of the variational inference method described in the paper. The variational inference algorithm presented in the paper is implemented from scratch. Next, simulated Dirichlet Process Gaussian mixture model data is generated in the same way as in the paper to test the algorithm. The same robot data used in the paper is also used to test. Finally, the results obtained from this project are compared to that of the original paper, and the original paper is evaluated.

2 Methods

In this section, we will discuss our implementation choices regarding the formula provided by the paper. We will use the same notations to make it easier to understand. We will start by applying the CAVI algorithm to a 2D case, as in the paper. However, all our functions and computations are implemented to work for any dimension, as it will be required to perform the held-out likelihood experiment and to apply our model to robot data.

2.1 Sampling data from a Gaussian Dirichlet process

It is assumed in the original paper that the data is generated according to a Gaussian-Gaussian Dirichlet Process :

$$\begin{aligned} G &\sim \mathrm{DP}\left(\alpha, G_0\right) \\ \eta_n &\sim G \\ X_n &\sim p\left(\cdot \mid \eta_n\right). \end{aligned}$$

Thus, the distribution G_0 which governs the sampling of the η^* is a Gaussian, and the data samples X_n will be generated according to a Gaussian distribution with fixed covariance matrix $\sigma^2\mathbb{I}$ and mean η_n.

As all the parameters are not specified in the paper, we chose them arbitrarily :

- G_0 is a Gaussian distribution with mean 0 and covariance matrix $10\mathbb{I}$,
- $\forall n \in \mathbb{N}$, η_n is sampled following the Dirichlet process given in the paper,
- X_n is sampled according to a Gaussian law with mean η_n and fixed covariance matrix $0.3\mathbb{I}$.

Tristan Perrot
Étienne Riguet
Romain Darous
Anthony Jones

We need to sample 100 data points from this Gaussian-Gaussian DP mixture.
An example of sampled dataset in 2D is provided here. The red points corresponds to the η^*.

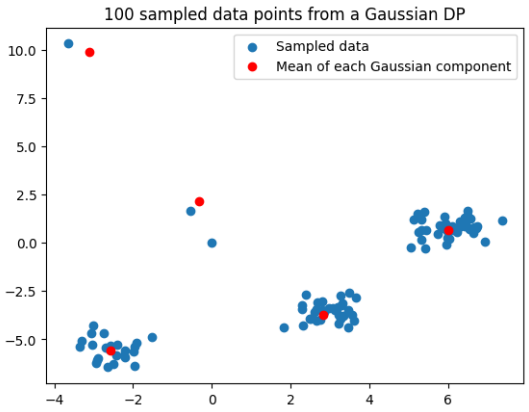

Figure 1: Sampled dataset

2.2 Assumptions for the Variational Inference algorithm

To implement the variational inference DP mixture algorithm, we had to make some additional assumptions :

First, we needed to choose a distribution among the one of the exponential family to get closed formulas for the underlying densities of the model, as well as for the mean-field posterior densities.

2.2.1 Conditional distribution of X_n given Z_n

The conditional distribution is given by this formula in the general case :

$$p\left(x_n \mid z_n, \boldsymbol{\eta}^*\right) = \prod_{i=1}^{K} \left(h\left(x_n\right) \exp\left\{\eta_i^{*T} x_n - a\left(\eta_i^*\right)\right\}\right)^{z_n^i}$$

As the sampled data is known to be generated according to a Gaussian-Gaussian DP mixture, we decided to set every component of the mixture as Gaussian with fixed variance σ^2 as well. Now, we need to find out how the η_i^* relate to the parameters of those Gaussian distributions.

