Variational Refinement for Importance Sampling Using the Forward Kullback-Leibler Divergence (Supplementary material)

Ghassen Jerfel[*1,2,3] **Serena Wang**[*1,2] **Clara Wong-Fannjiang**[2] **Katherine A. Heller**[1,3] **Yian Ma**[1,2,4]
Michael I. Jordan[2]

[1]Google Research, Mountain View, CA, USA
[2]University of California Berkeley, Berkeley, CA, USA
[3]Duke University, Durham, NC, USA
[4]University of California San Diego, San Diego, CA, USA

A FURTHER METHODOLOGY DISCUSSION

A.1 COMBINING HMC AND FKL VB

Using an MCMC method such as HMC can exploit the fact that the unnoramlized density r_i gets shallower and less multimodal over iterations which makes it increasingly easy to sample from once, at the beginning of each boosting iteration, via techniques such as Hamiltonian Monte Carlo [Neal, 2011]. While we saw some promising preliminary performance with this method on real data experiments using Bayesian logistic regression (BLR), the higher dimensional experiments with Bayesian neural networks (BNN) struggled with numerical instability that would require further tuning of the HMC hyperparameters which include the number of burn-in steps, learning rate, and number of leapfrog steps, among others.

A.2 STABILIZATION OF LIKELIHOOD RATIOS

To avoid high-variance gradient estimates due to a mismatch between the target and the proposal, especially at the beginning of inference, we stabilize the importance weights in an unbiased way that parallels the log-sum-exp trick [Nielsen and Sun, 2016] in order to handle potential under- or overflow when exponentiating large values:

$$r_s = \frac{p(\theta_s|x)}{q_i(\theta_s)}, \quad d_{max} = \max_s(\log p(\theta_s|x) - \log q_i(\theta_s))$$
$$d_s = exp\left(\log p(\theta_s|x) - \log q_i(\theta_s) - d_{max}\right)$$
$$w_s = \frac{d_s}{\sum_{s=1}^{S} d_s} = \frac{r_s}{\sum_{s=1}^{S} r_s}$$

As for the log residual $\log p/q$ we introduce a biased stabilization heuristic that is typical in variational boosting [Guo et al., 2016, Campbell and Li, 2019] with an $\epsilon = e^{-10}$:

$$\log\left(\frac{p(\theta|x)+\epsilon}{q_i(\theta_s)+\epsilon}\right) \approx \log\left(\frac{p(\theta|x)}{q_i(\theta_s)}\right) \tag{1}$$

A range of variance reduction schemes is applicable to our method such a weight clipping, re-sampling and re-weighting. However, we leave that for future work.

B DERIVATIONS

B.1 CONNECTING FORWARD KL TO OTHER METRICS USED FOR VI

By the monotonicity of Renyi-α divergences, given that $\lim_{\alpha\to 1} D_\alpha(p,q) = \mathrm{KL}(p\|q)$, and from [Dieng et al., 2017] :

$$\mathrm{KL}(p\|q) \leq D_2(p,q) \leq \chi^2(p,q) \tag{2}$$

Supplement for the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence (UAI 2021).

B.2 REVERSE KL REMAINDER: INTRINSIC ENTROPY REGULARIZATION

Computing the remainder-reverse KL objective using our approach leads to the well-known although usually ad-hoc entropy regularization (e.g. [Locatello et al., 2018]).

$$\mathrm{KL}(f_i \| r_i) = \mathrm{KL}(f_i \| \frac{p}{q_{i-1}}) = \mathbb{E}_{f_i}[\log \frac{f_i q_{i-1}}{p}] = \mathbb{E}_{f_i}[\log \frac{q_{i-1}}{p}] + \mathbb{E}_{f_i}[\log f_i] \tag{3}$$

As we can see, while the first term is the mean of the log-residual under the new component f_i, the typical objective for gradient boosting, the second term is the entropy of f_i.

