A gentle introduction to Variational Inference

Emilio Tylson Baixauli, Alfons Córdoba Meneses

Jun 2020

1 Introduction

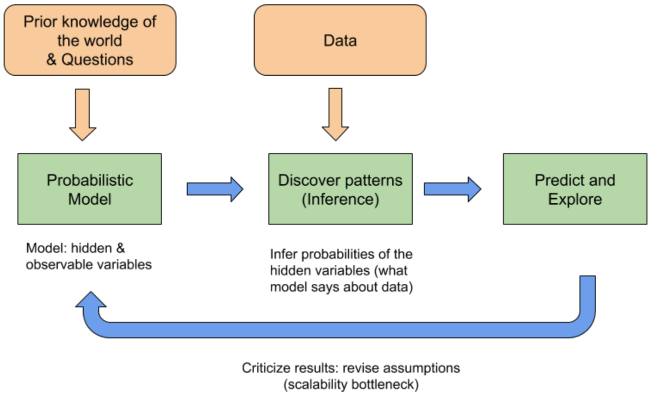
In most of the modern machine learning problems, scientists face the problem of extracting patterns and making predictions based on complex data structures. Due to the advancement of technology, data acquisition techniques allow to gather massive volumes of unstructured data from different sources. This type of information comes in different formats such as graphs,images or text but at the same time are interconnected in a direct or indirect way. Different techniques and methodologies allow to tackle this data in order to make predictions, identify patterns or understand how the internal mechanics that rule the data are, as it happens in causal inference. Probabilistic programming is a discipline from the stack of machine learning techniques that allows to create probabilistic models, and then connect them to the data. Therefore scalability on large volumes of information is a crucial factor, because complex and descriptive models are relatively easy to develop, but are difficult to scale with the data. Inference of posterior probabilities is one of the most important steps on probabilistic programming that can be solved with two main techniques. One technique is MCMC (Markov chain Monte Carlo) that approximates the posterior by sampling from it, making this methods hard to scale for massive volumes of data. The alternative technique is variational inference (VI) that approximates the posterior by optimization procedures.

In this document we provide an introduction to the latter method and an explanation of the fundamentals that make this technique essential for modern machine learning models. Its optimization nature allows to connect this methods with stochastic gradients making the approximation of the posterior easier to scale, in contrast to MCMC that depends on sampling from distributions using the entire dataset.

In the first part we provide a brief explanation of what a probabilistic model is and the probabilistic pipeline. Afterwards we provide a high level intuition of Variational Inference, and an analysis of its components. Then we derive the ELBO (evidence lower bound) function that is the core of VI. Having explained the ELBO we introduce the Gaussian Mixture Model with the intention of explaining the classical approach to solve VI with coordinate ascent. Finally, we close this document by explaining stochastic variational inference that allows to scale this technique to massive volumes of data.

2 Probabilistic Model

The probabilistic pipeline inspired in [3] graphically expresses the general approach to develop a probabilistic model.


Prior knowledge of the world & Questions
Data
Probabilistic Model
Discover patterns (Inference)
Predict and Explore
Model: hidden & observable variables
Infer probabilities of the hidden variables (what model says about data)
Criticize results: revise assumptions (scalability bottleneck)

The pipeline starts with the prior knowledge of the world and initial assumptions regarding the problem to solve. The probabilistic model tries to reflect this knowledge by modeling the crucial aspects into latent and observable variables. Bayesian probability assumes that every variable is either a random variable or data. The main assumptions of the model are centered in random variables that are not observed, latent variables, that captures the patterns that we aim to discover or reveal. Once the model with latent and observable variables is defined, data is used to infer the probability distribution of this latent random variables. This is called **inference**, and basically is the process of obtaining the posterior distribution of the latent variables given the data. Variational inference is a way of performing this latent variable inference by transforming it into a optimization problem. Finally, once the model and data are fitted, prediction and exploration in new data can be done. This allows to criticize the model's assumptions by evaluation, and it also provides new knowledge in order to modify or change the model.

In a general set up, we have $z = z_{1:m}$ latent variables and $x_{1:n}$ observable points. The goal of the probabilistic model is to relate this latent variables with the data. Hence, the probabilistic model can be expressed as the joint probability:

$$p(z_{1:m}, x_{1:n}) = p(z, x) = p(x|z)p(z) \tag{1}$$

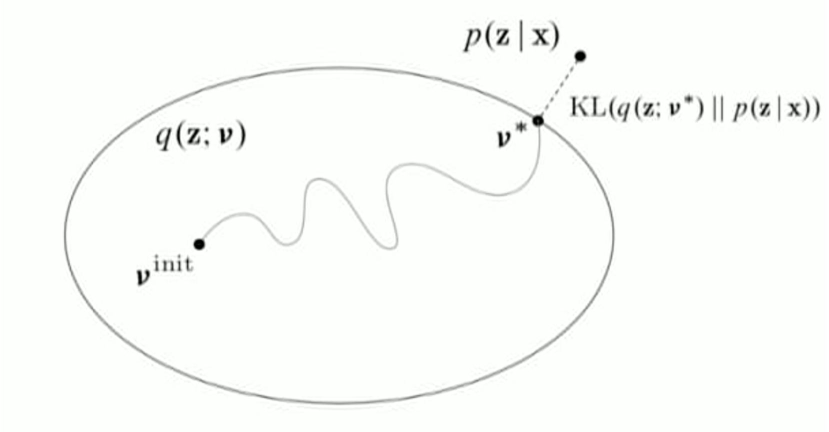
This Bayesian formulation of the model simply proposes a prior distribution for latent variables that are connected to the data by the likelihood $p(x|z)$. Once defined the model, it is required to obtain information of the posterior of the latent variables given the data $p(z|x)$

$$p(z|x) = \frac{p(x,z)}{p(x)} \tag{2}$$

The exact distribution of this probability distribution becomes intractable since the computation of $p(x)$ becomes intractable for complex models that involve several latent variables (marginalize over all discrete and continuous latent variables). For that reason methods such as MCMC or VI are used. VI is faster than MCMC and it is easier to scale for problems with large volume of data. MCMC has been the standard for many years, and despite it was highly studied and improved on sampled methods it is inappropriate for models that handle millions of documents as the case of topic extraction [5] for example.

