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

Louise Allain **Antoine Ratouchniak**
École Normale Supérieure Paris-Saclay
louise.allain@ens-paris-saclay.fr antoine.ratouchniak@ens-paris-saclay.fr

1 Introduction

In bayesian framework, when confronted with an intractable target distribution p_{target}, we want to learn a distribution p_{pred} that closely resembles the target. A popular way to infer such a distribution is Variational Inference [1] which selects the most suitable distribution from a parametric family of distributions, thus recasting the problem into an optimization problem. In this context, selecting an appropriate method to measure the similarity between two distributions becomes crucial. The reverse Kullback-Leibler divergence is often employed for this purpose. However, minimizing the reversed Kullback-Leibler divergence typically results in underestimating the tail of the posterior. For this reason, [8] investigate another objective as well as refinements for the Variation Inference. In addition to explaining the paper and providing pseudocode, we implement the algorithm and test it on synthetic data to determine how it performs on different distributions. We also try to construct a generative model based on the paper and test it on MNIST. Finally, we exemplify the issue of underestimating heavy tails when employing Importance Sampling by considering Gaussian distributions.

The source code is available at https://github.com/AntoineRtk/BML-Project.

2 Background

Let $\mathcal{X} \subset \mathbb{R}^d$ be an arbitrary space and $x \in \mathcal{X}$ some continuous or discrete samples from this space. In the Bayesian inference setup, we aim to estimate

$$\mathbb{E}_{Z\sim p(Z|x)}[f(Z)]$$

with f some measurable function, $Z \in \mathcal{Z}$ a latent variable, and $p(z|x)$ the often intractable true posterior distribution. In addition, the posterior $p(z|x)$ is often known only up to a normalization constant, hence the need to estimate instead of calculating it.

To tackle those limitations, we can resort to Variational Inference (VI) [1]. Formally, we seek to minimize the reversed Kullback-Leibler divergence defined as

$$p^*_{\text{pred}} = \underset{p_{\text{pred}}\in\mathcal{M}}{\arg\min}\, KL(p_{\text{pred}}||p) = \underset{p_{\text{pred}}\in\mathcal{M}}{\arg\min}\, \mathbb{E}_{p_{\text{pred}}}\left[\log \frac{p_{\text{pred}}}{p}\right] \tag{RKL}$$

between the true posterior $p(z|x)$ and $p_{\text{pred}}(z)$, an approximate distribution, usually belonging to a tractable family $\mathcal{M}$. Being a convex problem as it is the relative entropy plus a linear function, such an optimization is easy to process. The distribution obtained can then be sampled to estimate the expectancy.

One of the main methods for estimating the expectation is Markov Chain Monte Carlo (MCMC) [11] which samples $z_1, ..., z_n$ from $p(z|x)$ and calculates the expectation as

$$\mathbb{E}_{Z\sim p(Z|x)}[f(z)] \simeq \frac{1}{n}\sum_{z\in\mathcal{Z}} f(z)$$

However, when facing multimodal distributions or high dimensional data, the convergence happens to be very slow and computationally expensive. From now on, we refer to p_{target} as the probability distribution we wish to compute. Moreover, two new challenges arise when using this framework, notably due to the tendency of the reversed Kullback-Leibler divergence to underestimate the tail of distribution by encouraging the concentration of the mass around a mode. First, if the family $\mathcal{M}$ is misspecified, i.e if there is no distribution in the family capable of approaching the posterior, we will face an unknown bias in the Variational Inference solution [2]. Second, the approximation tends to neglect the heavy tail of some distributions. This last problem is due to the use of reversed Kullback-Leibler divergence that knowingly leads to poor estimates of heavy tails as for $p_{\text{target}} = 0$ and $p_{\text{pred}} > 0$, the divergence is infinite, enforcing the fact that $p_{\text{pred}} = 0$ when $p_{\text{target}} = 0$, we call this property zero-forcing and it typically leads to underestimation of the support of the distribution. Such difficulties particularly manifest when facing multimodal target distributions, as the lack of exploration caused by the light tail will foster single-mode estimates. This bias will generally lead us to oversampling in some important regions, especially around the modes. To correct this behavior, it can be interesting to use Importance Sampling (IS) [10, 15], whose basic idea is to draw further samples in regions where the target probability is higher. This will also result in a lower variance for the estimator. To do so, an importance weight w_i can be incorporated to the samples drawn from that estimate. This weight serves as a correction term during sampling. The importance weights are defined as

