Bayesian inference problem, MCMC and variational inference

Joseph Rocca
Baptiste Rocca

November 30, 2019

Abstract

Overview of the Bayesian inference problem in statistics.

1 Introduction

Bayesian inference is a major problem in statistics that is also encountered in many machine learning methods. For example, Gaussian mixture models, for classification, or Latent Dirichlet Allocation, for topic modelling, are both graphical models requiring to solve such a problem when fitting the data.

Meanwhile, it can be noticed that Bayesian inference problems can sometimes be very difficult to solve depending on the model settings (assumptions, dimensionality, ...). In large problems, exact solutions require, indeed, heavy computations that often become intractable and some approximation techniques have to be used to overcome this issue and build fast and scalable systems.

In this post we will discuss the two main methods that can be used to tackle the Bayesian inference problem: Markov Chain Monte Carlo (MCMC), that is a sampling based approach, and Variational Inference (VI), that is an approximation based approach.

Outline. In the first section we will discuss the Bayesian inference problem and see some examples of classical machine learning applications in which this problem naturally appears. Then in the second section we will present globally MCMC technique to solve this problem and give some details about two MCMC algorithms: Metropolis-Hasting and Gibbs Sampling. Finally in the third section we will introduce Variational Inference and see how an approximate solution can be obtained following an optimisation process over a parametrised family of distributions.

Note. The subsection marked by a (*) are pretty mathematical and can be skipped without hurting the global understanding of this post. Notice also that in this post p(.) is used to denote either probability, probability density or probability distribution depending on the context.

2 The Bayesian inference problem

In this section we present the Bayesian inference problem and discuss some computational difficulties before giving the example of Latent Dirichlet Allocation, a concrete machine learning technique of topic modelling in which this problem is encountered.

2.1 What is inference?

Statistical inference consists in learning about what we do not observe based on what we observe. In other words, it is the process of drawing conclusions such as punctual estimations, confidence intervals or distribution estimations about some latent variables (often causes) in a population, based on some observed variables (often effects) in this population or in a sample of this population.

In particular, Bayesian inference is the process of producing statistical inference taking a Bayesian point of view. In short, the Bayesian paradigm is a statistical/probabilistic paradigm in which a prior knowledge, modelled by a probability distribution, is updated each time a new observation, whose uncertainty is modelled by another probability distribution, is recorded. The whole idea idea that rules the Bayesian paradigm is embed in the so called Bayes theorem that expresses the relation between the updated knowledge (the "posterior"), the prior knowledge (the "prior") and the knowledge coming from the observation (the "likelihood").

A classical example is the Bayesian inference of parameters. Let's assume a model where data x are generated from a probability distribution depending on an unknown parameter θ. Let's also assume that we have a prior knowledge about the parameter θ that can be expressed as a probability distribution $p(\theta)$. Then, when data x are observed, we can update the prior knowledge about this parameter using the Bayes theorem as follows

2.2 Computational difficulties

The Bayes theorem tells us that the computation of the posterior requires three terms: a prior, a likelihood and an evidence. The first two can be expressed easily as they are part of the assumed model (in many situation, the prior and the likelihood are explicitly known). However, the third term, that is a normalisation factor, requires to be computed such that

$$p(x) = \int_{\theta} p(x|\theta)p(\theta)d\theta \tag{1}$$

Although in low dimension this integral can be computed without too much difficulties, it can become intractable in higher dimensions. In this last case, the exact computation of the posterior distribution is practically infeasible and some approximation techniques have to be used to get solutions to problems that require to know this posterior (such as mean computation, for example).

We can notice that some other computational difficulties can arise from Bayesian inference problem such as, for example, combinatorics problems when some variables are discrete. Among the approaches that are the most used to overcome these difficulties we find Markov Chain Monte Carlo and Variational Inference methods. Later in this post, we will describe

