Variational Refinement for Importance Sampling Using the Forward Kullback-Leibler Divergence

Eugène Berta
MVA, ENS Paris Saclay
Télécom Paris, IP Paris
eugene.berta@gmail.com

Abstract

In the framework of Bayesian statistics, it is often the case that the posterior distribution $p(\theta|x)$ is only known up to a normalization constant, making it impossible to sample from. In this setting, Importance Samplig (IS) can be used to compute expectations over the posterior $\mathbb{E}_{p(\theta|x)}[f(\theta)]$. However, the performances of IS crucially depends on the choice of the proposal distribution q. It has been suggested in the literature that one can build a good proposal distribution q using Variational Inference (VI) [1]. The paper we study in this report [2] builds upon this idea, noticing that VI classically minimizes the Reverse Kullback Leibler divergence (RKL) $KL(q||p)$ to the target distribution, which results in undesired properties for the proposal distribution q. In contrast, minimizing the Forward Kullback-Leibler divergence (FKL) $KL(p||q)$ might result in a richer q. However, it requires sampling from the target distribution p which is not possible in our framework. The authors propose a method to alleviate this difficulty. They demonstrate the convergence speed of their algorithm towards the ideal proposal distribution, and they test their method on a variety of examples. In this report, we first present this algorithm in details. We notice that the original paper is hardly reproducible and that the implementation proposed is based on the autograd [3] library, which is no longer being developed. We propose a PyTorch [4] implementation of the method and reproduce some of the results obtained in the original paper. Finally, we use our implementation to expose some limitations of the proposed algorithm.

1 How to build a good proposal distribution ?

1.1 RKL versus FKL

Variational Inference is a popular method in Bayesian Statistics. It casts the problem of approximating the posterior distribution $p(\theta|x)$ as an optimization problem:

$$q_R^* = \arg\min_{q \in Q} KL(q||p) = \arg\min_{q \in Q} \int q \log \frac{q}{p} \tag{1}$$

Where p is the posterior distribution or "target distribution" and q is optimized over a set of possible distributions Q. There is no guarantee that the posterior distribution p belongs to the set Q, so the result obtained is biased ($q_R^* \neq p$). However, a simple idea is to use the distribution q_R^* obtained as a proposal distribution for an importance sampling procedure to correct the bias in our estimates.

To understand whether q_R^* would make a good candidate as a proposal distribution for Importance Sampling, let's look in more details at the Kullback-Leibler divergence. We immediately observe that the term $\log \frac{q}{p}$ will blow to $+\infty$ if there exists some x such that $p(x) = 0$ and $q(x) > 0$. This implies that the support of the resulting distribution q_R^* will be included in the support of the posterior

Preprint. Under review.

p. In other words, the target distribution p will not be fully covered by the proposal distribution q_R^*, which is classically an issue when building a proposal distribution for IS. $\text{Supp}(p) \subset \text{Supp}(q)$ is well dealt with by IS as the weights associated with the uninformative samples will be set to zero ($\frac{p(x)}{q(x)} = 0$ if $p(x) = 0$). On the other hand, $\text{Supp}(q) \subset \text{Supp}(p)$ will introduce a bias in the result as some regions of the target distribution p will never be visited by samples from the distribution q.

However, we can take advantage of this property of the KL divergence by simply inverting p and q in the objective function in (1). This way we guarantee, using the same observation as before, that the proposal distribution q^* will cover all the support of the posterior p. This is a strong argument in favor of minimizing the FKL $KL(p||q)$ instead of the RKL $KL(q||p)$ as is traditionally done for VI.

We illustrate this by fitting a two dimensional Gaussian distribution on a bimodal target distribution in Figure 1. We see that minimizing the RKL leads to a "mode seeking" behavior where the mass of q concentrates around a single mode while minimizing the FKL results in a "mass covering" behaviour where the mass of q is spread over the whole target distribution. In this simple example, we clearly see that the distribution q_F^* resulting from the FKL minimization is a better candidate for Importance Sampling. It will produce samples from every region of the target distribution landscape, resulting in a non-biased IS estimator.