$$w(z) = \frac{p_{\text{target}}(z)}{p_{\text{pred}}(z)} \tag{1}$$

This adjustment on f helps to compensate sampling from p_{pred} instead of p_{target}. We then have

$$\mathbb{E}_{p_{\text{pred}}}\left[\frac{p_{\text{target}}(Z)}{p_{\text{pred}}(Z)} f(Z)\right] = \sum_{z \in \mathcal{Z}'} p_{\text{pred}}(z) \frac{p_{\text{target}}(z)}{p_{\text{pred}}(z)} f(z) dz = \sum_{z \in \mathcal{Z}} p_{\text{target}}(z) f(z) dz = \mathbb{E}_{p_{\text{target}}}[f(Z)]$$

as long as p_{pred} and p_{target} are equivalent, implying that $\mathcal{X}' = \mathcal{X}$. One can show that such an estimate is consistent as the bias decreases at a rate $\mathcal{O}(1/S)$ [14], but can display high or even infinite variance, especially when the support of the estimate is too small compared to the real support [5].

Example: Let $f(x) = x$ with both p_{target} and p_{pred} Gaussians such that $p_{\text{target}} = \mathcal{N}(0, 1)$ and $p_{\text{pred}} = \mathcal{N}(0, \sigma), \sigma > 0$. Then

$$\text{Var}_{p_{\text{pred}}}(f(X)) = \int_{\mathbb{R}} \frac{(f(x) p_{\text{target}}(x))^2}{p_{\text{pred}}(x)} dx = \int_{\mathbb{R}} x^2 \left(\frac{e^{-\frac{x^2}{2}}}{\sqrt{2\pi}}\right)^2 \frac{\sqrt{2\pi}\sigma}{e^{-\frac{x^2}{2\sigma^2}}} dx = \frac{\sigma}{\sqrt{2\pi}} \int_{\mathbb{R}} x^2 e^{-\frac{x^2(2-\sigma^{-2})}{2}} dx$$

By computing the integral, we observe that the variance is infinite for $\sigma \leq \frac{1}{2}$ and $\sigma > \frac{1}{2}$,

$$\text{Var}_{p_{\text{pred}}}[f(X)] = \frac{\sigma}{(2 - \sigma^{-2})^{\frac{3}{2}}}$$

This instability makes it necessary to correct the light tail of our estimate, as a high variance would force us to perform more samples to attain the desired precision. Jerfel et al. [8] propose to change the objective function from the reversed Kullback-Leibler divergence $KL(p_{\text{pred}}||p_{\text{target}})$ to the forward Kullback-Leibler divergence defined as

$$KL(p_{\text{target}}||p_{\text{pred}}) = \mathbb{E}_{p_{\text{target}}}\left[\log \frac{p_{\text{target}}}{p_{\text{pred}}}\right] \tag{FKL}$$

One can see that the quantity is infinite when $p_{\text{pred}} \approx 0$ and $p_{\text{target}} > 0$, enforcing that the support of p_{pred} is greater than the one of the target, correcting the zero-forcing behavior of the reversed Kullback-Leibler divergence. This also enables us to combine the computational efficiency of Variational Inference with the consistency of Importance Sampling by controlling the error on the Importance Sampling estimation thanks to better tail coverage. In fact, the variance of the Importance Sampling estimate would then scale exponentially with FKL as the Importance weight should make up for the overestimation of the support [4].