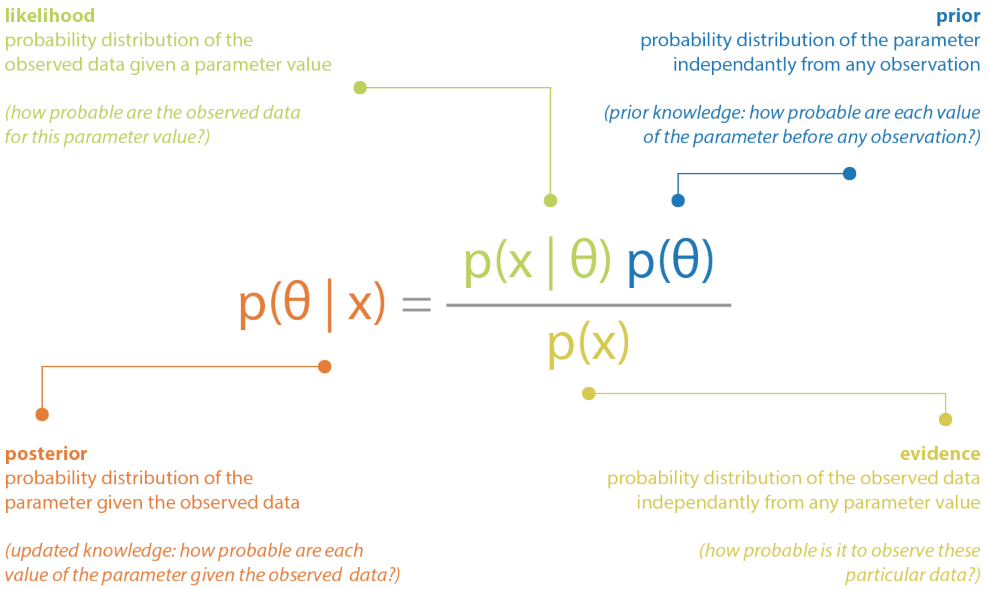

Figure 1: Illustration of the Bayes theorem applied to the inference of a parameter given observed data.

these two approaches focusing especially on the "normalisation factor problem" but one should keep in mind that these methods can also be precious when facing other computational difficulties related to Bayesian inference.

In order to make things a lit bit more general for the upcoming sections, we can observe that, as x is supposed to be given and can, so, be teated as a parameter, we face a situation where we have a probability distribution on θ defined up to a normalisation factor

$$\pi_x(\theta) \equiv p(\theta|x) \propto p(x|\theta)p(\theta) \equiv g_x(\theta) \tag{2}$$

Before describing MCMC and VI in the next two sections, let's give a concrete example of Bayesian inference problem in machine learning with Latent Dirichlet Allocation.

2.3 Example

Bayesian inference problem naturally appears, for example, in machine learning methods that assume a probabilistic graphical model and where, given some observations, we want to recover latent variables of the model. In topic modelling, the Latent Dirichlet Allocation (LDA) method defines such a model for the description of texts in a corpus. Thus, given the full corpus vocabulary of size V and a given number of topics T, the model assumes:

- there exists, for each topic, a "topic-word" probability distribution over the vocabulary (with a Dirichlet prior assumed)
- there exists, for each document, a "document-topic" probability distribution over the topics (with another Dirichlet prior assumed)

- each word in a document have been sampled such that, first, we have sampled a topic from the "document-topic" distribution of the document and, second, we have sampled a word from the "topic-word" distribution attached to the sampled topic

The purpose of the method, whose name comes from the Dirichlet priors assumed in the model, is then to infer the latent topics in the observed corpus as well as the topic decomposition of each documents. Even if we won't dive into details of LDA, we can say very roughly, denoting w the vector of words in the corpus and z the vector of topics associated to these words, that we want to infer z based on the observed w in a Bayesian way:

$$p(z|w) = \frac{p(w|z)p(z)}{p(w)} = \frac{p(w|z)p(z)}{\int_z p(w|z)p(z)dz} \tag{3}$$

Here, beyond the fact that the normalisation factor is absolutely intractable due to a huge dimensionality, we face a combinatoric challenge (as some variables of the problem are discrete) that require to use either MCMC or VI to get an approximate solution. The reader interested by topic modelling and its specific underlying Bayesian inference problem can take a look at this reference paper on LDA.