Let f be the density function of a **D**-dimensional Gaussian variable with mean μ and covariance matrix σ^2.

$$\begin{aligned}
f(x) &= \frac{1}{(2\pi\sigma^2)^{D/2}} \exp\left(-\frac{1}{2\sigma^2}(x-\mu)^T(x-\mu)\right) \\
&= \frac{1}{(2\pi\sigma^2)^{D/2}} \exp\left(-\frac{1}{2\sigma^2}(x^T x - 2\mu^T x + \mu^T\mu)\right) \\
&= \frac{\exp\left(-\frac{x^T x}{2\sigma^2}\right)}{(2\pi\sigma^2)^{D/2}} \exp\left(\frac{\mu^T x}{\sigma^2} - \frac{\mu^T\mu}{2\sigma^2}\right) \\
&= h(x) \exp\left(\eta^{*T} x - a(\eta^*)\right)
\end{aligned}$$

where

$$h(x) = \frac{\exp\left(-\frac{x^T x}{2\sigma^2}\right)}{(2\pi\sigma^2)^{D/2}}, \quad \eta^* = \frac{\mu}{\sigma^2}, \quad a(\eta^*) = \frac{\mu^T\mu}{2\sigma^2} = \sigma^2\frac{\eta^{*T}\eta^*}{2}$$

Thus, identifying the parameters reveals that for all i in $\{1, \ldots, K\}$, the i-the component follows a Gaussian distribution with mean $\sigma^2\eta_i^*$ and covariance matrix $\sigma^2\mathbb{I}$.

Tristan Perrot
Étienne Riguet
Romain Darous
Anthony Jones

2.2.2 Conjugate distribution on η_i^* given λ

The distribution is given by this formula in the general case :

$$p\left(\eta^* \mid \lambda\right) = h\left(\eta^*\right) \exp\left\{\lambda_1^T \eta^* + \lambda_2\left(-a\left(\eta^*\right)\right) - a(\lambda)\right\}$$

.

For the same reason as before, we assume this distribution to be Gaussian with mean ν and covariance matrix $s^2\mathbf{I}$. Now, we need to find out how λ_1, λ_2 relate to the parameters of this Gaussian distribution.

Let g be the density function of a $\mathbf{D}$-dimensional Gaussian variable with mean ν and covariance matrix $s^2\mathbb{I}$.

$$\begin{aligned}
g(\eta^*) &= \frac{1}{(2\pi s^2)^{D/2}} \exp\left(-\frac{1}{2s^2}(\eta^* - \nu)^T(\eta^* - \nu)\right) \\
&= \frac{1}{(2\pi s^2)^{D/2}} \exp\left(-\frac{1}{2s^2}(\eta^{*T}\eta^* - 2\nu^T\eta^* + \nu^T\nu)\right) \\
&= \frac{1}{(2\pi s^2)^{D/2}} \exp\left(-\frac{1}{2s^2}\eta^{*T}\eta^* + \frac{\nu^T\eta^*}{s^2} - \frac{\nu^T\nu}{2s^2}\right) \\
&= \frac{1}{(2\pi s^2)^{D/2}} \exp\left(\frac{\nu^T\eta^*}{s^2} - \frac{1}{2s^2}\eta^{*T}\eta^* - \frac{\nu^T\nu}{2s^2}\right) \\
&= h\left(\eta^*\right) \exp\left\{\lambda_1^T \eta^* + \lambda_2\left(-a\left(\eta^*\right)\right) - a(\lambda)\right\}
\end{aligned}$$

where

$$\begin{aligned}
&h(\eta^*) = \frac{1}{(2\pi s^2)^{D/2}}, \quad a(\eta^*) = \frac{\eta^{*T}\eta^*}{2} \text{ (Bernardo \& Smith, 1994)}, \\
&\lambda_1 = \frac{\nu}{s^2}, \quad \lambda_2 = \frac{1}{s^2}, \quad a(\lambda) = \frac{\nu^T\nu}{2s^2} = \frac{\lambda_1^T\lambda_1}{2\lambda_2}
\end{aligned}$$

It follows that η^* follows a Gaussian distribution with mean λ_1/λ_2 and covariance matrix $(1/\lambda_2)\mathbb{I}$.

