Extending Variational Boosting to Multiple Competing Variational Families

Naitong Chen

April 24, 2021

1 Introduction

In recent years, Bayesian statistical models have received a lot of attention thanks to their ability to model data of complex structures, incorporate domain expertise, and coherently quantify uncertainties around the estimates of the latent parameters. One of the key challenges of applying Bayesian statistical models in practice is to obtain samples from the often intractable posterior distribution. Markov chain Monte Carlo (MCMC) and Variational Inference (VI) are the two main approaches that address this challenge. MCMC generates posterior samples by simulating a Markov chain whose invariant distribution is the target posterior distribution. One of the main advantage of MCMC is that there is a tradeoff between computation time and the quality of the posterior samples: as more samples are generated by simulating the Markov chain for longer time, one can expect the samples to be better approximate the posterior distribution. However, MCMC does not scale well to problems with large datasets. Specifically, due to the iterative nature of MCMC, the log likelihood of each observation in the entire dataset needs to be evaluated before a single posterior sample can be generated. VI handles this scalability issue by approximating the target using some tractable distribution. More precisely, VI finds some distribution from a tractable distribution family that minimizes some divergence to the target distribution. The tractablity of the posterior approximation allows us to easily sample from the posterior distribution. However, the quality of the posterior approximation under the VI approach is fundamentally limited by the choice of the variational family. In other words, the approximation does not get closer to the true target distribution with more computation if the variational family is poorly chosen.

Inspired by the idea of boosting, the Variational Boosting (VB) algorithm proposed in [MFA17] addressed this shortcoming of the VI methods. Specifically, VB builds a mixture approximation of the target by interatively adding components from some variational family. This approach not only increases the expressibility of a fixed variational family by extending the variational family to its convex hull, but also forms the tradeoff between computation time and the quality of approximation, just like MCMC. A later paper then

confirmed that under certain conditions, as the number of components increased, the mixture approximation from VB would indeed converge to the true target distribution [Loc+18].

We note that the work in [MFA17] focused on a single variational family. Although it has been shown that the mixture approximation converges to the target distribution as the number of mixture components increase, the choice of variational family may still result in varying efficiencies of the algorithm. An example is to approximate a heavy-tailed target distribution using mixtures of Gaussian components. The light-tailedness of the Gaussian distribution may hinder the rate of convergence in terms of the number of components, requiring many mixture components to obtain reasonable approximations of the target distribution's tail behaviour. One possible way to address this limitation is to further diversify the variational family by considering multiple candidate components from different distribution families at each iteration. More concretely, at each iteration, instead of optimizing the next component to be added to the mixture approximation from a single distribution family, multiple candidate components from different distribution families can be optimized at the same time. Then the component that yields the mixture approximation with the smallest divergence to the target distribution is added.

In this report, we begin in Section 2 by reviewing the VB algorithm and two practical considerations that improve the efficiency of the method introduced in [MFA17]. Experiments conducted on synthetic data are shown in Section 3 to verify and critique the effectiveness of the VB algorithm. We then describe our proposed extension and analyze some experiment results on some synthetic data in Sections 4 and 5.

The code used to run the experiments and generate the plots can be found in `https://github.com/NaitongChen/STAT520B_Project`.

2 Variational Boosting

The variational boosting algorithm uses the mixture

$$q^{(C)}(x;\psi) = \sum_{c=1}^{C} \lambda_c q_c(x;\psi_c) \quad \text{s.t.} \quad \lambda_c \geq 0 \quad \text{and} \quad \sum_c \lambda_c = 1$$

to approximate some (possibly unnormalized) target density π, where ψ_c denotes the parameters that characterize each of the C components from some pre-specified familty of distributions and λ_c's are the corresponding mixture weights. The VB algorithm builds such mixture approximations by iteratively adding a new component to the existing approximation. Specifically, at iteration t, we find

$$\psi_t^*, \lambda_t^* \in \arg\min_{\psi_t, \lambda_t} D_{KL}\left(\lambda_t q_t(\cdot;\psi_t) + (1-q_t) q^{(t-1)}(\cdot;\psi) \| \pi\right), \tag{1}$$

With $q^t(\cdot;\psi) = \lambda_t^* q_t(\cdot;\psi_t^*) + (1-q_t^*) q^{(t-1)}(\cdot;\psi)$ becoming the updated mixture approximation. If this subproblem of component optimization can be solved efficiently, we will have greatly

exdended the single pre-specified variational family to its convex hull.