However, a problem persists: similar to its reversed version, the forward Kullback-Leibler divergence is intractable most of the time as the integral might not have a closed-form solution, preventing us from knowing the target probability. Thus, instead of computing directly the forward Kullback-Leibler

divergence, we approximate it by Self-Normalized Importance Sampling (SNIS) which corresponds to setting

$$KL(p_{\text{target}}||p) = \sum_{s=1}^{S} z_s \log\left(\frac{p_{\text{target}}(z_s)}{p_{\text{pred}}(z_s)}\right) \quad \text{with} \quad z_s \sim p_{\text{pred}}(Z) \quad r_s = \frac{r_s}{\sum_{s=1}^{S} r_s} \quad r_s = \frac{p_{\text{target}}(z_s)}{p_{\text{pred}}(z_s)} \tag{2}$$

Despite being consistent [13], such an estimate can have very high variance depending on the prediction p_{pred}, especially when optimizing over the same distribution from which samples are drawn.

Variational boosting: To bypass this issue, one approach can be variational boosting [12]. The idea of boosting is to iteratively add components to approximate the target distribution. Following Miller's et al. work [12], we consider the family of Gaussian mixture $\mathcal{M} = \{q(x;\theta) \mid q(\cdot;\theta) = \sum_{i=1}^{N} \lambda_i \mathcal{N}(\mu_i, \Sigma_i),\ \sum_{i=1}^{N} \lambda_i = 1,\ \lambda_i > 0\ \forall i \in \{1,\dots,N\}\}$ and denote by $p^k_{\text{pred}}(\cdot;\mu_k,\Sigma_k) \sim \mathcal{N}(\mu_k,\Sigma_k)$ the k^{th} component of the distribution. When fitting a new component, at the first iteration $i = 0$, we usually take a gradient step towards minimizing the reversed KL divergence defined in equation (RKL) between the prediction, belonging to $\mathcal{M}$, and the target one.
Then, based on Jerfel's et al. [8] article, at the i^{th} iteration, $i > 1$, we minimize the forward KL divergence (FKL) between the new distribution and the target distribution while keeping the weights and parameters of previously learned components identical. Formally, the k^{th} boosting iteration reads

$$(\mu_k^*, \Sigma_k^*, \gamma^*) = \underset{\mu_k,\Sigma_k,\gamma}{\arg\min}\, KL\left(p_{\text{target}}||\gamma f_k(\cdot;\mu_k,\Sigma_k) + (1-\gamma)\sum_{j=1}^{k-1}\lambda_j q_j(\cdot;\mu_j,\Sigma_j)\right) \tag{3}$$

where $q_j(\cdot;\mu_j,\Sigma_j) = p^j_{\text{pred}}(\cdot;\mu_j,\Sigma_j)$, $f_k \sim \mathcal{N}(\mu_k,\Sigma_k)$ and $\gamma \in [0,1]$. In practice, γ is initially set to 0.5. However, it is not clear if the authors chose to update it along the iterations. We decide to update γ only at the end. Finally, the weights $\{\lambda_k\}_{i=1}^K$ of p_{pred}, with K the number of boosting iterations, are re-optimized at the end using the simplex-projected gradient descent algorithm [3], allowing to keep the constraint $\sum_{i=1}^{K} \lambda_i = 1$. Other strategies could have been considered such as line search. As mentioned above, we also use the SNIS approximation (2). In the end, the problem we aim to solve is

$$(\{\mu_k\}_{k=1}^K, \{\Sigma_k\}_{k=1}^K, \{\lambda_k\}_{k=1}^K) = \underset{\mu,\Sigma,\lambda}{\arg\min}\, KL\left(p_{\text{target}}||\sum_{k=1}^{K}\lambda_k\mathcal{N}(\mu_k,\Sigma_k)\right) \tag{4}$$

We describe the Algorithm in 1.

3 Experiments

In this section, we perform simple experiments with the boosting strategy and the forward KL.