2.2.3 Variational approximation of the η_i^*

In the variational approximation :

$$q\left(\mathbf{v}, \boldsymbol{\eta}^*, \mathbf{z}, K\right) = \prod_{i=1}^{K} q\left(v_i \mid \gamma_i\right) \prod_{i=1}^{K} q\left(\eta_i^* \mid \tau_i\right) \prod_{n=1}^{N} q\left(z_n \mid \phi_n\right),$$

The η_i^* have the same exponential distribution as the one above, with the parameters $\tau_i = (\tau_{i,1}, \tau_{i,2})$.
Using same identification as before, we also set $q(\eta_i^* \mid \tau_i)$ to be Gaussian distribution with mean τ_{i1}/τ_{i2} and covariance matrix $\frac{1}{\tau_{i,2}}\mathbb{I}$. $\tau_{i,1}$ has the dimension of η_i^* and τ_{i2} is a real number.
Concerning the other random variables, their distributions are already well known and given in the paper.

2.2.4 The role of those close formulas

Those equivalents will be required to estimate the density of the data given the parameters of the variational estimated density and to update the parameters while running the algorithm.

See Section 2.4 to see how is estimated the underlying density of the data given the variational approximation of the latent variable distributions.

Tristan Perrot
Étienne Riguet
Romain Darous
Anthony Jones

2.3 Implementing the CAVI algorithm

Now that all the distributions and the parameters are well defined, we can implement the algorithm. The update rules are given in the original paper. One last thing to do is to set up the **hyperparameters** of the model, which are :

- the fixed variance σ^2 of the data (according to the assumption of the model)
- $\lambda = (\lambda_1, \lambda_2)$, the parameters of the distribution of η^*
- α, the second parameter of the true Beta law of the V_i.

2.3.1 For the sampled DP mixture - 2D case

The hyper parameters of this dataset are easy to set, as we know the underlying structure of the data. Hence, we will have :

- $\sigma^2 = 0.3$
- $\lambda_1 = \mathbb{E}_{G_0}[\eta^*]/\mathrm{Var}_{G_0}[\eta^*] = (0, 0)$
- $\lambda_2 = 1/\mathrm{Var}_{G_0}[\eta^*] = 1/10$

For the dataset provided in the report above, we found that $\alpha = 0.5$ gave relevant results, but the choice is arbitrary and could be made using grid search, for instance.

Following indications of the paper, we set

2.3.2 For the robot data

We load the Robot data from an external file. Thus, we don't have as much information as for the sampled data.

According to the paper, the covariance matrix is set to be the sample covariance, and the mean of the hyper parameter λ is the sample mean.

Hence, we have :

- σ^2 = empirical variance
- λ_1 = empirical mean / empirical variance
- λ_2 = 1 / empirical variance

The mean and variance parameters are computed using only the training data points of the robot data.

After testing with different values, we set $\alpha = 0.1$, which an arbitrary choice giving a good ELBO value.

We also tried our best to have an efficient algorithm, by avoiding loop cascades and using the operations built into the torch and numpy modules.

2.4 Estimating the density of the data

Once the algorithm successfully gave us the parameters for the posterior distribution, we needed to approximate the distribution of the dataset to be able to compare our result to the data.

For that, we used the following formula, provided in the paper :

$$p\left(x_{N+1} \mid \mathbf{z}, \alpha, \lambda\right) = \sum_{i=1}^{K} \mathrm{E}\left[\theta_i \mid \gamma\right] p\left(x_{N+1} \mid \tau_i\right)$$

Tristan Perrot
Étienne Riguet
Romain Darous
Anthony Jones

where $p\left(x_{N+1} \mid \tau_i\right)$ is the i-th component of $p\left(x_n \middle| z_n, \boldsymbol{\eta}^*\right)$ (see Section 2.2.1) where η_i^* is replaced by its mean, computed using the VI algorithm, which is τ_{i1}/τ_{i2}.

Then, we get that $\forall i \in \{1, ..., K\}$,

$$\begin{aligned}\mathrm{E}\left[\theta_i \mid \gamma\right] &= \mathrm{E}\left[V_i\right] \prod_{j=1}^{i-1} \left(1 - \mathrm{E}[V_j]\right) \text{ as the } V_i \text{ are independant} \\ &= \frac{\gamma_{i,1}}{\gamma_{i,1} + \gamma_{i,2}} \prod_{j=1}^{i-1} 1 - \frac{\gamma_{j,1}}{\gamma_{j,1} + \gamma_{j,2}} \text{ as the } V_i \sim \mathrm{Beta}(\gamma_{i,1}, \gamma_{i,2})\end{aligned}$$