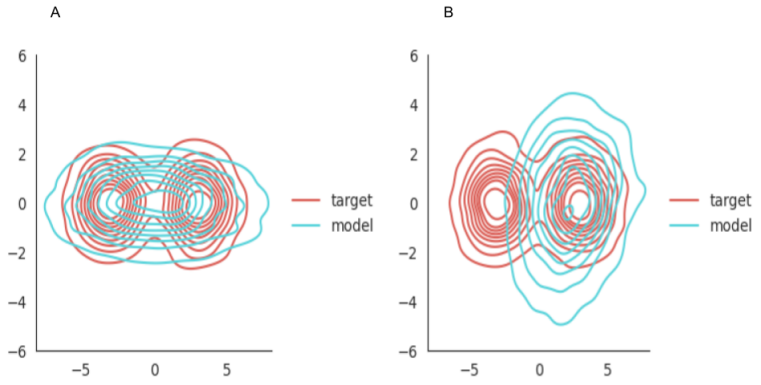

Figure 1: Fitting a two dimensional Gaussian distribution "model" on a bimodal distribution "target" by minimizing the KL divergence. A: using FKL with random mean initialization. B: using RKL with random mean initialization.

1.2 Minimizing the FKL

These observations lead us to believe that minimizing the FKL instead of the RKL in (1) leads to a better proposal distribution q_F^* than the traditional q_R^* obtained with VI. Thus, we aim at minimizing the FKL:

$$q_F^* = \arg\min_{q \in Q} KL(p||q) = \arg\min_{q \in Q} \mathbb{E}_p[\log \frac{p}{q}] \tag{2}$$

We see that this requires evaluating the expectation of a given function over the posterior p, which brings us back to the initial problem. The authors propose bypassing this problem using the Importance Sampling trick:

$$KL(p||q) = \mathbb{E}_p[\log \frac{p}{q}] = \mathbb{E}_q[\frac{p}{q} \log \frac{p}{q}]$$

With this inversion, we can sample from q instead of p and we can evaluate the expectation using a Monte Carlo estimate. The last issue to solve is that we can only evaluate p up to an unknown normalization constant. We need to make sure that the weights $\frac{p}{q}$ sum up to 1. For this, we can use Self-Normalized Importance Sampling (SNIS) as described in Algorithm 1.

Equipped with this method, we can compute the Forward KL divergence. Moreover, we can use any modern automatic-differentiation library to compute the gradients of the model distribution q with respect to the FKL "loss function" $\nabla_g KL(p||q)$. However, directly optimizing q using this method can be problematic. Indeed, we are minimizing the expectation over q of a function depending on q.

Algorithm 1 Computing the FKL with SNIS

Sampling: $\theta_s \sim q(\theta)$
IS weights: $r_s = \frac{p(\theta_s|x)}{q(\theta_s)}$
Normalizing weights: $w_s = \frac{r_s}{\sum r_s}$
$KL(p||q) = \sum w_s \log \frac{p(\theta_s|x)}{q(\theta_s)}$

This co-dependency can result in a very high variance in the FKL estimate. Moreover, the variance of the SNIS estimate can be arbitrary large. In practice we will see that this method works only for very simple models, hence the need to reduce the variance of the FKL estimator.

1.3 Forward KL Boosting

The authors suggest building a proposal distribution q in the set of Gaussian Mixture Models (GMM) $Q = \{\sum_i \lambda_i f_i | \lambda_i \geq 0, \sum_i \lambda_i = 1, f_i \sim \mathcal{N}(\mu_i, \Sigma_i)\}$. These models are known to be very expressive and can theoretically approximate any distribution with good precision. A classical way to fit a GMM on a target distribution is to fit Gaussian particles f_i and associated weights λ_i one at a time using a "boosting procedure". This idea has the advantage of providing a beautiful solution to our problem of variance of the FKL estimator.

Indeed, at each iteration of the boosting procedure, we can sample in the current state distribution of the model q_{i-1} and optimize the new Gaussian particle f_i with respect to the FKL at the next state $KL(p||q_i)$ (where $q_i = (1 - \lambda_i)q_{i-1} + \lambda_i f_i$). With this sequential method, we can re-write the optimization problem at each step as:

$$f_i^*, \lambda_i^* = \underset{f_i \in \mathcal{N}(\mu,\Sigma), \lambda_i \in [0,1]}{\arg\min} \mathbb{E}_{q_{i-1}} \left[\frac{p}{q_{i-1}} \log \frac{p}{(1-\lambda_i)q_{i-1} + \lambda_i f_i} \right]$$

We immediately see that the optimization on f_i, λ_i does not depend on the distribution q_{i-1} on which we sample to compute the FKL. This alleviates the issue of the co-dependency and thus drastically reduces the variance of the FKL estimate. Equipped with this method, we can hope to fit a GMM on the target p sequentially, by minimizing the Forward KL divergence.