3.1 Reducing the gap between modes

To observe how the algorithm performs when the modes get closer, we consider a mixture of two Gaussian distributions of variance 1 and vary their modes. It can be seen in Figure 1 that it gets harder for the algorithm to get a good approximation in terms of the number and the variance of the modes as they get closer. More precisely, when the means are too close, the algorithm seems to overestimate the number of Gaussians in the mixture. However, as pointed out in [8], when the modes are well separated, our implementation does not face the problems of zero-forcing and over-pruning.

3.2 Larger scale dataset

There is, to the best of our knowledge, no generative model designed using the forward KL divergence. Usually, a variational distribution $q_\phi(z|x)$ is introduced where ϕ is a set of parameters. Then, a

Algorithm 1: Boosting with Forward Kullback-Leibler Variational Inference using SNIS

Input: Target distribution p_{target}, number of boosting iterations K, weights $\{\lambda_i\}_{i=1}^{N_b}$, mixture components (μ, σ), number of iterations N_{iter}, learning rate η

Output: q_{N_b} the estimated distribution

for $k = 1, \ldots, K$ **do**
- Set initial parameters $\theta = (\mu_k, \Sigma_k)$
- $\mu_k \leftarrow 0$
- $\Sigma_k \leftarrow Diag(\mathbf{1})$
- **if** $k > 1$ **then**
  - $\mu_k \leftarrow \arg\max_{x \in \mathcal{X}} \frac{p_{\text{target}}(x)}{p_{\text{pred}}(x)}$ `// initialize the mean of the distribution at the maximum of the remainder`
- $f_k \sim \mathcal{N}(\mu_k, \Sigma_k)$
- $\gamma \leftarrow 0.5$
- **for** $i = 1, \ldots, N_{iter}$ **do**
  - Draw samples $\boldsymbol{x} = (x_1, \ldots, x_n) \sim p_{\text{target}}$
  - **if** $k > 1$ **then**
    - $q_k \leftarrow \gamma f_k + (1-\gamma) \sum_{j=1}^{k-1} \lambda_j q_j$
    - $w(\boldsymbol{x}) \leftarrow \frac{p_{\text{target}}(\boldsymbol{x})}{\sum_{j=1}^{k-1} \lambda_j q_j(\boldsymbol{x})}$
    - $w(\boldsymbol{x}) = \frac{w(\boldsymbol{x})}{\sum_x w(\boldsymbol{x})}$ `// self-normalized importance sampling (2)`
  - **else**
    - $w(\boldsymbol{x}) \leftarrow \mathbf{1}$ `// weights are set to 1 for the first component`
  - **if** $i = 1$ **then**
    - $g_\theta \leftarrow \nabla_\theta \mathbb{E}_q[w \log \frac{q}{p_{\text{target}}}]$ `// reversed KL for the first iteration`
  - **else**
    - $g_\theta \leftarrow \nabla_\theta \mathbb{E}_{p_{\text{target}}}[w \log \frac{p_{\text{target}}}{q}]$ `// forward KL for the other ones`
  - $\theta \leftarrow \theta - \eta \cdot g_\theta$ `// perform a gradient step on the current parameters`
- **for** $i = 1, \ldots, N_{iter}$ **do**
  - $g_{\lambda_{\leq k}} \leftarrow \nabla_{\lambda_{\leq k}} \mathbb{E}_{p_{\text{target}}} \left[\log \frac{p_{\text{target}}}{\sum_{j=1}^{k} \lambda_j q_j} \right]$
  - $\lambda_{\leq k} \leftarrow \lambda_{\leq k} - \eta \cdot g_{\lambda_{\leq k}}$ `// perform a gradient step on the mixture weights`
  - $\{\lambda_k\}_{k=1}^K \leftarrow \textit{Simplex}(\{\lambda_k\}_{k=1}^K)$ `// apply the simplex algorithm [3]`
- $\lambda_{\leq i} \leftarrow \sigma(\lambda_{\leq i})$ `//` $\sigma(\cdot)$ `denotes the sigmoid function`
- $q_k \leftarrow f_k$

generative model $p_\theta(x|z)$, where θ is another set of parameters, is learned to generate samples from this distribution. These parameters are commonly learned by maximizing the ELBO [9] defined as

$$ELBO(q) = \mathbb{E}_q[\log p_\theta(x|z)] - KL(q_\theta(z)||p_\theta(z)) \tag{5}$$