3 Variational Inference

As mentioned before variational inference tackles the problem of inference of the posterior probability with optimization. The method consists in proposing a family of distributions that we are going to call q and are parametrized by ν, the variational parameters. The objective is to find the closest member of the family through optimal set ν^* that is closest to $p(z|x)$. The closeness is measure by Kullback-Leiber divergence $\mathrm{KL}(q(z,\nu^*)||p(z|x))$. This metric is positive, not symmetric and it measures the similarity between two probability distributions. Hence, the problem is converted into a minimization problem of the Kullabck-Leiber between $q(z;\nu)$ and $p(z|x)$. The following picture based on [3] graphically explains what was explained before.


$p(\mathbf{z}\mid\mathbf{x})$
$\mathrm{KL}(q(\mathbf{z};\boldsymbol{\nu}^*)\mid\mid p(\mathbf{z}\mid\mathbf{x}))$
$q(\mathbf{z};\boldsymbol{\nu})$
$\boldsymbol{\nu}^*$
$\boldsymbol{\nu}^{\text{init}}$

The ellipse represents the family of distributions parametrized by ν, and the path is the optimization procedure to find the optimal ν^* that minimizes $\mathrm{KL}(q(z,\nu^*)||p(z|x))$. Notice that the objective function depends on the $p(z|x)$ that is precisely the value that is intractable and it is unwanted to compute.

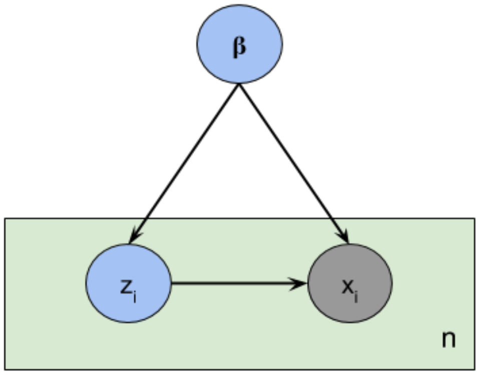
Therefore, instead of minimizing the KL divergence an alternative function called ELBO (evidence lower bound) is maximized. To follow up, we are going to show how this new objective function is related to the KL divergence. Despite it does not represent the exact value of the KL value, optimizing this bound provides a good representation of the probability distribution q.

Following we are going to analyse the particularities and components of the models, in order to derive and better understanding of the objective ELBO.

3.1 Conditional Conjugate Models and Exponential Family

Conditional conjugate models are a class of models that contemplate the most used models. It is to say, more of the well known models can be described as conditional conjugate model. If the model belongs to this class, it can be described its complete conditional expression as an exponential family, bringing a set of properties that will be useful for obtaining the close form of coordinate ascent updates.

The conditional conjugate model, can be summarized with the following graphical model



The principal characteristic of the model is that it is based on global latent variables described by β and local latent variables z_i. Each data point x_i only depends exclusively to the local z_i, the global β and no other local variable. At the same time z_i depends only on the global latent variable β. Hence, the general representation of the model can be written as follows using the chain rule property.

$$p(\beta, z, x) = p(\beta) \prod_{i=1}^{n} p(z_i, x_i|\beta) \tag{3}$$

This class of models is a general representation of well known models such as LDA, Bayesian mixtures , PCA, etc (all these models can be written as a joint of local and global latent variables)

The conditional conjugate structure allows to express the **complete conditional** of each local and global variable as an **exponential family**. The complete conditional is the probability of a latent variable given the others latent variable and the data. Moreover, we assume that this complete conditional belongs to the exponential family. The exponential family is a general way of expressing most of the classical distributions. This family includes the Gaussian, Gamma, Poisson, Bernoulli. The general exponential expression is as follows.

$$p(x) = h(x)\text{exp}\left\{\eta^T t(x) - \alpha(\eta)\right\} \tag{4}$$

Where $h(x)$ is the base density, η is the natural parameter, $t(x)$ are the sufficient statistics and α is the log normalizer, the parameter that ensures the expression integrates to 1.

Having defined the generic exponential distribution, now we define the complete conditional of the conditional conjugate model for each local and global latent variable.

$$p(z_i|\beta, x_i) = h(z_i)\text{exp}\left\{\eta_l(\beta, x_i)^T z_i - \alpha(\eta_l(\beta, x_i)\right\} \tag{5}$$

$$p(\beta|z, x_i) = h(\beta)\text{exp}\left\{\eta_g(z, x_i)^T z_i - \alpha(\eta_g(z, x_i)\right\} \tag{6}$$

The previous expressions are well defined complete conditionals because z_i depends on β and x_i by the natural parameter. And the same happens to β that depends on the other latent variables z_i and the data.

The general intuition for deriving a general coordinate ascent optimization scheme for any probabilistic model is to first define the model. If the model can be expressed as conditional conjugate model then we can derive the complete conditionals for each latent variable given the others and the data. Moreover, this conditionals will belong to the exponential family.

3.2 ELBO function

The main objective is to minimize the KL divergence between $q(z, \beta)$ and $p(z, \beta|x)$. Therefore, if we expand by the definition of Kullback-Leiber we obtain:

$$\text{KL}(q(z, \beta)||p(z, \beta|x)) = \mathbb{E}_q\left[\text{log}q(z, \beta)\right] - \mathbb{E}_q\left[\text{log}p(z, \beta|x))\right] \tag{7}$$

Notice the expectations are taken with respect to q. We expand to the following expansion:

$$\text{KL}(q(z, \beta)||p(z, \beta|x)) = \mathbb{E}_q\left[\log\, q(z, \beta)\right] - \mathbb{E}_q\left[\log\, p(z, \beta, x)) + \log\, p(x))\right] \tag{8}$$

We can isolate $\log(x)$ as it does not depend on q. Hence we can describe KL divergence as following.

$$\text{KL}(q(z, \beta)||p(z, \beta|x)) = \mathbb{E}_q\left[\log\, q(z, \beta)\right] - \mathbb{E}_q\left[\text{log}p(z, \beta, x))\right] + \log\, p(x) \tag{9}$$

As $p(x)$ is constant with respect to q and greater than 0, we define the ELBO(evidence lower bound) function as follows

$$\text{ELBO}(q) = \mathbb{E}_q\left[\log p(z,\beta,x))\right] - \mathbb{E}_q\left[\log\ q(z,\beta)\right] = -\text{KL}(q(z,\beta)||p(z,\beta|x)) + \log\ p(x) \tag{10}$$

ELBO is a lower bound of log of the evidence since the KL divergence is greater than zero by definition. Hence, by maximazing de ELBO we minimize the KL divergence ($p(x)$ constant with respect to q). The final expression of the ELBO is the following. But alternative definitions or derivations can be found in the literature.

$$\text{ELBO}(q) = \mathbb{E}_q\left[\log p(z,\beta,x))\right] - \mathbb{E}_q\left[\log\ q(z,\beta)\right] \tag{11}$$

By carefully analysing the expression we conclude that there is a trade-off. The left-hand term is the expected log complete likelihood with respect to q. In order to maximize this quantity, it needs q to be as similar to the likelihood as possible. Hence, the best q is the one that places all the mass on whichever values that β and z maximize the regularized likelihood, in other words MAP.