and

$$p\left(x_{N+1} \mid \tau_i\right) \sim \mathcal{N}\left(\mathrm{E}[\eta_i^*], \sigma^2 \mathbb{I}\right) \sim \mathcal{N}\left(\frac{\tau_{i,1}}{\tau_{i,2}}, \sigma^2 \mathbb{I}\right)$$

as $\eta_i^* \sim \mathcal{N}\left(\frac{\tau_{i,1}}{\tau_{i,2}}, \frac{1}{\tau_{i,2}} \mathbb{I}\right)$ as computed earlier. This allows us to get an approximated density of the data.

3 Results

There are a number of points to note about the obtained results.

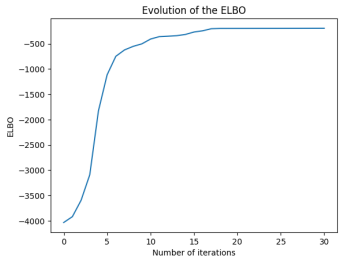

Figure 2: ELBO evolution for the fitting on the simulated data

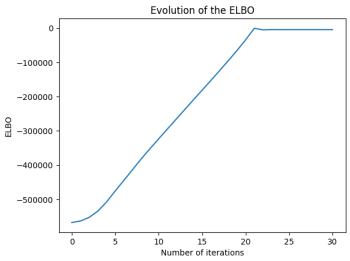

Figure 3: ELBO evolution for the fitting on the robot data

Firstly, we can see that the ELBO converges quickly. It takes approximately 17 iterations to converge to the generated synthetic data as we can see in the figure 2, and approximately 22 iterations to converge to the robot data [3. Running on a personal computer, the model takes about **0.6 seconds to converge** to the synthetic data, and **2 minutes 30 seconds** for the robot data.