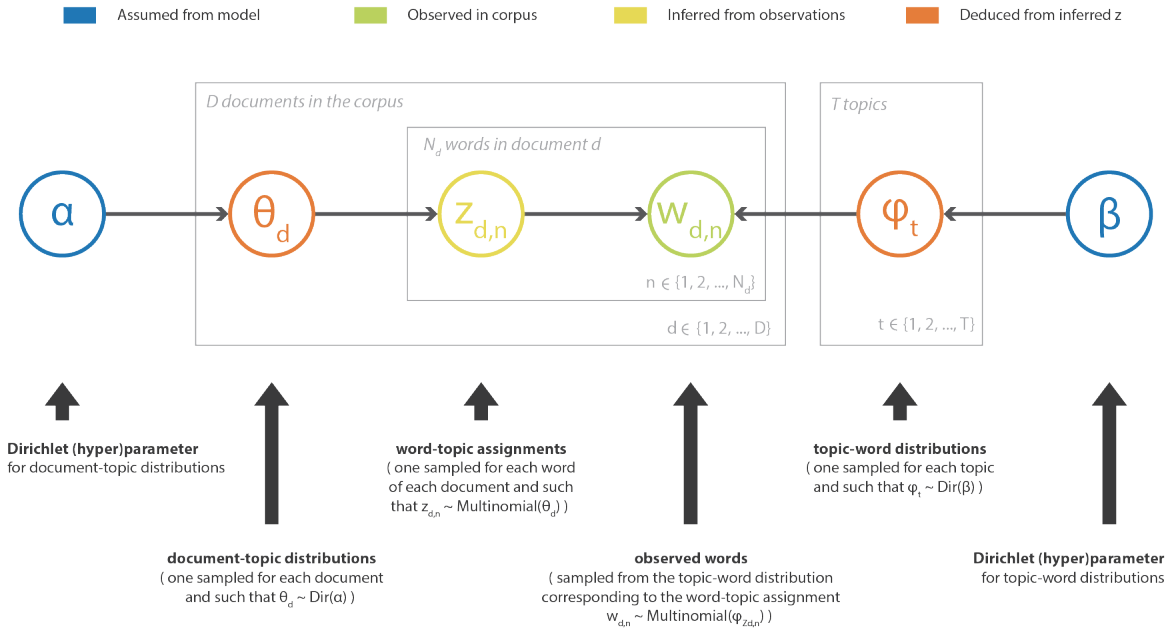

Figure 2: Illustration of the Latent Dirichlet Allocation method.

3 Markov Chains Monte Carlo (MCMC)

As we mentioned before, one of the main difficulty faced when dealing with a Bayesian inference problem comes from the normalisation factor. In this section we describe MCMC sampling methods that constitute a possible solution to overcome this issue as well as some others computational difficulties related to Bayesian inference.

3.1 The sampling approach

The idea of sampling methods is the following. Let's assume first that we have a way (MCMC) to draw samples from a probability distribution defined up to a factor. Then, instead of trying to deal with intractable computations involving the posterior, we can get samples from this distribution (using only the not normalised part definition) and use these samples to compute various punctual statistics such as mean and variance or even to approximate the distribution by Kernel Density Estimation.

Contrarily to VI methods described in the next section, MCMC approaches assume no model for the studied probability distribution (the posterior in the Bayesian inference case). As a consequence, these methods have a low bias but a high variance and it implies that results are most of the time more costly to obtain but also more accurate than the one we can get from VI.

To conclude this subsection, we outline once more the fact that this sampling process we just described is not constrained to the Bayesian inference of posterior distribution and can also, more generally, be used in any situation where a probability distribution is defined up to its normalisation factor.

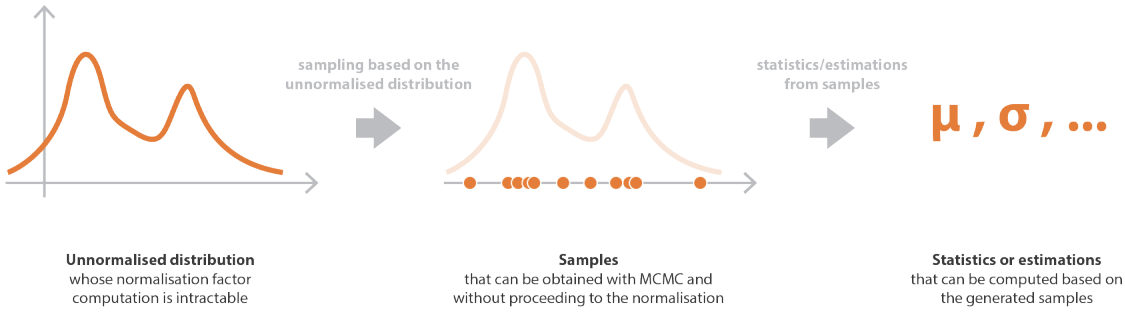

Figure 3: Illustration of the sampling approach (MCMC).

3.2 The idea of MCMC

In statistics, Markov Chain Monte Carlo algorithms are aimed at generating samples from a given probability distribution. The "Monte Carlo" part of the method's name is due to the sampling purpose whereas the "Markov Chain" part comes from the way we obtain these samples (we refer the reader to our introductory post on Markov Chains).

In order to produce samples, the idea is to set up a Markov Chain whose stationary distribution is the one we want to sample from. Then, we can simulate a random sequence of states from that Markov Chain that is long enough to (almost) reach the steady state and then keep some generated states as our samples.