1.4 Theoretical analysis

The authors provide a theoretical analysis of this algorithm. They discuss in particular the convergence speed and moment estimation error. We will not detail these contributions here and we orient the interested reader to the original paper.

2 Limitations, Personal contributions

In this section, we use our PyTorch implementation of the method described in the paper to run several experiments. We reproduce some results of the original paper and illustrate what we consider are limitations of the algorithm. We refer to the github repository of the project for more details on the implementation and experiment settings `https://github.com/eugeneberta/mva-bml`.

2.1 Strenghts of the method

First, our experiments confirm several strong points of the algorithm.

FKL is well adapted to under-fitted models. As illustrated in Figure 1, when fitting a Gaussian particle on a bimodal target, minimizing the FKL instead of the RKL allows to cover the two modes of the distribution. In real world applications, one may try to estimate a complex multi-modal distribution without knowing precisely it's shape or number of modes. In this case, using the FKL provides a guarantee that all modes will be covered by the model q. By opposition, if the model is too simple (not enough particles) a GMM fitted with RKL will almost surely miss some modes of the target distribution. This appealing property of FKL was anticipated by the authors and seems

to be confirmed empirically when using the SNIS estimate of FKL without the boosting procedure. However, we discussed the fact that this estimate can only be used in very simple cases. When using the boosting procedure in more complex applications, it is not guaranteed that this property of FKL will remain true.

Boosting is well adapted to reduce SNIS variance. To alleviate the variance difficulty, the authors propose to fit a GMM via a boosting procedure: fitting particles one at a time using the previous state of the model as a proposal distribution for the SNIS estimator. In our experiments, we observe that this method improves drastically the reliability of the FKL loss. Running the optimization after initializing the mixture model produces consistent results. However it poses the problem of the initialization of the model.

2.2 The initialization problem

The diffuse initialization problem. To illustrate the effect of initialization on the algorithm, we fit the same bimodal target distribution as in the previous experiment. We initialize a GMM with a centered and diffuse (high variance) Gaussian distribution. We run the boosting procedure described in the paper for a single step, adding a Gaussian particle to our hierarchical model. The results are visible in Figure 2.

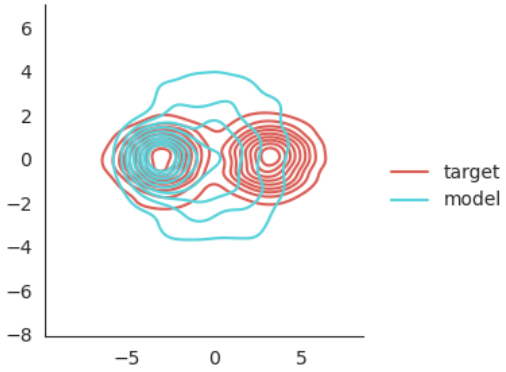

Figure 2: Fitting a bimodal target distribution with FKL, using a 1 step boosting procedure with a centered diffuse initialization.

We observe that the result is (sometimes) very different from what we expect from FKL. Instead of covering the two modes, the new particle is concentrated only on the left mode of the distribution, resulting in a very unbalanced model.

We see here that the initialization can alter the idealized behavior of FKL. Indeed, the diffuse initialization puts a minimum amount of mass everywhere and it satisfies the mass covering constraint in the FKL objective (2). This can create local minimas in the loss landscape for which the new particle covers only one mode of the distribution. The model relies on it's diffuse initialization to satisfy the hard constraint that it must cover the whole target distribution but it is highly unbalanced towards one mode. This issue would not appear if we could fit our model in one step, it is a collateral damage from the boosting procedure.

The RKL initialization problem. One could argue that this problem can be eliminated by initializing the GMM with a particle fitted with RKL as the authors propose in the paper. In this case the initialization would cover only one mode of the target distribution and the mass covering constraint of the FKL would have some effect when fitting the second particle.

This would result in the second particle covering the second mode of the distribution. We recall that this does nos correspond to the original idea of the paper, which to our opinion, is about avoiding to fit one particle on each mode of the target as is done with RKL.