On the right hand term, we have the classic negative entropy. We want the entropy to be 0, so the ELBO value can be maximized. Therefore, this terms spreads the mass of q in order to be as flatten as possible (opposite effect to the first term).

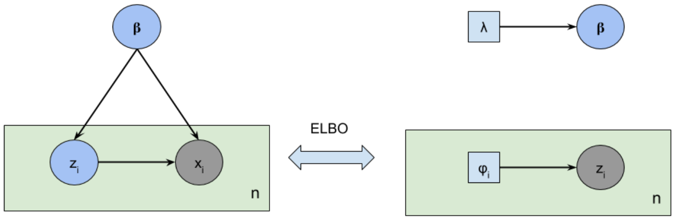
3.3 Mean field family

Until now we have deeply explored the class of probability models and the ELBO objective function. Now we will specify the family class of distributions $q(\beta, z)$, the variational family. The variational family will consist of the mean field family. The crucial characteristic of this family is that every latent random variable is independent and ruled by its own parameters. It is to say that it allows to factorize the joint distribution.

$$q(\beta, z; \lambda, \phi) = q(\beta; \lambda) \prod_i q(z_i; \phi_i) \tag{12}$$

This may seem counter intuitive, but in fact this independence assumption allows us to simplify the solutions and still obtain a near optimal result. In this family we introduce the variational parameters. By optimizing this parameters we will make the distribution q to be as close as possible to the posterior $p(z|x)$. The variational parameters consist of λ that rules the global latent variable β, and local variational parameter ϕ_i that rules each local random latent variable z_i. Each latent variable is governed by its own parameter independent from others.

Notice that in q family x, the data, does not take place. Nevertheless, linking the q distribution with the data is done by optimizing the ELBO. The following image visually represents that process, the optimization algorithm tweaks the local and global parameters of q in order to model the distribution as similar as the original p posterior distribution that depends on data, thus it connects the data with the q.



Furthermore, we can define each factor to belong to the same family as the corresponding complete conditional. Hence, if $p(\beta|z, x)$ is exponential, then the factor $q(\beta, \lambda)$ is exponential.

$$p(\beta|z, x_i) = h(\beta)\text{exp}\left\{\eta_g(z, x_i)^T z_i - \alpha(\eta_g(z, x_i)\right\} \tag{13}$$

Then,

$$q(\beta, \lambda) = h(\beta)\text{exp}\left\{\lambda^T \beta - \alpha(\lambda)\right\} \tag{14}$$

The variational parameter takes the function of the natural parameter of the exponential family. Same happens to the factor of local variables, if each complete conditional $p(z_i|\beta, x_i)$ belongs to exponential, then $q(z_i : \phi_i)$ belongs to exponential with ϕ_i natural parameter.

In other words, in order to summarize all the things that we have shown until now: suppose we have a model with complete conditionals that are a Gaussian and the other a Poisson. The mean field family provides a factorized model, with two random variables that are independent, and one is a Gaussian and the other a Poisson with their independent variational parameters. Optimizing the ELBO, modifies the variational parametrs so the particular realization of the factors minimize the KL divergence with respect to the posterior with the data. Having defined the framework of this variational inference scheme, we can define the ELBO with respect to the variational parameters.

$$\text{ELBO}(\lambda, \phi) = \mathbb{E}_q\left[\log\, p(z, \beta, x); \lambda, \phi\right] - \mathbb{E}_q\left[\log\, q(z, \beta); \lambda, \phi\right] \tag{15}$$

3.4 General Coordinate Ascent

This simple but not scalable algorithm is meant to iteratively update each variational factor by holding the others variational variables fixed, similar to the Gibbs sampling idea. The general coordinate update has the following form

$$\lambda^* = \alpha + \sum_{i=1}^{n} \mathbb{E}_{\phi_i}\left[\eta_g(z, x)\right] \tag{16}$$

$$\phi_i^* = \mathbb{E}_{\lambda}\left[\eta_l(\beta, x_i)\right] \tag{17}$$

The optimization of each variable is the expectation of the natural parameter of its complete conditional. And the same time the natural parameter is a function of the other latent variables and the observations. Hence the expectation depends on all the other variational parameters that is not the one being updated. Moreover, we have postulated that the variational parameter and its corresponding η are in the same space. The algorithm is straight forward. It receives as input the data and the model p. It randomly initializes the variational parameters and then it starts by first updating the local variational paramters with respect to the others. After the local variational parameters are updated then it updates the global variational parameter using the local ones.

Algorithm 1 Coordinate Ascent (CAVI)

```
procedure CAVI(p($\beta, z, x$), $x$)
    $\lambda \leftarrow$ Initialize
    $\phi_i \leftarrow$ Initialize
    while ELBO convergence do
        for all each data point i do
            $\phi_i^* \leftarrow \mathbb{E}_{\lambda}\left[\eta_l(\beta, x_i)\right]$        ▷ Update local variational parameter
        end for
        $\lambda^* \leftarrow \alpha + \sum_{i=1}^{n} \mathbb{E}_{\phi_i}\left[\eta_g(z, x)\right]$        ▷ Update global variational parameter
    end while
end procedure
```

As mentioned in the beginning of the document, this coordinate ascent algorithm does not scale on data. The starting initialization of the global variable is crucial. Imagine the global variable is initialized randomly and we have 10 millons of observations. Only when all local parameters for each of those 10 millions documents were updated, the global parameters can be updated based on the whole corpus of observation. Therefore, the process for updating global variables takes longer as the size of the observation increase. So most of locals variables updates based on global ones are meaningless because they are based on the global variables that are slower to update. Improvements on the way of global initialization has been performed, such as initializing them with k-menas, but the main bottleneck is still present, global parameters takes longer as data size increases.