B.3 SNIS DERIVATION

Since we do not assume to know the normalization constant of p, we shall approximate the above quantities by self-normalized importance sampling while making the distinction between the normalized p and the un-normalized $\hat{p}$:

$$\theta_s \sim q_{i-1}, \quad w^s = \frac{p(\hat{\theta}_s)}{q_{i-1}(\theta_s)}, \quad w^s_{norm} = \frac{w^s}{\sum_s w^s} \tag{4}$$

$$\mathbb{E}_{q_{i-1}}[\frac{p}{q_{i-1}} \log \frac{p}{\lambda f_i + (1-\lambda) q_{i-1}}] = \frac{\mathbb{E}_{q_{i-1}}[\frac{\hat{p}}{q_{i-1}} \log \frac{p}{\lambda f_i + (1-\lambda) q_{i-1}}]}{\mathbb{E}_{q_{i-1}}[\frac{\hat{p}}{q_{i-1}}]}$$

$$\approx \sum_s \frac{\frac{p(\hat{\theta}_s)}{q_{i-1}(\theta_s)}}{\sum_s \frac{p(\hat{\theta}_s)}{q_{i-1}(\theta_s))}} [\log \frac{p(\theta_s)}{\lambda f_i(\theta_s) + (1-\lambda) q_{i-1}(\theta_s)}] = \sum_s w^s_{norm} [\log p(\theta_s) - \log (\lambda f_i(\theta_s) + (1-\lambda) q_{i-1}(\theta_s))]$$

B.4 GRADIENTS OF MIXTURE WEIGHTS

For forward KL:

$$\nabla_{\lambda_i} \mathbb{E}_p \left[\log p - \log \sum_j^K \lambda_j q_j \right] = -\mathbb{E}_p \left[\nabla_{\lambda_i} \log \sum_j^K \lambda_j q_j \right]$$

$$= -\mathbb{E}_p \left[\frac{q_i}{\sum_j^K \lambda_j q_j} \right] = -\mathbb{E}_{q_i} \left[\frac{p}{q} \right]. \tag{5}$$

For reverse KL:

$$\nabla_{\lambda_i} \mathbb{E}_q [\log q - \log p] = \mathbb{E}_{\sum_j^K \lambda_j q_j} [\nabla_{\lambda_i} \log (\sum_j^K \lambda_j q_j)]$$

$$= \mathbb{E}_{q_i} [\log q - \log p]$$

B.5 THE FUNCTIONAL GRADIENT OF THE FORWARD KL DIVERGENCE

We assume $\operatorname{supp} p \subseteq \operatorname{supp} q$: that is, p is absolutely continuous with respect to the variational approximation q_i which can be ensured by the design of the variational family $\mathcal{Q}$.

Let $D(q) = \mathrm{KL}(p \| q)$. Functional gradient $\frac{\delta D}{\delta q}$ can be computed from the Taylor expansion of the KL functional [Friedman, 2001] as follows:

$$\lim_{\epsilon \to 0} \frac{D(q + \epsilon \cdot h) - D(q)}{\epsilon} = \int \frac{\partial D}{\partial q} h dx \tag{6}$$

$$\begin{aligned}\frac{D(q+\epsilon\cdot h)-D(q)}{\epsilon} &= \frac{1}{\epsilon}\int p\log p - p\log(q+\epsilon h)\\ &\quad + p\log p - p\log q\\ &= -\frac{1}{\epsilon}\int p\log(q+\epsilon h) - p\log q\\ &= -\frac{1}{\epsilon}\int p\log(1+\epsilon\frac{h}{q})\end{aligned}$$

We have the logarithmic inequality $\frac{x}{x+1} \leq \log(1+x) \leq x \forall x > -1$ where we can substitute $\epsilon\frac{h}{q} > 0$ for x and arrive at

$$-\frac{h}{q} \leq -\frac{1}{\epsilon}\log(1+\epsilon\frac{h}{q}) \leq -\frac{\frac{h}{q}}{1+\epsilon\frac{h}{q}}$$

By the monotone convergence theorem we can take the limit inside the integral and arrive at

$$\lim_{\epsilon\to 0}\int -p\frac{1}{\epsilon}\log(1+\epsilon\frac{h}{q}) = \int -p\frac{h}{q} \tag{7}$$

$$\frac{\delta D(q)}{\delta q} = -\frac{p}{q}$$

B.6 BOOSTING CONVERGENCE ANALYSIS

For a convex and strongly smooth functional, the greedy sequential approximation framework of [Zhang, 2003] provides an asymptotic guarantee for the convergence to a target distribution in the convex hull of the base family at a rate of $O(1/K)$ where K is the number of boosting iterations. This framework does not require each iteration to exactly solve for the optimal mixture component which can be difficult in variational inference.