Tristan Perrot
Étienne Riguet
Romain Darous
Anthony Jones

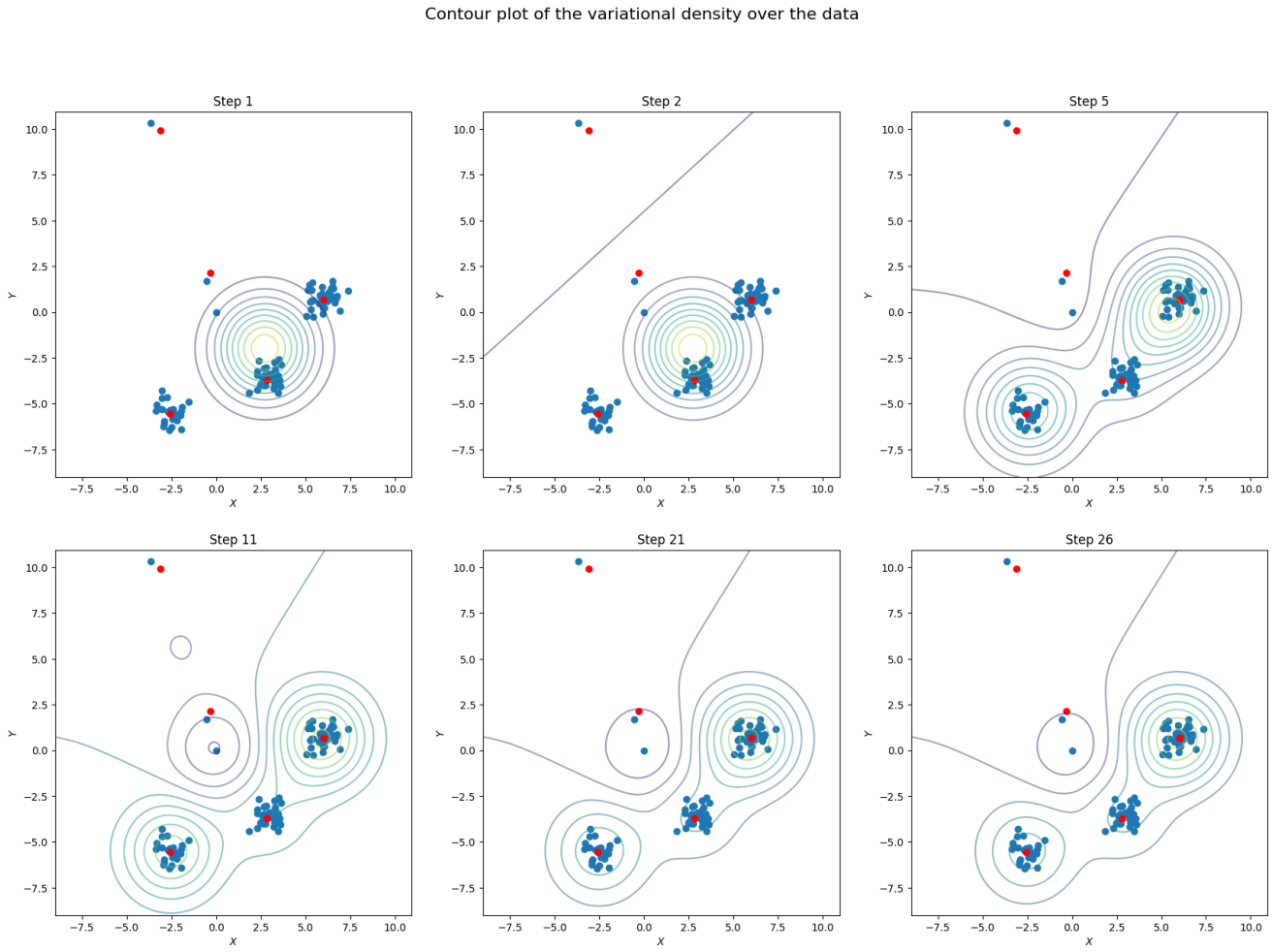

Figure 4: Evolution of the distribution over iterations for the simulated data

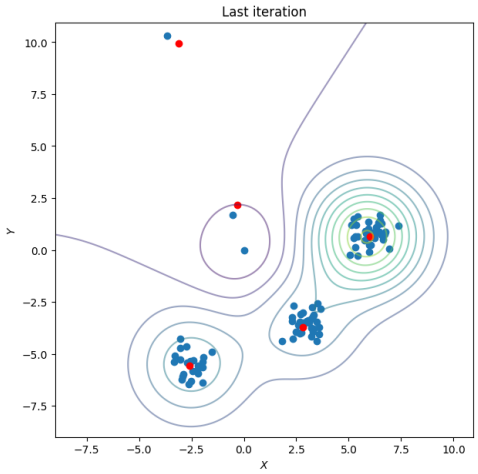

Figure 5: Last iteration result for the simulated data

Tristan Perrot
Étienne Riguet
Romain Darous
Anthony Jones

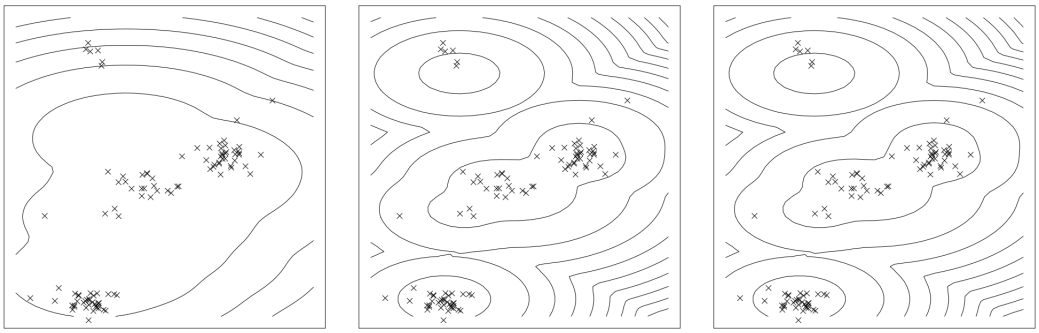

Figure 6: Result obtained in the original paper

Secondly, now on the simulated data, we can see that the algorithm gives good results on the figure 5, similarly to that of the paper (figure 6) but it seems to converge a little bit slower. Indeed, in the paper, they stopped the algorithm to 5 iterations and even if we had good results at 5 iterations we did 30 iterations to make sure that our algorithm was converging. The approximated density seems to fit the data. Indeed, we can see that the algorithm well identifies 2/3 clusters. In areas where the points are less agglomerated, the contours turn to purple, indicating a lower density value, and vice versa. Like in the paper, for the cluster of points that is between two well separated clusters, we can see that the density is combining like it is expected. Moreover, the algorithm predicts two well defined clusters easily identified when we see the dataset. Overall the algorithm works well and allows to get an accurate reproduction of the results obtained in the paper. One last experiment to evaluate its performance is to take a look a the **held-out log-likelihood**. We will also see the evolution of the performances when the dimension increases.

3.1 Held-out log-likelihood on the simulated data

As done in the paper, we did it for data whose the dimension goes from 5 to 50. Thus, we needed to generate data points with higher dimension than 2. For that, we used the function **multivariate_normal** from **numpy.random**, and we adjusted the dimension of the mean and the covariance matrix. We also made sure that our CAVI algorithm could be used for higher dimension than 2.