Among the random variables generation techniques, MCMC is a pretty advanced kind of methods (we already discussed an other method in our post about GANs) that makes possible to get samples from a very difficult probability distribution potentially defined only

up to a multiplicative constant. The counter-intuitive fact that we can obtain, with MCMC, samples from a distribution not well normalised comes from the specific way we define the Markov Chain that is insensitive to these normalisation factor.

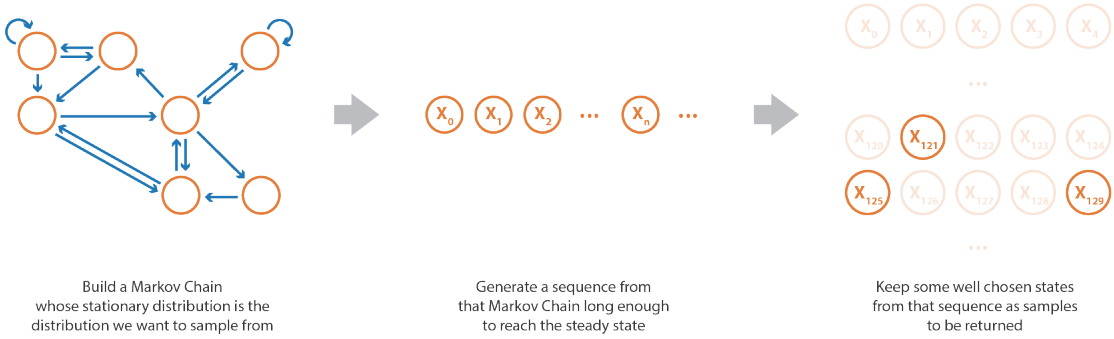

Figure 4: The Markov Chain Monte Carlo approach is aimed at generating samples from a difficult probability distribution that can be defined up to a factor.

3.3 Definition of the Markov Chain

The whole MCMC approach is based on the ability to build a Markov Chain whose stationary distribution is the one we want to sample from. In order to do so, Metropolis-Hasting and Gibbs Sampling algorithms both use a particular property of Markov Chains: reversibility.

A Markov Chain over a state space E with transition probabilities denoted by

$$k(\alpha, \beta) \equiv p(X_{n+1} = \beta | X_n = \alpha) \tag{4}$$

is said to be reversible if there exists a probability distribution γ such that

$$k(\alpha, \beta)\gamma(\alpha) = k(\beta, \alpha)\gamma(\beta) \qquad \forall \alpha, \beta \in E \tag{5}$$

For such Markov Chain, we can easily verify that we have

$$\int_{\beta \in E} k(\beta, \alpha)\gamma(\beta)d\beta = \int_{\beta \in E} k(\alpha, \beta)\gamma(\alpha)d\beta = \gamma(\alpha) \tag{6}$$

and, then, γ is a stationary distribution (the only one if the Markov Chain is irreducible).

Let's now assume that the probability distribution π we want to sample from is only defined up to a factor

$$\pi(.) = C \times g(.) \propto g(.) \tag{7}$$

(where C is the unknown multiplicative constant). We can notice that the following equivalence holds

$$\begin{aligned} & k(\alpha, \beta)\pi(\alpha) = k(\beta, \alpha)\pi(\beta) \qquad \forall \alpha, \beta \in E \\ \iff & k(\alpha, \beta)g(\alpha) = k(\beta, \alpha)g(\beta) \qquad \forall \alpha, \beta \in E \end{aligned} \tag{8}$$

and, then, a Markov Chain with transition probabilities k(.,.) defined to verify the last equality will have, as expected, π as stationary distribution. Thus, we can define a Markov Chain that have for stationary distribution a probability distribution π that can't be explicitly computed.

3.4 The Gibbs Sampling transitions (*)

Let's assume that the Markov Chain we want to define is D-dimensional, such that

$$X_n = (X_{n,1}, X_{n,2}, ..., X_{n,D}) \tag{9}$$

The **Gibbs Sampling** method is based on the assumption that, even if the joint probability is intractable, the conditional distribution of a single dimension given the others can be computed. Based on this idea, transitions are defined such that, at iteration n+1, the next state to be visited is given by the following process.