While the convexity of $\mathrm{KL}(p\|q)$ in q is well established in the literature (proven with the log-sum inequality) for the forward KL divergence functional, we can show that FKL is also β-smooth in q where β depends on the maximum and minimum values that the density q can take. To establish strong smoothness, on the other hand, stricter assumptions about the densities are necessary. If we assume that all densities are bounded away from 0 and from above q_1 then for any pair of densities q_1 and q_2 there exists a $\beta = \sup\frac{p}{q_1 * q_2} \geq 0$ such that the functional gradient $\frac{\delta D}{\delta q}$ is β- Lipschitz, that is $\left|\frac{\delta D}{\delta q}(q_2) - \frac{\delta D}{\delta q}(q_1)\right| \leq \beta|q_2 - q_1|$. We can verify this choice of β:

$$\left|\frac{\delta D}{\delta q}(q_2) - \frac{\delta D}{\delta q}(q_1)\right| = \left|\frac{-p}{q_2} - \frac{-p}{q_1}\right| \tag{8}$$

$$= \left|\frac{p(q_2-q_1)}{q_2 q_1}\right| \tag{9}$$

$$= \frac{p}{q_2 q_1}|q_2 - q_1| \tag{10}$$

$$\leq \beta|q_2 - q_1| \tag{11}$$

Note that the boundedness assumptions are not unrealistic in practice and can translate to a bounded parameter space for a given family of distributions.

$$\mathrm{KL}(p\|q_i) = \mathrm{KL}(p\|\sum_i^k \lambda_i f_i) = O(1/k) \tag{12}$$

C AN ALTERNATIVE APPROACH TO FKL-BASED BOOSTING: MINIMIZING THE REMAINDER

As we seek to construct an optimal proposal through the minimization of an SNIS approximation of FKL, a trade-off arises: "should we make the distribution easier to sample from in order to minimize the SNIS variance or should we bring it closer to the target in order to improve the worst-case IS estimation error?" In particular, the closer the proposal gets to a multimodal target, the harder it may be to sample from. Therefore, this trade-off translates to two distinct approaches for the greedy additive construction of an optimal proposal mixture distribution.

The first approach described in Section 4.2 is the most straightforward as it minimizes the forward KL between the mixture q_i and the target p while holding the parameters of previously-learned mixture components fixed.

Alternatively, define the remainder distribution at iteration i as $r_i(\theta) = \frac{p(\theta|x)}{q_i(\theta)}$. A second approach is to minimize the FKL between each new component and r_i, which may be simpler with fewer modes than p:

$$\underset{f_i}{\operatorname{argmin}} \operatorname{KL}(r_i \| f_i) = \underset{f_i}{\operatorname{argmin}} \operatorname{KL}\left(\frac{p_i}{q_{i-1}} \middle\| f_i\right).$$

The mixture weight can be estimated in this scenario for each mixture component by gradient descent using the gradient with respect to FKL (see Appendix B.4).

This second approach is appealing because, at each boosting iteration, r_i becomes shallower with fewer modes which makes it easier to sample from than the multimodal proposal q_i. This approach can also be motivated by gradient boosting [Friedman, 2001] or matching pursuit [Mallat and Zhang, 1993] where one seeks to identify the mixture component that best fits the functional residual. Furthermore, this approach might be less prone to degeneracy. In fact, a derivation of this approach for the reverse KL, in Appendix B.2, identifies intrinsic entropy regularization which is often incorporated ad-hoc in similar objectives.

D ADDITIONAL SIMULATION EXPERIMENT RESULTS

Fig. 1 provides moment estimation results for the simulation with well-separated modes on a mixture of 20 2-dimensional Gaussians.