4 Gaussian Mixture Variational Inference Example

We are going to further review variational inference with an example. In this case we derive the classic coordinate ascent method for a Gaussian Mixture model, and we will see that it corresponds to the general description of the algorithm provided before. This model is classic example because it contains all the characteristics that we have mentioned before. First it contains latent variables that is of our interest since they are local for each point and also it contains global. Then it is a conditional conjugate model, so it can be approximated with a mean field family q.

4.1 Model definition

We have n data points $x = x_{1:n}$ that are sampled from one of the k Gaussian distribution with μ_1 to μ_k. In order to describe this process in a probabilistic model we introduce for each data point x_i a selector variable c_i that consist of a k-dimensional vectors that has a value of 1 in the position of the latent Gaussian where the x_i was sampled and contains 0 in all of the other positions (one-hot vector encoder). Therefore we have just defined μ_i and c_i as latent variables. Then we assume that μ_i is sampled form $\mathcal{N}(0, \sigma^2)$ and c_i from a Categorical(π) distribution. [2] propose to set π and σ^2 as latent variables, but following [4] in order to ease the derivations we set those as hyper-parameters (particularly π a vector of $\frac{1}{k}$ hence the categorical distribution is a uniform over k discrete values). Therefore, the probabilistic model, the joint distribution is conjugate conditional structure.

$$p(\mu, c, x) = p(\mu) \prod_{i=1}^{n} p(c_i) p(x_i | c_i, \mu) \tag{18}$$

Then we define the mean field family q and its variational parameters as following:

$$q(\mu, c) = q(\mu; m, s^2) q(c, \phi) = \prod_{i=1}^{K} q(\mu_i; m_i, s_i^2) \prod_{i=1}^{n} q(c_i; \phi_i) \tag{19}$$

Notice that in the q family the distribution is factorized and each variable is independent form the others. Variational parameters are m_i, s_i that rule mu_i and ϕ_i that rule c_i

4.2 Coordinate Ascent

Then we describe the ELBO. In order to derive the update rule we start in a generic form and then we do the same in the context of Gaussian Mixture model.

First assume that all latent variables are named z. And z is composed by z_i. Then again we have:

$$\text{ELBO}(q) = \mathbb{E}_q \left[\log p(z, x))\right] - \mathbb{E}_q \left[\log\ q(z)\right] \tag{20}$$

As $q(z)$ can be factorized in $\prod z_i$. And also the product inside of a logarithm can be transformed to a sum logarithms then we have:

$$\text{ELBO}(q) = \mathbb{E}_q \left[\log p(z_j, z_{-j}, \beta, x))\right] - \sum_{q_i} \mathbb{E}_{q_i} \left[\log\ q_j(z_j)\right] \tag{21}$$

If we express the ELBO only with respect all j then the rest of variables can be described as a constant with respect to j ($-j$ means all indices that are not j)

$$\text{ELBO}(q) = \mathbb{E}_j \left[\mathbb{E}_{-j} \left[\log p(z_j, z_{-j}, x|z_j))\right]\right] - \mathbb{E}_{q_j} \left[\log\ q_j(z_j)\right] + \text{const} \tag{22}$$

As z_j and z_{-j} are independent:

$$\text{ELBO}(q) = \mathbb{E}_j \left[\mathbb{E}_{-j} \left[\log p(z, x))\right]\right] - \mathbb{E}_{q_j} \left[\log\ q_j(z_j)\right] + \text{const} \tag{23}$$

The previous expression can be viewed as the KL divergence between q_j and $\mathbb{E}_{-j} \left[\log p(z, x))\right]$. Therefore, the optimal solution is:

$$\log\ q_j(z_j) \propto \mathbb{E}_{-j} \left[\log p(z, x))\right] \tag{24}$$

Now we will do the same reasoning with the Gaussian Mixture variables. It is to say, we will express the ELBO with respect to one variational parameter and leave the others as constant.

$$\text{ELBO}(m, s^2, \phi) = \mathbb{E}\left[\log\ p(\mu, c, x); m, s^2, \phi\right] - \mathbb{E}\left[\log\ q(\mu, c; m, s^2, \phi)\right] \tag{25}$$

By replacing the definitions of p and q (each term contains the dependence to the respective variational parameter)

$$\begin{aligned}\text{ELBO}(m, s^2, \phi) = \sum_{k=1}^{K} \mathbb{E}\left[\log\ p(\mu_k); m_k, s_k^2\right] \\ + \sum_{i=1}^{n} \left(\mathbb{E}\left[\log\ p(c_i); \phi_i\right] \mathbb{E}\left[p(x_i|c_i, \mu); m, s^2, \phi\right]\right) \\ - \sum_{k=1}^{K} \mathbb{E}\left[\log\ q(\mu_k; m_k, s_k^2)\right] \\ - \sum_{i=1}^{n} \mathbb{E}\left[\log\ q(c_i; \phi_i)\right]\end{aligned} \tag{26}$$

By using the results of eq.(35) we derive the following expression:

$$\log\ q_j(c_j; \phi_j) \propto \log\ p(c_j) + \mathbb{E}_{-j}\left[\log p(x_j|c_j, \mu); m, s^2)\right] \tag{27}$$

By isolating q_j

$$q_j(c_j;\phi_j) \propto \exp\ \left\{\log\, p(c_j) + \mathbb{E}_{-j}\left[\log p(x_j|c_j,\mu);m,s^2)\right]\right\} \tag{28}$$

The exponent is the joint distribution for variable c_j. Recall that c_j is an indicator variable sampled form a categorical(discrete uniform over K values). Therefore, we can express

$$\log\, p(c_i) = \log\, \frac{1}{K} = -\log K \tag{29}$$

On the other hand ,we have the expectation term that is expressed with respect to μ (m and s^2 variational parameters). It is to say it is the expected log of the j the Gaussian density. Then we have:

$$p(x_j|c_j,\mu) = \prod_{i=1}^{K} p(x_j|\mu_i)^{c_{jk}} \tag{30}$$

Replacing the expression in the expectation term we obtain

$$\begin{aligned}\mathbb{E}_{-j}\left[\log p(x_j|c_j,\mu);m,s^2)\right] &= \sum_{i=1}^{K} c_{jk}\, \mathbb{E}\left[\log\, p(x_j|\mu_i);m_i,s_i^2\right] \\ &= \sum_{i=1}^{K} c_{jk}\, \mathbb{E}\left[\frac{-(x_j-\mu_i)^2}{2};m_i,s_i^2\right] + \text{const} \\ = \sum_{i=1}^{K} c_{jk}\, \mathbb{E}\left[\mathbb{E}\left[\mu_i;m_i,s_i^2\right] - \frac{\mathbb{E}\left[\mu_i^2;m_i,s_i^2\right]}{2}\right] &+ \text{const}\end{aligned} \tag{31}$$