To build the held-out sets, instead of generating 100 samples, we generated 100 and kept the 100 last samples as held-out set.

An example of held out set is shown below, in two dimensions :

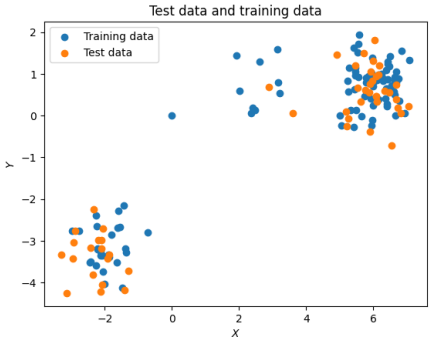

Figure 7: Example of training and testing sets.

The steps done to compute the held-out likelihood and the training time of the model (given the dimension of the input data) are summed up below :

Tristan Perrot
Étienne Riguet
Romain Darous
Anthony Jones

- Generate 200 samples of a Gaussian-Gaussian DP mixture with the same parameters as above (zero mean and fixed variance of 0.3) for the dimensions 5, 10, 20, 30, 40 and 50,
- Use the 100 first samples as a train set and the last 100 samples as a held-out set,
- Perform the CAVI algorithm and compute the estimated density of the data for each dimension, storing the running time,
- Compute the log-density for each point of the test set using the estimated density,
- Get the mean of all the computed log-density to get the log-likelihood of the test set.

We get the following results concerning the running time and the held-out likelihood :

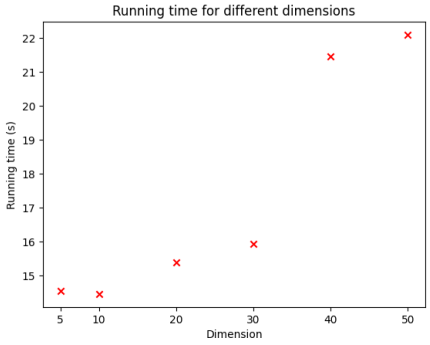

Figure 8: Running time

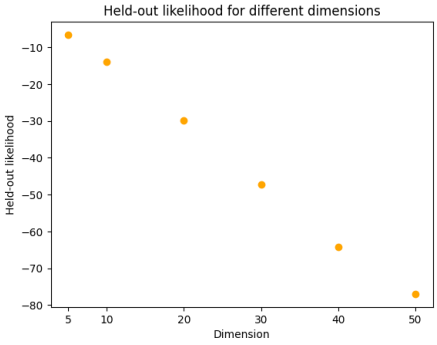

Figure 9: Held-out likelihood

We can observe that the running time tends to increase with the dimension of the input data. However, it barely reaches 30 seconds, even for an input data with 50 dimensions. It confirms the results of the paper, which was showing that the variational method was a lot faster to train than the two other tested methods.
Concerning the held-out likelihood, we observe the same decreasing trend as in the paper with respect to the dimension. The values of the held-out log-likelihood are different though. It seems normal as the computations details of the held-out likelihood are not precised in the paper. It might take into account constants that we don't compute, for example.
Anyway, the evolution shows that the CAVI algorithm performances decrease with the dimension of the data.