First we randomly choose an integer d among the D dimensions of X_n. Then we sample a new value for that dimension according to the corresponding conditional probability given that all the other dimensions are kept fixed:

$$d \sim Uniform(\{1, 2, ..., D\}) \quad , \quad X_{(n+1),j} = X_{n,j} \quad \forall j \neq d \quad \text{and} \quad X_{(n+1),d} \sim \pi_d(\cdot|X_{n,\neg d}) \tag{10}$$

where

$$\pi_d(\cdot|X_{n,\neg d}) = \frac{\pi(X_{n,1}, ..., X_{n,(d-1)}, \cdot, X_{n,(d+1)}, ..., X_{n,D})}{\int_u \pi(X_{n,1}, ..., X_{n,(d-1)}, u, X_{n,(d+1)}, ..., X_{n,D})du} = \frac{g(X_{n,1}, ..., X_{n,(d-1)}, \cdot, X_{n,(d+1)}, ..., X_{n,D})}{\int_u g(X_{n,1}, ..., X_{n,(d-1)}, u, X_{n,(d+1)}, ..., X_{n,D})du} \tag{11}$$

is the conditional distribution of the d-th dimension given all the other dimensions.

Formally, if we denote

$$\alpha \underset{d}{\sim} \beta \iff \alpha_i = \beta_i \qquad \forall i \neq d \tag{12}$$

the transition probabilities can then be written

$$k(\alpha, \beta) = \begin{cases} \frac{1}{D} \frac{g(\beta)}{\int_{\gamma \underset{d}{\sim} \alpha} g(\gamma)d\gamma} & \text{if } \beta \underset{d}{\sim} \alpha \\ 0 & \text{otherwise} \end{cases} \qquad \forall \beta \neq \alpha \tag{13}$$

and, so, the local balance is verified as expected with, for the only non-trivial case,

$$g(\alpha)k(\alpha, \beta) = \frac{1}{D} \frac{g(\alpha)g(\beta)}{\int_{\gamma \underset{d}{\sim} \alpha} g(\gamma)d\gamma} = \frac{1}{D} \frac{g(\beta)g(\alpha)}{\int_{\gamma \underset{d}{\sim} \beta} g(\gamma)d\gamma} = g(\beta)k(\beta, \alpha) \tag{14}$$

3.5 The Metropolis-Hasting transitions (*)

Sometimes even conditional distributions involved in Gibbs methods are far too complex to be obtained. In such cases, **Metropolis-Hasting** can then be used. For this, we start by

defining a side transition probability h(.,.) that will serve at suggesting transitions. Then, at iteration n+1, the next state to be visited by the Markov Chain is defined by the following process. We first draw a "suggested transition" x from h and compute a related probability r to accept it:

$$x \sim h(X_n, .) \qquad \text{and} \qquad r = \min\left(1, \frac{g(x)h(x, X_n)}{g(X_n)h(X_n, x)}\right) \tag{15}$$

Then the effective transition is chosen such that

$$X_{n+1} = \begin{cases} x & \text{with probability } r \\ X_n & \text{with probability } 1 - r \end{cases} \tag{16}$$

Formally, the transition probabilities can then be written

$$k(\alpha, \beta) = h(\alpha, \beta) \min\left(1, \frac{g(\beta)h(\beta, \alpha)}{g(\alpha)h(\alpha, \beta)}\right) \qquad \forall \beta \neq \alpha \tag{17}$$

and, so, the local balance is verified as expected

$$\begin{aligned} g(\alpha)k(\alpha, \beta) &= g(\alpha)h(\alpha, \beta) \min\left(1, \frac{g(\beta)h(\beta, \alpha)}{g(\alpha)h(\alpha, \beta)}\right) = \min\left(g(\alpha)h(\alpha, \beta), g(\beta)h(\beta, \alpha)\right) \\ &= g(\beta)h(\beta, \alpha) \min\left(1, \frac{g(\alpha)h(\alpha, \beta)}{g(\beta)h(\beta, \alpha)}\right) = g(\beta)k(\beta, \alpha) \end{aligned} \tag{18}$$

3.6 The sampling process

Once our Markov Chain has been defined, we can simulate a random sequence of states (randomly initialised) and keep some of them chosen such as to obtain samples that, both, follow the targeted distribution and are independent.

First, in order to have samples that (almost) follow the targeted distribution, we need to only consider states far enough from the beginning of the generated sequence to have almost reach the steady state of the Markov Chain (the steady state being, in theory, only asymptotically reached). Thus, the first simulated states are not usable as samples and we call this phase required to reach stationarity the "burn-in time". Notice that, in practice it is pretty difficult to know how long this burn-in time has to be.

Second, in order to have (almost) independent samples, we can't keep all the successive states of the sequence after the burn-in time. Indeed, the Markov Chain definition implies a strong correlation between two successive states and we then need to keep as samples only states that are far enough from each other to be considered as almost independent. In practice, the lag required between two states to be considered as almost independent can be estimated through the analysis of the autocorrelation function (only for numeric values).