Notice that all variables that do not depend on c_j are encapsulated in "**const**". Then, the variational parameter update step for ϕ_{jk} can be computed by:

$$\phi_{jk} \propto \exp\left\{\mathbb{E}\left[\mu_i;m_i,s_i^2\right]x_i - \frac{\mathbb{E}\left[\mu_i^2;m_i,s_i^2\right]}{2}\right\} \tag{32}$$

The expectations of the ith Gaussian mixture can be computed as:

$$\mathbb{E}\left[\mu_i;m_i,s_i^2\right] = m_i \tag{33}$$

$$\mathbb{E}\left[\mu_i^2;m_i,s_i^2\right] = s_i^2 + m_i^2 \tag{34}$$

Then we can write the update step as:

$$\phi_{jk} \propto \exp\left\{m_i x_i - \frac{m_i^2+s_i^2}{2}\right\} \tag{35}$$

Then we are going to derive the m and s update rule. Recalling eq.(35), we have

$$q(\mu_k) \propto \exp \left\{ \log\, p(\mu_k) + \sum_{i=1}^{n} \mathbb{E}\left[\log\, p(x_i|c_i, \mu); \phi_i, m_{-k}, s^2_{-k}\right] \right\} \tag{36}$$

Taking into account that $\mathbb{E}\left[c_{ik}; \phi_{ik}\right] = \phi_{ik}$ and c_i is a selector variable:

$$\begin{aligned}
\log\, q(\mu_k) = \log\, p(\mu_k) + \sum_{i=1}^{n} \mathbb{E}\left[\log\, p(x_i|c_i, \mu); \phi_i, m_{-k}, s^2_{-k}\right] \\
= \frac{-\mu_k}{2\sigma^2} + \sum_{i=1}^{n} \mathbb{E}\left[c_{ik} \log\, p(x_i|\mu_k); \phi_i\right] + \text{const} \\
= \frac{-\mu_k}{2\sigma^2} + \sum_{i=1}^{n} \mathbb{E}\left[c_{ik}; \phi_i\right] \log\, p(x_i|\mu_k) + \text{const} \\
= \frac{-\mu_k}{2\sigma^2} + \sum_{i=1}^{n} + \phi_{ik} \frac{-(x_i - \mu_k)^2}{2} + \text{const} \\
= \frac{-\mu_k}{2\sigma^2} + \sum_{i=1}^{n} \phi_{ik} x_i \mu_k - \frac{\phi_{ik}\mu_k^2}{2} + \text{const} \\
= (\sum_{i=1}^{n} \phi_{ik} x_i)\mu_k - (\frac{1}{2\sigma^2}) + \sum i = 1^n \frac{\phi_{ik}}{2})\mu_k^2 + \text{const}
\end{aligned} \tag{37}$$

Then, expressing everything with exponential:

$$q(\mu_k) = \exp \left\{ (\sum_{i=1}^{n} \phi_{ik} x_i)\mu_k - (\frac{1}{2\sigma^2}) + \sum i = 1^n \frac{\phi_{ik}}{2})\mu_k^2 + \text{const} \right\} \tag{38}$$

Therefore, eq.(38) belongs to an exponential family. Recalling eq.(4) that describes the exponential family, we conclude that the update step for $q(\mu_k)$ is an exponential family with sufficient statistics $\left\{\mu_k, \mu_k^2\right\}$ and natural parameters $\left\{\sum_{i=1}^{n} \phi_{ik} x_i, \frac{1}{2\sigma^2}) + \sum i = 1^n \frac{\phi_{ik}}{2}\right\}$

Then expressing the variational parameters update step we obtain:

$$m_k = \frac{\sum_i \phi_{i,k} x_i}{\frac{1}{\sigma^2} + \sum_i \phi_{ik}} \tag{39}$$

$$s_k^2 = \frac{1}{\frac{1}{\sigma^2} + \sum_i \phi_{ik}} \tag{40}$$

Intuitively we can conclude that this update is a posterior update of the variables. Then mean of the Gaussian cluster is average of the points weighted by the responsibility variable of belonging to that cluster.

Algorithm 2 Coordinate Ascent for Gaussian Mixture(CAVI)

```
procedure CAVI(data x, number of K clusters)
    $\lambda \leftarrow$ Initialize
    $\phi_{i,k} \leftarrow$ Initialize
    while ELBO convergence do
        for all each data point i do
            $\phi^*_{ik} \leftarrow \exp\left\{m_i x_i - \frac{m_i^2+s_i^2}{2}\right\}$        ▷ Update local variational parameter
        end for
        for all each data K cluster do
            $m_k \leftarrow \frac{\sum_i \phi_{i,k} x_i}{\frac{1}{\sigma^2}+\sum_i \phi_{ik}}$
            $s_k^2 \leftarrow \frac{1}{\frac{1}{\sigma^2}+\sum_i \phi_{ik}}$
        end for                                  ▷ Update global variational parameter
    end while
end procedure
```

5 Stochastic Variational Inference

Stochastic Variational Inference is an improvement over the classical coordinate ascent update. Before, it was explained that this classical approach starts by initializing the global variables with some strategy(could be random initialization, or k-means for example, but both are far from a local optimal). Then optimize each local variable for each data point. Only by going through all the data points the global variables can be updated again. Not only the global parameters take longer as the data size increases, each local variable is updated with the non-updated version of the global parameter. In order to maximize the ELBO or any loss or cost function we can use a gradient ascent method over the variational parameters. Suppose our variational parameter is ν. This iterative process requires to follow the direction of the gradient in order to update ν. Hence, the update rule will be:

$$\nu_{t+1} = \nu_t + \rho_t \nabla_\nu \text{ELBO} \tag{41}$$

Stochastic gradient introduced by Herbert Robbinsand Sutton Monro [7] is a way of generating cheaper noisy versions of the gradient that are used to optimize the variational parameters. Combined to the gradient ascent presented before and under certain conditions this gradient updates with noisy gradients are guarantee to converge to a local optimum. This methods is one of the cornerstones of modern machine learning, as allows to obtain approximations of the gradient involving less data, helping models to scale with massive corpus of observations.