More importantly, we argue that initializing with RKL also drastically reduces the expressiveness of the model. Indeed, at each step of the boosting procedure, samples from the previous state of

the model q_{i-1} are used to compute an estimate of the FKL for the current state of the model q_i. With RKL initialization, theses samples would be concentrated around a single mode, providing little information about the loss in other regions of the target distribution. This initialization will fail to carry useful information about how to cover other modes in the case of a multi modal target distribution. To illustrate this limitation, we run a new experiment. We design a more complex target distribution with four equally weighted, well separated modes. We fit a GMM with a 3 steps boosting procedure using: a centered diffuse initialization (Figure 3.A) and a RKL fitted initialization (Figure 3.B). This experiment illustrates well the two problems we have been discussing. When initializing with a diffuse centered Gaussian, the target distribution is correctly covered, but the model is unbalanced towards certain modes. When initializing with RKL, the model hardly manages to escape the initialization mode. This is due to the fact that the proposal distributions for the following steps are not informative about the rest of the target distribution.

In the examples of the original paper the model still manages to reach well separated modes. Our guess is that the larger number of samples they use plays a role. Still, it also takes them many boosting iterations to reach distant modes whereas the use of the FKL should guarantee mass covering whatever the size of the model.

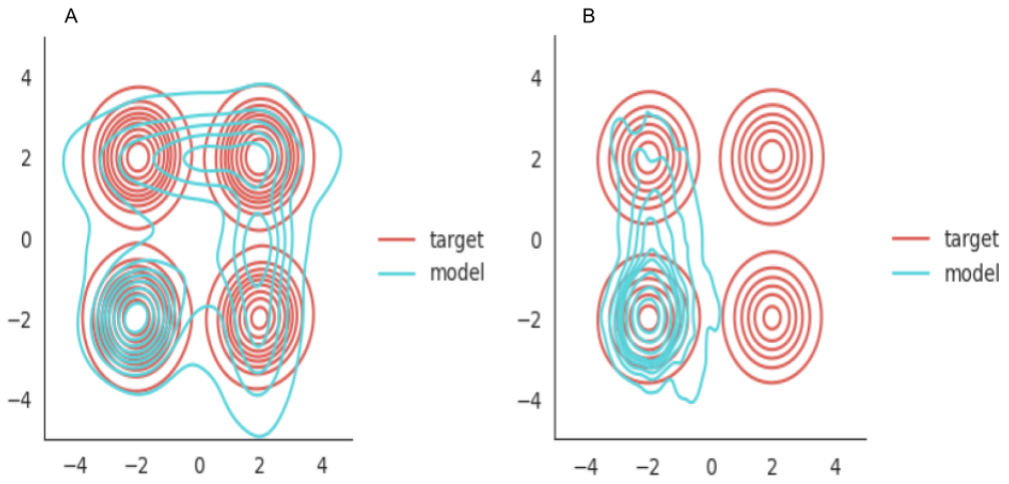

Figure 3: Fitting a GMM "model" on a well separated multi-modal distribution "target" by minimizing the Forward Kullback-Leibler divergence. A: using a boosting procedure with diffuse centered initialization. B: using a boosting procedure with RKL initialization.

Conclusion

Building a good proposal distribution for Importance Sampling when dealing with a potentially complex (multidimensional, multi-modal) posterior distribution is challenging. The paper we studied points out that the Forward Kullback-Leibler divergence has properties that adapt very well to this problem. Fitting a model with FKL would provide good guarantees about the proposal distribution obtained. Moreover, the boosting procedure they introduce makes an excellent job at reducing the variance of the SNIS estimate of the FKL, making it possible to fit a model using FKL. Despite these breakthroughs, we argue that the sequential aspect of the method introduced corrupts the expected behavior of the FKL, which reduces it's interest. In particular, we argue that the choice between diffuse and RKL initialization is a problem with no satisfying answer.

References

[1] Yuling Yao, Aki Vehtari, Daniel Simpson, and Andrew Gelman. Yes, but did it work?: Evaluating variational inference. 2018.

[2] Ghassen Jerfel, Serena Wang, Clara Fannjiang, Katherine A. Heller, Yian Ma, and Michael I. Jordan. Variational refinement for importance sampling using the forward kullback-leibler divergence, 2021.

[3] Dougal Maclaurin, David Duvenaud, and Ryan P Adams. Autograd: Effortless gradients in numpy. In *ICML 2015 AutoML workshop*, volume 238, 2015.

[4] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library, 2019.