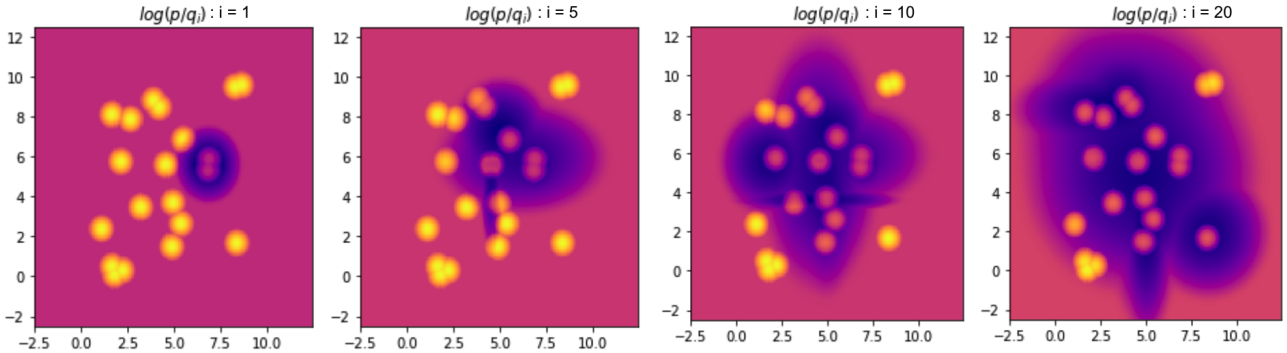

Figure 1: Evolution of (exact) FKL divergence and the mean-squared error of moment estimation (using samples from the FKL solution) on the task of estimating a 2-dimensional GMM of 20 components [Ma et al., 2019].

E ADDITIONAL REAL DATA EXPERIMENT DETAILS AND RESULTS

We provide additional details and results for the experiments on real data.

E.1 PARAMETER TRANSFORMATIONS

For the covariance matrices of each component, we optimize over the square root of diagonal matrices to ensure the non-negativity of the final diagonal covariance estimate.

To ensure non-negative or zero mixture weights, we optimize over the logits of the weights from which we recover the final weights by logistic transformation.

E.2 HYPERPARAMETERS

Hyperparameters for both the RKL and FKL boosting methods include the learning rate for the mean and learning rate for the covariance matrix when optimizing each boosting component. These were each tuned between $\{0.0001, 0.001, 0.01, 0.1\}$. The number of steps for each boosting component was tuned between 200 and 1000, and the number of samples for each

gradient computation was tuned between $\{25, 50, 100, 200\}$. The variance σ when initializing the covariance matrix for each component was tuned between $\{0.0001, 0.0005, 0.001, 0.005, 0.01\}$.

For evaluation, we draw 6000 parameter samples from the final mixture of Gaussians from which we compute the final metrics. Increasing this number to 50000 did not lead to a significant change and posed a strain on computational resources.

For the comparison to HMC, each HMC chain was initialized with a sample drawn from $\mathcal{N}(0, \sigma^2 I)$, with $\sigma = 0.01$. For all BLR tasks, the HMC comparison was run using an adaptive step size schedule with a starting step size of 1.0, 1000 burn-in steps, and 800 adaptation steps. For the BNN tasks, the HMC comparison was run using a fixed step size, tuned between $\{0.001, 0.01\}$.

E.3 DATASETS

We use four datasets from UCI, listed in Table 1. Table 1 reports the number of attributes in the dataset, or the dimensionality of the input x, but the actual dimensionality of the sampling problem in each experiment is higher than the number of attributes depending on the size of the weight vector w, and the additional variance parameters given in Sections 7.1 and 7.2. All tasks are regression tasks.

Dataset name	# attributes	# examples
wine [Cortez et al., 2009]	11	4898
boston [Harrison and Rubinfeld, 1978]	13	506
concrete [Yeh, 1998]	8	1030
power [Tüfekci, 2014]	4	9568

Table 1: Datasets used in experiments.

E.4 COMPUTATIONAL CONSIDERATIONS

In our experiments we observed that even in high dimensions, there do not seem to be significant computational differences between our method for optimizing the FKL and optimizing the RKL. The reported results use the same number of IS samples and optimization iterations for both RKL-VI and FKL-VI. In terms of wall clock time, we evaluate the highest dimensional experiment using BNNs on the Boston dataset ($d = 753$). FKL VI had a wall clock time of 783.10 seconds and RKL VI had a wall clock time of 862.93 seconds after optimizing a single boosting component for 200 gradient steps when run on a single 8-core machine with an Intel Xeon CPU @ 2.20GHz.