Stochastic gradient is referenced as follows $\hat{\nabla}\mathcal{L}$. Two conditions must be fulfilled in order for gradient ascent with stochastic gradient to converge. The first condition is that the stochastic gradient is unbiased, this is to say the expectation of the stochastic gradient equals to the true gradient. This provides the key intuition that this cheaper and noisy stochastic gradient converges is that the expectation is equal to the true gradient.

$$\mathbb{E}\left[\hat{\nabla}_{\nu}\text{ELBO}\right] = \nabla_{\nu}\text{ELBO} \tag{42}$$

The second condition is that the update step ρ follows a decreasing criterion (such as Robbins-Monro rule, Momentum or Armijo rule). Having this two conditions covered the update rule is guaranteed to converge to local optimum.

Conditional conjugate models, as it is the case of variational inference provide a close form for the natural gradient (a special type of gradients that preserve geometrical structure [1]) of the global latent variable(that we call it λ) [6]. Hence, the natural gradient of the ELBO with respect to λ is the follows:

$$\nabla_{\lambda}^{\text{nat}} ELBO(\lambda) = (\alpha + \sum_{i=1}^{n} \mathbb{E}_{\phi_i^*}\left[t(z_i, x_i)\right]) - \lambda \tag{43}$$

By observing carefully the previous expression, we can conclude that the natural gradient of the global variational parameters equals to the coordinate ascent update rule minus the current value of the global parameter. Therefore, if we want to compute a cheaper and noisy gradient of the ELBO, we can randomly select a point (or a random set of points, minibatch) from the data instead of using all the points, and compute the stochastic version of the natural gradient as follows:

$$j \sim \text{Uniform}(1, ...n) \tag{44}$$

$$\nabla_{\lambda}^{\text{nat}} ELBO(\lambda) = (\alpha + n\,\mathbb{E}_{\phi_j^*}\left[t(z_j, x_j)\right]) - \lambda \tag{45}$$

Now we can see that using this update rule, the global variational parameter is optimized with one or a few data points, agilizing the update of the variables of the whole model. It is important to remember that the expectation of this noisy gradient that depends in a few points is equal to the gradient that uses the total of observations (unbiased).

The stochastic variational inference algorithm will be as follows.: Sampling the data, updating the local variables then updating the global variation variables with respect to how the selected observation refelects the information.

Algorithm 3 Stochastic VI

```
procedure StochasticVI(p($\beta, z, x$), $x$)
    $\lambda \leftarrow$ Initialize
    $\phi_i \leftarrow$ Initialize
    while ELBO convergence do
        $i \sim$ Uniform($1, ...n$)
        $\phi_i^* \leftarrow \mathbb{E}_\lambda \left[\eta_l(\beta, x_i)\right]$        ▷ Update local variational parameter
        $\hat{\lambda} \leftarrow \alpha + n\,\mathbb{E}_{\phi_i} \left[\eta_g(z, x)\right]$        ▷ Update global intermediate variational parameter
        $\lambda = (1 - \rho_t)\lambda + \rho_t\hat{\lambda}$
    end while
end procedure
```

6 Implementation and Results

We have already explained the basics of variational inference as well as its application for the case of Gaussian mixtures. For the sake of simplicity of exposition we have explained it with the approach took by David Blei. However, we have not simplified the problem when coding it and we have adopted the [2] approach for this very same problem. The code uses the notation of [2] and it references some equations there.

Using the approach from [2] it uses new parameters because it puts a prior on the precision matrix (the inverse of the covariance matrix) of each of the gaussians. Concretely it makes the mean and precision matrix of the gaussian follow a Gaussian-Wishart distribution. Introducing these priors and new parameters changes the equations but the idea is fundamentally the same. See bishop section 10.2 Illustration: Variational Mixture of Gaussians for more details.

To test the algorithms we create data-sets with gaussians of same number of points and therefore the mixture coefficients are homogeneous.

In order to initialize the gaussians inferred we use two methods. Firstly we use the one proposed by David Blei that consists in sampling the centers from a gaussian the center of which is the average point of the whole dataset and the covariance is the covariance of the whole dataset too. Using this we can see the results in Fig. 1. We are showing there the program ran using the same number of gaussians distributions that there were. Secondly, we have used the k-means algorithm to initialize the centers of the gaussians. This algorithm converges pretty fast and in general makes the ELBO converge faster. There are cases like when the true gaussians have centers too close together and similar covariances that it might not be better than using the first method of initialization. We show an example in Fig. 3.

The number of gaussians initialized does not need to be the same number of underlying gaussians. The algorithm finds the underlying number of gaussians by making the mixture coefficient of the not needed gaussians indistinguisable from zero as we see in Fig. 2 where there are only 2 gaussians but we have ini-