3.2 Held-out log-likelihood on the robot data

We could also compute the held-out likelihood on the test set of the robot data, made of 250 samples. We got a held-out likelihood of **-12.7**. It seems like a good result, as it remains in the same order of magnitude as the values obtained for simulated data for dimension 8 (the dimension of the robot data).

3.3 Conclusion

To conclude, we could reproduce similar performance to the original paper, even we had algorithms requiring more iterations to converge. On the robot data, the running time was longer as well, which might be due to the implementation we did, which might be less efficient as the one done in the paper.

4 Discussion

Blei & Jordan's paper introduces variational inference as an effective strategy to overcome the computational complexities in performing exact Bayesian inference within DP models. The approach used in this paper relies on addressing the challenges posed by the infinite-dimensional nature of the DP and the computational demands of

complex Bayesian non-parametric models. Therefore in their approach, the authors deal with the problem of exact inference by using variational inference, which approximates the posterior distribution with a more convenient variational distribution. They work on making this simpler distribution as close as possible to the real one by minimizing the Kullback-Leibler divergence from the true posterior. By doing so, they introduce a computationally feasible method to approximate the posterior distribution over latent variables, such as cluster assignments or mixture parameters, in DP-based models. From the results, we can see that this method provides an excellent trade-off between computational efficiency and accuracy. It allows a model to flexibly choose the form of the approximating variational distribution, thus balancing computational tractability with the closeness of the approximation to the true posterior. This way of working makes it possible to use these models on large sets of information or in complicated situations where it's too complex to model precisely, specifically in situations where exact inference methods might be impractical or infeasible.

To conclude with the approach we can say that by introducing variational inference techniques to approximate the a posteriori distribution in DP-based models, the authors show the possibility of using non-parametric Bayesian statistics It offers a practical way to manage the complexities of these models, making them easier to use in different areas like grouping data, creating mixtures, and estimating densities. This happens by lessening the computer work needed to figure things out exactly in cases where there are endless possibilities, making these models more useful and relevant across various fields.

While the advances made possible by this method cannot be overstated, the paper has a number of flaws that make it difficult to understand and reproduce the results found by the authors; in particular the lack of directly applicable formulas and the abundance of non-explicit parameters and densities. Indeed, throughout the paper, the authors give us many different mathematical formulas, but few of these formulas are directly usable as is, so further research is needed to apply the method computationally. Additionally, the fact that the authors juggle with formulas throughout the article, relying on formulas and explanations stated well in advance of their usefulness, makes it all the more difficult to fully understand the outlined method. The gaps in the specification of experimental parameters make replication of experiments difficult. This imprecision can hamper the validation of results, and give rise to uncertainty as to the robustness of the conclusions drawn from these experiments.

Another problematic aspect is the lack of clarity in the choice of exponential families used. These choices, crucial in probabilistic analysis, lack clear specifications, which can lead to variable interpretations and divergent results. Thus to maximize the practical impact of this document, it could be a good idea to further develop the theoretical explanations with concrete examples, to clarify the choices of exponential families and to specify the experimental parameters exhaustively. This improvement in clarity and specificity would enable a better understanding and more effective use of these methods in real-life applications.

Perhaps further criticism can be made regarding the data used to experiment with the models in the paper. Firstly, when testing with synthetic data, the authors only use 100 data-points which isn't necessarily enough to get a truly representative sample. Perhaps it would be good to increase the number of synthetic data points generated to test the algorithm. Secondly, concerning the robot data, we have a running time comparison and the likelihood value comparison between using the variational inference method and the Gibbs sampling method. However, the authors provide no visualization of the results compared to the actual robot data. The models are compared to each other, but not with the true distribution.

In sum, "Variational Methods for the Dirichlet Process" represents an important milestone in this area of scientific literature, but validation and experimentation with practical value of the paper remains hampered by the lack of specific details and clearly defined parameters, necessitating further efforts to make its methods more accessible and applicable in real-life contexts.