So, in order to get our independent samples that follow the targeted distribution, we keep states from the generated sequence that are separated from each other by a lag L and that come after the burn-in time B. Thus, if the successive states of the Markov Chain are denoted

$$(X_n)_{n \geq 0} = X_0, X_1, X_2, ... \tag{19}$$

we only keep as our samples the states

$$X_B, X_{B+L}, X_{B+2L}, X_{B+3L}, ... \tag{20}$$

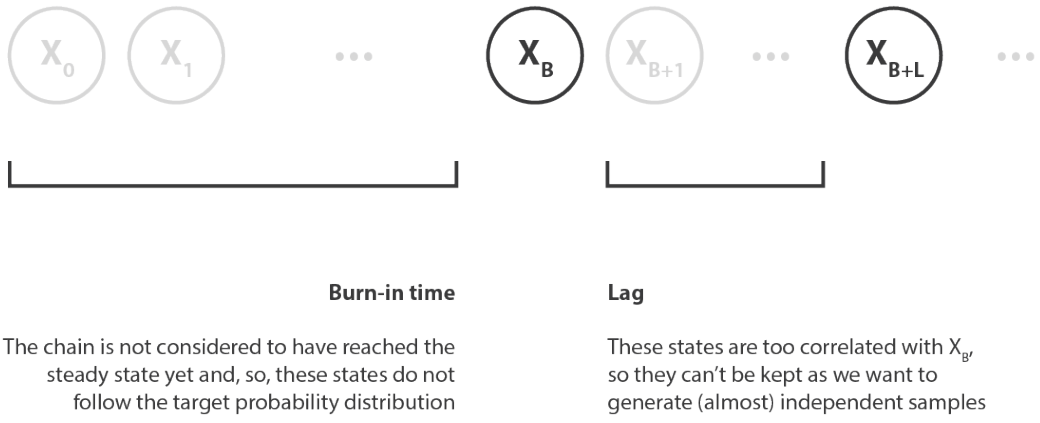

Figure 5: MCMC sampling requires to consider both a burn-in time and a lag.

4 Variational Inference (VI)

Another possible way to overcome computational difficulties related to inference problem is to use Variational Inference methods that consist in finding the best approximation of a distribution among a parametrised family. In order to find this best approximation, we follow an optimisation process (over the family parameters) that only require the targeted distribution to be defined up to a factor.

4.1 The approximation approach

VI methods consist in searching for the best approximation of some complex target probability distribution among a given family. More specifically, the idea is to define a parametrised family of distributions and to optimise over the parameters to obtain the closest element to the target with respect to a well defined error measure.

Let's still consider our probability distribution π defined up to a normalisation factor C:

$$\pi(.) = C \times g(.) \propto g(.) \tag{21}$$

Then, in more mathematical terms, if we denote the parametrised family of distributions

$$\mathcal{F}_\Omega = \{f_\omega; \omega \in \Omega\} \qquad \Omega \equiv \text{set of possible parameters} \tag{22}$$

and we consider the error measure $E(p, q)$ between two distributions p and q, we search for the best parameter such that

$$\omega^* = \arg\min_{\omega \in \Omega} E(f_\omega, \pi) \tag{23}$$

If we can solve this minimisation problem without having to explicitly normalise π, we can use f_{ω^*} as an approximation to estimate various quantities instead of dealing with intractable computations. The optimisation problem implied by variational inference approaches is,

indeed, supposed to be much simpler to handle than issues coming from direct computations (normalisation, combinatorics, ...).

Contrarily to sampling approaches, a model is assumed (the parametrised family), implying a bias but also a lower variance. In general VI methods are less accurate that MCMC ones but produce results much faster: these methods are better adapted to big scale, very statistical, problems.

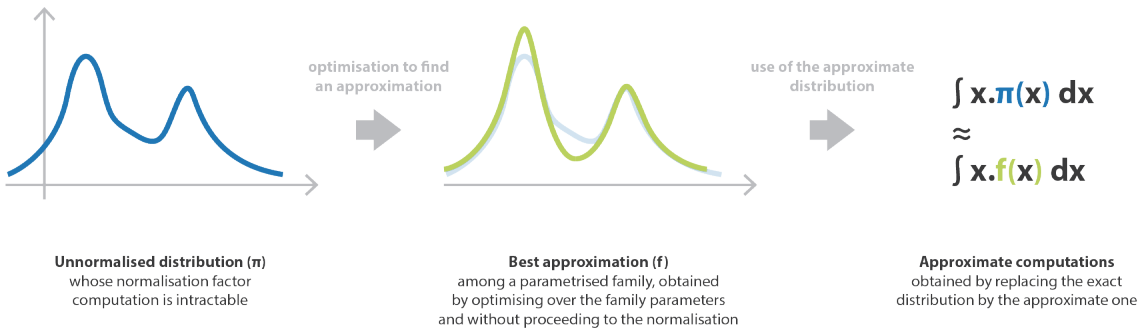

Figure 6: Illustration of the approximation approach (Variational Inference).