E.5 ADDITIONAL EXPERIMENT RESULTS

In [Miller et al., 2017], the posterior predictive distribution is simply estimated as an average over samples from the posterior $p(\theta|\mathcal{D}_{\text{train}})$ using variational boosting. In Tables 2, 3 and 4, we report the results for the RKL boosting methods where the posterior predictive distribution is computed without importance sampling (as in Eq. (15)), and is instead computed by directly averaging over samples from the variational distribution, as in Eq. (13).

$$p(y|x^*, \mathcal{D}_{\text{train}}) \approx \frac{1}{L} \sum_{l=1}^{L} p(y|x^*, \theta^{(l)}), \quad \theta^{(l)} \sim p(\theta|\mathcal{D}_{\text{train}}) \tag{13}$$

We also report results for HMC with 3 chains run in parallel. To compute the final predictive log probabilities, 2000 samples were drawn from each chain, and the predictive log probability was averaged over all 6000 combined samples using Eq. (13).

Method	Wine ($d = 14$)	Boston ($d = 16$)	Concrete ($d = 11$)	Power ($d = 7$)
HMC (3 chains)	-1.003 ($\pm$ 0.012)	-2.923 ($\pm$ 0.035)	-3.781 ($\pm$ 0.013)	-2.942* ($\pm$ 0.017)
RKL VI (no IS)	-1.003 ($\pm$ 0.012)	-2.924 ($\pm$ 0.035)	-3.780 ($\pm$ 0.013)	-2.921 ($\pm$ 0.006)
RKL VB 2 (no IS)	-1.003 ($\pm$ 0.012)	-2.923 ($\pm$ 0.035)	-3.781 ($\pm$ 0.013)	-2.994 ($\pm$ 0.004)
RKL VB 3 (no IS)	-1.003 ($\pm$ 0.012)	-2.924 ($\pm$ 0.035)	-3.781 ($\pm$ 0.013)	-2.972 ($\pm$ 0.005)

Table 2: Predictive log probabilities on test for BLR with Gaussian prior (mean $\pm$ standard error over 20 train/test splits). (*Results from only 5 train/test splits due to computational constraints.)

Method	Wine ($d = 653$)	Boston ($d = 753$)	Concrete ($d = 503$)	Power ($d = 303$)
HMC (3 chains)	-0.988 ($\pm$ 0.014)	-2.706 ($\pm$ 0.093)	-3.279 ($\pm$ 0.019)	-2.824* ($\pm$ 0.017)
RKL VI (no IS)	-0.991 ($\pm$ 0.015)	-2.858 ($\pm$ 0.019)	-3.230 ($\pm$ 0.015)	-2.850 ($\pm$ 0.009)
RKL VB 2 (no IS)	-0.990 ($\pm$ 0.015)	-2.835 ($\pm$ 0.020)	-3.231 ($\pm$ 0.015)	-2.943 ($\pm$ 0.011)
RKL VB 3 (no IS)	-0.983 ($\pm$ 0.014)	-2.753 ($\pm$ 0.015)	-3.232 ($\pm$ 0.015)	-2.997 ($\pm$ 0.011)

Table 3: Predictive log probabilities on test for BNNs with Gaussian prior (mean $\pm$ standard error over 20 train/test splits). (*Results from only 5 train/test splits due to computational constraints.)

Method	Wine ($d = 13$)	Boston ($d = 15$)	Concrete ($d = 10$)	Power ($d = 6$)
HMC (3 chains)	-1.008 ($\pm$ 0.012)	-3.085 ($\pm$ 0.056)	-3.824 ($\pm$ 0.023)	-2.942* ($\pm$ 0.017)
RKL VI (no IS)	-1.002 ($\pm$ 0.012)	-2.915 ($\pm$ 0.034)	-3.780 ($\pm$ 0.013)	-2.923 ($\pm$ 0.006)
RKL VB 2 (no IS)	-1.001 ($\pm$ 0.013)	-2.920 ($\pm$ 0.034)	-3.780 ($\pm$ 0.013)	-2.982 ($\pm$ 0.003)
RKL VB 3 (no IS)	-1.001 ($\pm$ 0.013)	-2.920 ($\pm$ 0.034)	-3.781 ($\pm$ 0.013)	-2.961 ($\pm$ 0.005)

Table 4: Predictive log probabilities on test for BLR with heavy tailed prior (mean $\pm$ standard error over 20 train/test splits). (*Results from only 5 train/test splits due to computational constraints.)