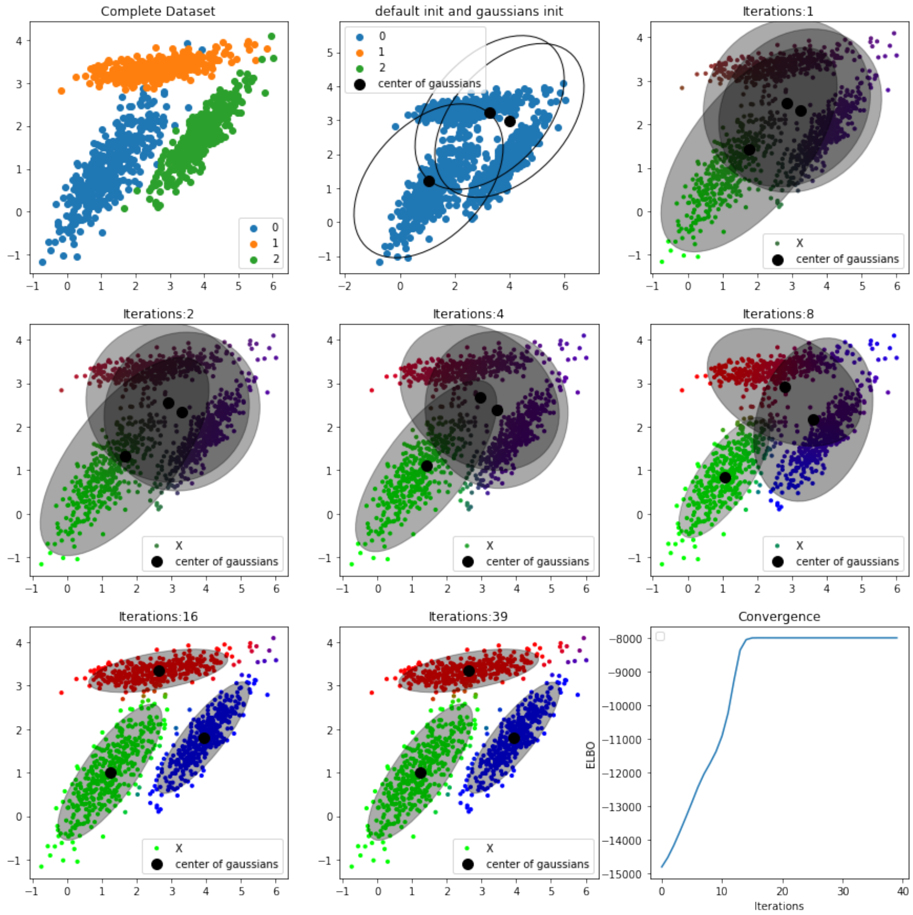


Figure 1: Plots containining the different stages of the program. We can first see the complete dataset obtained at random sampling from 3 gaussians. Then we can see the initial 3 centroids generated at random sampled from a gaussian based on all the points. Afterwards we see various iterations. The colors encode the responsibilities and therefore the probability of each point to belong to each of the gaussians using RGB. The ellipses show the centers of the gaussians, their dimensions correspond to 2 times the standard deviation and the opacity encodes the mixture coefficient of said gaussian. And the last plot is the lower bound or ELBO that we are maximizing.

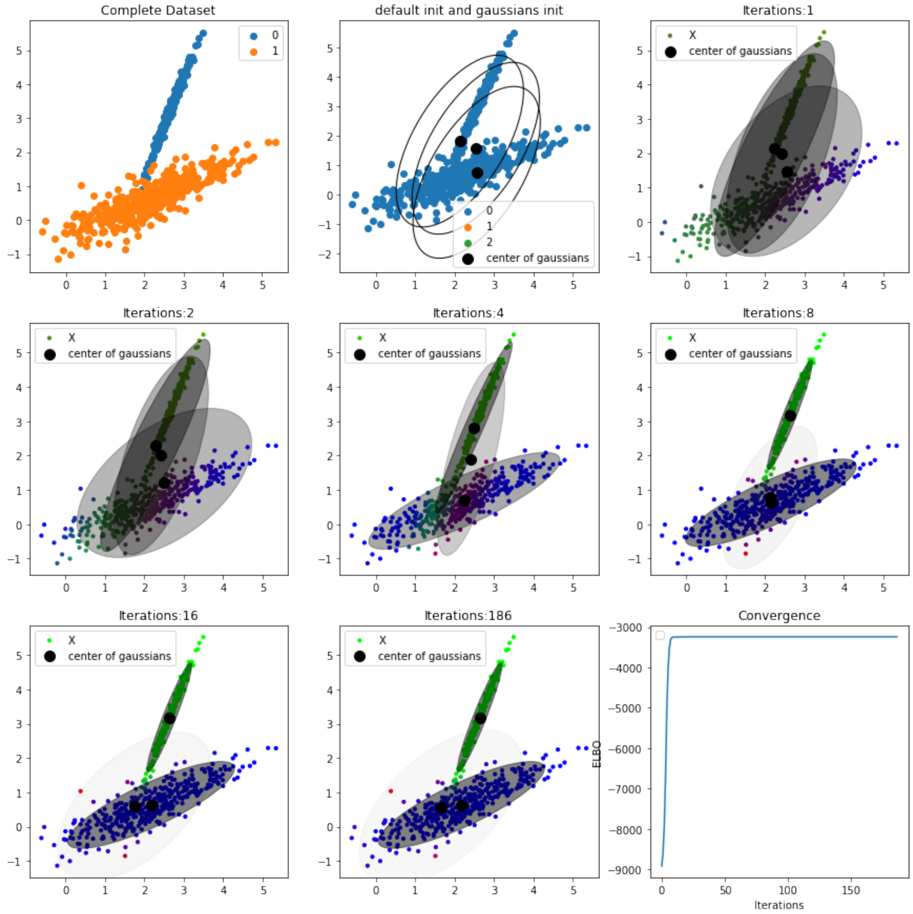


Figure 2: Plots containing the different stages of the program. We can first see the complete dataset obtained at random sampling from 2 gaussians. Then we can see the initial 3 centroids generated at random sampled from a gaussian based on all the points. Afterwards we see various iterations. The colors encode the responsibilities and therefore the probability of each point to belong to each of the gaussians using RGB. The ellipses show the centers of the gaussians, their dimensions correspond to 2 times the standard deviation and the opacity encodes the mixture coefficient of said gaussian. And the last plot is the lower bound or ELBO that we are maximizing.