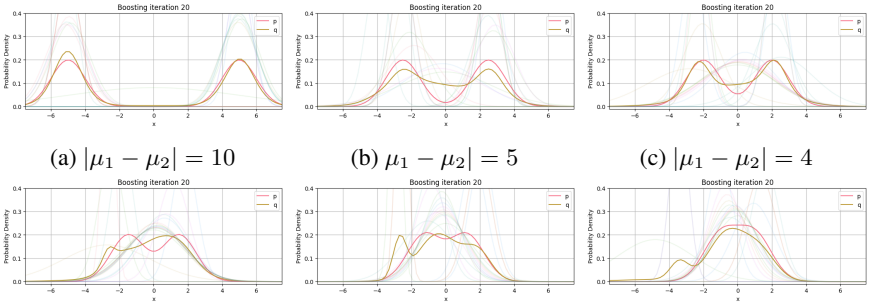
In this section, we try to use the forward KL in the NVAE model [16]. In simple words, this architecture aims to represent the data through the layers as factorial Normal distributions. Formally,

$$q_\phi(z|x) = \Pi_l q_\phi(z_l|z_{<l}, x) \sim \mathcal{N}(\mu(z_{<l}) + \Delta\mu_(z_{<l}, x), \text{diag}(\sigma^2(z_{<l}) \cdot \sigma^2(z_{<l}, x)))$$

$$p_\theta(z) = \Pi_l p_\theta(z_l|z_{<l}) \sim \mathcal{N}(\mu(z_{<l}), \text{diag}(\sigma^2(z_{<l})))$$

with μ and σ obtained by a neural network. By doing this, they define the loss function as:

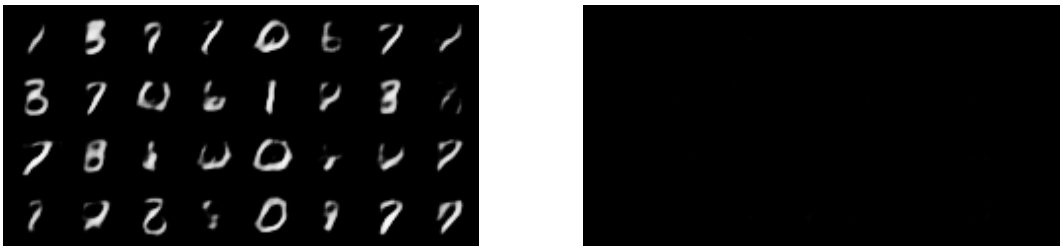
$$\mathcal{L}(x) = \mathbb{E}_q[\log p_\theta(z|x)] - KL(q_\phi(z_1|x)||p_\theta(z_1)) - \sum_{l=2}^{L} \mathbb{E}_q[KL(q_\phi(z_l|x, z_{<l}||p_\theta(z_l|z_{<l}))]$$



(a) $|\mu_1 - \mu_2| = 10$ (b) $\mu_1 - \mu_2| = 5$ (c) $|\mu_1 - \mu_2| = 4$

(d) $|\mu_1 - \mu_2| = 3$ (e) $|\mu_1 - \mu_2| = 2.5$ (f) $|\mu_1 - \mu_2| = 2$

Figure 1: Results of the Variational Inference depending on the gap between the modes