4.2 Family of distribution

The first thing we need to set up is the parametrised family of distributions that defines the space in which we search for our best approximation.

The choice of the family defines a model that control both the bias and the complexity of the method. If we assume a pretty restrictive model (simple family) then we have a high bias but the optimisation process is simple. Contrarily, if we assume a pretty free model (complex family) the bias is much lower but the optimisation is harder (if not intractable). Thus, we have to find the right balance between a family that is complex enough to ensure a good quality of the final approximation and a family that is simple enough to make the optimisation process tractable. We should keep in mind that if no distribution in the family is close to the target distribution, then even the best approximation can give poor results.

The **mean-field variational family** is a family of probability distributions where all the components of the considered random vector are independent. Distributions from this family have product densities such that each independent component is governed by a distinct factor of the product. Thus, a distribution that belongs to the mean-field variational family has a density that can be written

$$f(z) = \prod_{j=1}^{m} f_j(z_j) \tag{24}$$

where we have assumed a m-dimensional random variable z. Notice that, even if it has been omitted in the notation, all the densities f_j are parametrised. So, for example, if each density f_j is a Gaussian with both mean and variance parameters, the global density f is then defined by a set of parameters coming from all the independent factors and the optimisation is done over this entire set of parameters.

Figure 7: The choice of the family in variational inference sets both the difficulty of the optimisation process and the quality of the final approximation.

4.3 Kullback-Leibler divergence

Once the family has been defined, one major question remains: how to find, among this family, the best approximation of a given probability distribution (explicitly defined up to its normalisation factor)? Even if the best approximation obviously depends on the nature of the error measure we consider, it seems pretty natural to assume that the minimisation problem should not be sensitive to normalisation factors as we want to compare masses distributions more than masses themselves (that have to be unitary for probability distributions).

So, let's now define the Kullback-Leibler (KL) divergence and see that this measure makes the problem insensitive to normalisation factors. If p and q are two distributions, the KL divergence is defined as follows

$$KL(p,q) = \mathbb{E}_{z\sim p}[\log p(z)] - \mathbb{E}_{z\sim p}[\log q(z)] \tag{25}$$

From that definition, we can pretty easily see that we have

$$KL(f_\omega, Cg) = \mathbb{E}_{z\sim f_\omega}[\log f_\omega(z)] - \mathbb{E}_{z\sim f_\omega}[\log(Cg(z))] = \mathbb{E}_{z\sim f_\omega}[\log f_\omega(z)] - \mathbb{E}_{z\sim f_\omega}[\log g(z)] - \log C \tag{26}$$

which implies the following equality for our minimisation problem

$$\omega^* = \arg\min_{\omega\in\Omega} KL(f_\omega, \pi) = \arg\min_{\omega\in\Omega} KL(f_\omega, Cg) = \arg\min_{\omega\in\Omega} KL(f_\omega, g) \tag{27}$$

Thus, when choosing KL divergence as our error measure, the optimisation process is not sensitive to multiplicative coefficients and we can search for the best approximation among our parametrised family of distributions without having to compute the painful normalisation factor of the targeted distribution, as it was expected.

Finally, as a side fact, we can conclude this subsection by noticing for the interested readers that the KL divergence is the cross-entropy minus the entropy and has a nice interpretation in information theory.

4.4 Optimisation process and intuition

Once both the parametrised family and the error measure have been defined, we can initialise the parameters (randomly or according to a well defined strategy) and proceed to the

optimisation. Several classical optimisation techniques can be used such as gradient descent or coordinate descent that will lead, in practice, to a local optimum.