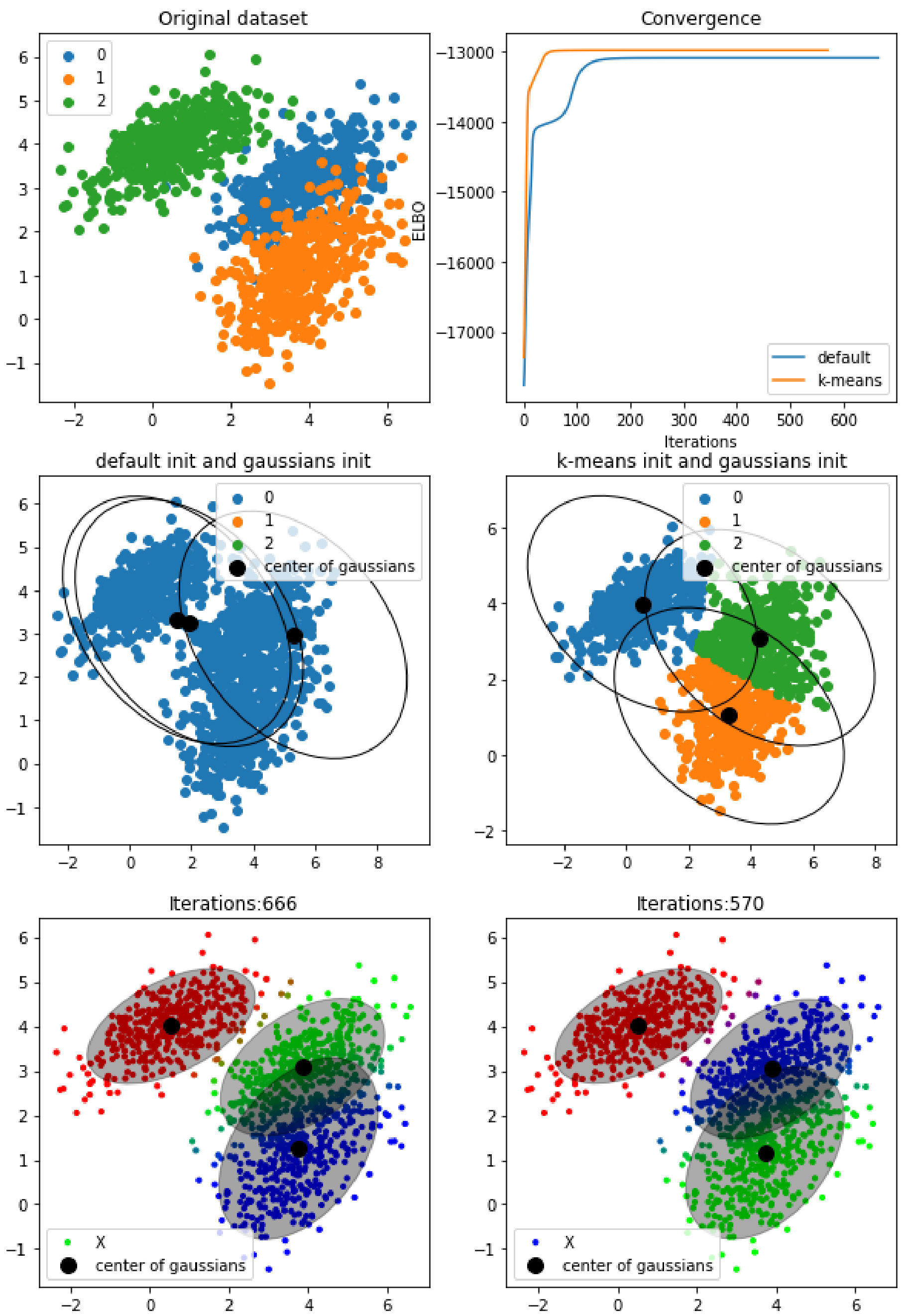


Figure 3: Plots comparing initializing methods: gaussian or default one and k-means one. We can first see the complete dataset obtained at random sampling from 3 gaussians. Then we can see the initial 3 centroids generated at random sampled from a gaussian based on all the points. Afterwards, we see he lower bound or ELBO that we are maximizing for each method. Below we see the initial gaussians whose centers are precisely calculated or chosen according to the initialization mode or method. In the k-means initialization we show the subsets that this algorithm has computed. In the last row we show the last iteration result of the gaussians. The colors encode the responsibilities and therefore the probability of each point to belong to each of the gaussians using RGB. The ellipses show the centers of the gaussians, their dimensions correspond to 2 times the standard deviation and the opacity encodes the mixture coefficient of said gaussian.

tialized the algorithm with 3. This makes this method not need cross-validation to find the optimal number of gaussians.

To end we have tried to implement Stochastic Gradient Ascent, concretely our code can run mini-batch gradient descent with the number of batches that can go from 1 (mono-batch or vanilla gradient ascent) to N, where N is the number of points of the dataset and so SGA. The results show that monobatch is the best method in terms of convergence and time per iteration. SGA isn't faster because the most computationally expensive step is the E-step or the calculation of the responsibilities in it, to be more specific. And it is done independently of the size of the dataset.

7 Conclusions

In this document we started first by reviewing the key componenets of probabilistic model and the importance of inference. Basically inference of latent variables is a key problem that can be tackled by different methodologies. Sampling methods like MCMC have been heavily studied and different versions have been developed in order to improve the sampling from posterior densities. Moreover we have seen that this type of techniques does not scale with large volumes of data. Variational Inferences is gaining momentum in the probabilistic programming field since allows to approximate the posterior densities. Describing the model in a vast class of models called conditional conjugate, and approximating with a simpler family of models called mean field class, variational inference method can be described in a generic way. So based on the generic derivations, we have shown how this method adapts to a simple Gaussian Mixtures. The objective of this explanation was to give a general intuition to the reader, so they can distinguish different components of the variational inference: the model, the objective function and the optimization algorithm. Starting form this intuition the idea is to promote further reading of the latest studies and not get lost in variational inference basics.

Coordinate ascent method is not scalable, after initializing all the global variational parameters, the method goes until all the data points and its local variables until it updates the global parameters. That makes the method to get slower as the data grows. Stochastic variational inference, based on the stochastic gradients helps the model to scale to massive data since it speed the global parameters updates. The model only subsample the data points from the observation, evaluate how those points reflects the global structure, and then the global parameters are updated. This strategy helps the model to massive volumes of data.

This document does not cover black box variational inference. This methods that covers all the things that we have seen helps to speed up the creation and evaluation of the models, since it does not need all the mathematical work that is behind of each model. It relies on autograd features and other mathematical tricks that helps to develop deeper and complex model without the worries of doing the derivatives by hand.

Many authors suggest that there are plenty of areas of study for variational inference, since it is less studied than others methods. Certainly variational inference opens the application of probabilistic models and bayesian statistics to the modern world of massive volume of data analysis.

References

[1] Shun-Ichi Amari. Natural gradient works efficiently in learning. *Neural computation*, 10(2):251–276, 1998.

[2] Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.

[3] David Blei. Black box variational inference. `https://www.youtube.com/watch?v=-H2N4tVDK7I&list=PL_PW0E_Tf2qvXBEpl1OY39RULTN-ExzZQ&index=10&t=0s`, Nov 2018.

[4] 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, 2017.

[5] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[6] Matthew D Hoffman, David M Blei, Chong Wang, and John Paisley. Stochastic variational inference. *The Journal of Machine Learning Research*, 14(1):1303–1347, 2013.

[7] Herbert Robbins and Sutton Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951.