(a) Reversed KL (b) Forward KL

Figure 2: MNIST samples obtained with RKL and FKL

where L is the number of layers. By doing this, we hope to have a hierarchical representation of the data through the layers. Using the reparametrization trick, we assume that the output of each layer in the decoder follows a Gaussian distribution. Then, we try to change the loss function to:

$$\mathcal{L}(x) = \mathbb{E}_q[\log p_\theta(z|x)] - KL(q_\phi(z_1|x)||p_\theta(z_1)) - \sum_{l=2}^{L} \mathbb{E}_q[KL(p_\theta(z_l|z_{<l})||q_\phi(z_l|x, z_{<l}))]$$

where we apply the forward KL between the intermediate layers while keeping the ELBO at the top one. The calculations are done in the notebook. We try to train a small model with this loss on MNIST, unfortunately, the results are not very positive as can be seen in Figure 2. Indeed, the model is not able to sample a digit from the model, which can be explained by (1) the large dimension of data, performing variational inference with the RKL is mandatory (2) the original loss was derived from the ELBO, and consequently, there was no guarantee of getting correct samples with the FKL.

4 Discussion

This paper provides some interesting insights about commonly used frameworks and proposes a new method to overcome their limitations relying on a strong theoretical basis. However, the choice of the datasets and metrics remain surprising as those are already solved datasets and we thus wonder about the actual use cases of such a proposition. Finally, our experiment to generalize the proposed approach to more challenging cases than the ones used to illustrate the capabilities of the algorithm in the paper showed disappointing results.

5 Conclusion

In this work, we reviewed the framework proposed by Jerfel and al. and implemented it with the ambition of using it in a generative setting. Even though the forward KL is not a really popular alternative, the Franke-Wolfe algorithm, also known as boosting, has been explored for instance in [7], where tasks, described as a mixture of distributions, are learned iteratively. It could be pushed even further for example in [6] where the prior is chosen to be a multimodal mixture of Gaussians. Thus, one could imagine a boosting strategy to learn the modes of the mixture components.

References

[1] 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.

[2] 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, April 2017.

[3] Sébastien Bubeck et al. Convex optimization: Algorithms and complexity. *Foundations and Trends® in Machine Learning*, 8(3-4):231–357, 2015.

[4] Sourav Chatterjee and Persi Diaconis. The sample size required in importance sampling, 2017.

[5] Adji B. Dieng, Dustin Tran, Rajesh Ranganath, John Paisley, and David M. Blei. Variational inference via χ-upper bound minimization, 2017.

[6] Nat Dilokthanakul, Pedro AM Mediano, Marta Garnelo, Matthew CH Lee, Hugh Salimbeni, Kai Arulkumaran, and Murray Shanahan. Deep unsupervised clustering with gaussian mixture variational autoencoders. *arXiv preprint arXiv:1611.02648*, 2016.

[7] Evgenii Egorov, Anna Kuzina, and Evgeny Burnaev. Boovae: Boosting approach for continual learning of vae. *Advances in Neural Information Processing Systems*, 34:17889–17901, 2021.

[8] Ghassen Jerfel, Serena Wang, Clara Wong-Fannjiang, Katherine A Heller, Yian Ma, and Michael I Jordan. Variational refinement for importance sampling using the forward kullback-leibler divergence. In *Uncertainty in Artificial Intelligence*, pages 1819–1829. PMLR, 2021.

[9] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[10] Teun Kloek and Herman K Van Dijk. Bayesian estimates of equation system parameters: an application of integration by monte carlo. *Econometrica: Journal of the Econometric Society*, pages 1–19, 1978.

[11] Jun S Liu and Jun S Liu. *Monte Carlo strategies in scientific computing*, volume 10. Springer, 2001.

[12] Andrew C. Miller, Nicholas Foti, and Ryan P. Adams. Variational boosting: Iteratively refining posterior approximations, 2017.

[13] Kevin P. Murphy. Machine learning - a probabilistic perspective. In *Adaptive computation and machine learning series*, 2012.

[14] Art B Owen. Monte carlo theory, methods and examples, 2013.

[15] Surya T Tokdar and Robert E Kass. Importance sampling: a review. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(1):54–60, 2010.

[16] Arash Vahdat and Jan Kautz. Nvae: A deep hierarchical variational autoencoder. *Advances in neural information processing systems*, 33:19667–19679, 2020.