In order to better understand this optimisation process, let's take an example and go back to the specific case of the Bayesian inference problem where we assume a posterior such that

$$p(z|x) \propto p(x|z)p(z) = p(x,z) \tag{28}$$

In this case, if we want to get an approximation of this posterior using variational inference, we have to solve the following optimisation process (assuming the parametrised family defined and KL divergence as error measure)

$$\begin{aligned}
\omega^* &= \arg\min_{\omega\in\Omega} KL(f_\omega(z), p(z|x)) \\
&= \arg\min_{\omega\in\Omega} KL(f_\omega(z), p(x,z)) \\
&= \arg\max_{\omega\in\Omega} (-KL(f_\omega(z), p(x,z))) \\
&= \arg\max_{\omega\in\Omega} (\mathbb{E}_{z\sim f_\omega}[\log p(z)] + \mathbb{E}_{z\sim f_\omega}[\log p(x|z)] - \mathbb{E}_{z\sim f_\omega}[\log f_\omega(z)]) \\
&= \arg\max_{\omega\in\Omega} (\mathbb{E}_{z\sim f_\omega}[\log p(x|z)] - KL(f_\omega, p(z)))
\end{aligned} \tag{29}$$

The last equality helps us to better understand how the approximation is encouraged to distribute its mass. The first term is the expected log-likelihood that tends to adjust parameters so that to place the mass of the approximation on values of the latent variables z that explain the best the observed data. The second term is the negative KL divergence between the approximation and the prior that tends to adjust the parameters in order to make the approximation be close to the prior distribution. Thus, this objective function expresses pretty well the usual prior/likelihood balance.

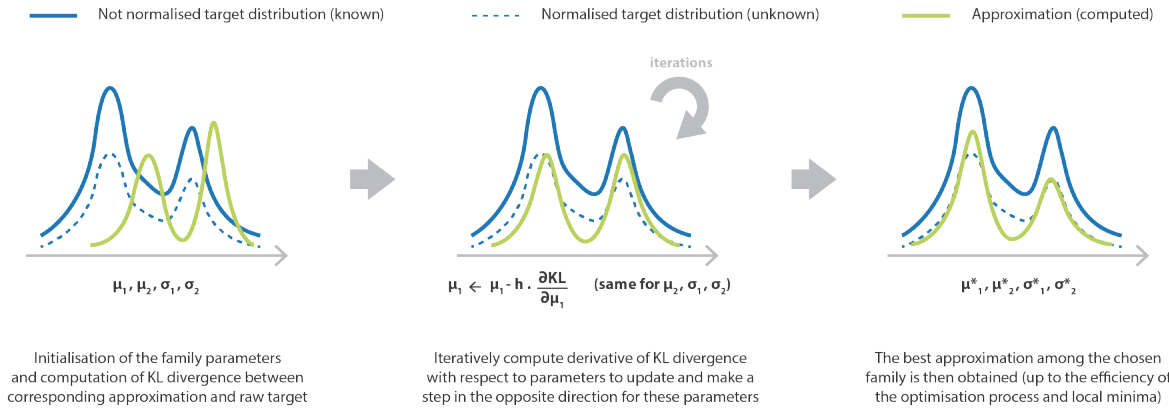

Figure 8: Optimisation process of the Variational Inference approach.

5 Takeaways

The main takeways of this article are:

- Bayesian inference is a pretty classical problem in statistics and machine learning that relies on the well known Bayes theorem and whose main drawback lies, most of the time, in some very heavy computations
- Markov Chain Monte Carlo (MCMC) methods are aimed at simulating samples from densities that can be very complex and/or defined up to a factor
- MCMC can be used in Bayesian inference in order to generate, directly from the "not normalised part" of the posterior, samples to work with instead of dealing with intractable computations
- Variational Inference (VI) is a method for approximating distributions that uses an optimisation process over parameters to find the best approximation among a given family
- VI optimisation process is not sensitive to multiplicative constant in the target distribution and, so, the method can be used to approximate a posterior only defined up to a normalisation factor

As already mentioned, MCMC and VI methods have different properties that imply different typical use cases. In one hand, the sampling process of MCMC approaches is pretty heavy but has no bias and, so, these methods are preferred when accurate results are expected, without regards to the time it takes. In the other hand, although the choice of the family in VI methods can clearly introduce a bias, it comes along with a reasonable optimisation process that makes these methods particularly adapted to very large scale inference problem requiring fast computations.

Additional comparisons between MCMC and VI can be found in the excellent Variational Inference: A Review For Statisticians, that we also highly recommend for readers interested in VI only. For further readings about MCMC, we recommend this general introduction as well as this machine learning oriented introduction. The reader interested to learn more about Gibbs Sampling applied to LDA can refer to this Tutorial on Topic Modelling and Gibbs Sampling (combined with these lecture note on LDA Gibbs Sampler for cautious derivation).

Finally, let's conclude with a little bit of teasing and mention that in an upcoming post we will discuss Variational Auto Encoder, a deep learning approach that is based on variational inference... so stay tuned!

Thanks for reading and feel free to share if you think it deserves to be!