To build this mixture approximation, we can simply start by using the approximation from standard VI methods as the first component in the VB mixture approximation. Common choices include ADVI and BBVI [RGB14; Kuc+17]. The component optimization problem as shown in Eq. (1) may seem difficult to solve at the beginning. However, note that given a mixture approximation q^C, we can take advantage of the mixture structure and get

$$D_{KL}\left(q^{(C)}\|\pi\right) = \mathbb{E}_{q^{(C)}}\left[\ln q^{(C)}(x;\psi,\lambda) - \ln \pi(x)\right] = \sum_{c=1}^{C} \lambda_c \mathbb{E}_{q_c}\left[\ln q^{(C)}(x;\psi,\lambda) - \ln \pi(x)\right]. \tag{2}$$

Note the expectation is now with respect to each mixture component. If the mixture components are Gaussian, we can then apply the reparameterization trick on each component [KW13]. Then the gradient of the above expression can be estimated by sampling from a standard normal distribution and transform these samples based on each components mean and variance. This way the component optimization problem can be solved using stochastic gradient descent (SGD) methods. Since all of the existing components' means and variances stay fixed, at each iteration we only need to optimize the parameters of the component to be added along with its mixture weight. By repeating this process iteratively, we have not only increased the expressibility of our variational family but also created the desirable tradeoff between computation time and quality of approximation.

As hinted above, the Gaussian distribution family is one of the most commonly used in the context of variational boosting. However, as the dimension increases, the number of latent parameters to be optimized in the covariance matrix also increases. This can make the optimization of each component extremely difficult. [MFA17] suggested imposing a "low rank plus diagonal" structure on the covariance matrix. Namely, we let $\Sigma = D + FF^T$, where $\Sigma, D \in \mathbb{R}^{d\times d}, D$ being diagonal, and $F \in \mathbb{R}^{d\times r}$, with $\text{rank}(F) = r$. This formulation can greatly reduce the number of parameters in the covariance matrix by setting $r \ll d$. A heuristic for selecting r is also mentioned in [MFA17], whose intuition comes from approximating a single multivariate normal target. With some structure imposed on the covariance matrix, we are bound to underestimate the variance of the target distribution. Starting with $r = 1$, as we increase the rank, the marginal variances of the variational approximation should also increase. Therefore, it is suggested that, when fitting the initial component in the VB framework, we can keep increasing the rank until the change in the marginal variances are negligible. All subsequent components then share this covariance structure picked by the first component. This intuition makes sense in practice. In high dimensional data, in many cases we have dimensions that are close to marginally independent. Then ignoring some of these covariance terms can simplify the optimization without greatly hurting the quality of the approximation.

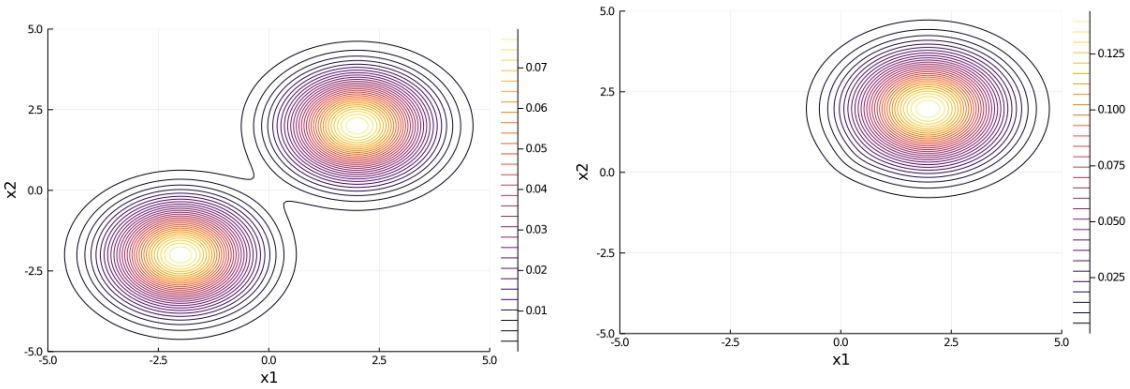
Finally, [MFA17] also addressed another common issue in the implementation of variational boosting that it is sensitive to the initialization of each component. As an example, when

approximating some bimodal target distribution, we may get stuck in one of the modes that are already well-captured if the initialization of the next component falls under this mode. Therefore, when initializing the next component, we would like to place it in a region of the target that is currently under-approimated. The suggest approach is the weighted EM algorithm. Assuming we are still using Gaussian mixture component, at iteration t, we can first sample from the current approximation of $t-1$ components before finding t clusters. With the first $t-1$ cluster's mean and variances fixed, we optimize the mean and variance, as well as the mixture weight of the tth cluster, which will be used as the initialization of the next component. By incorporating the weights, which is the ratio of the likelihoods between the target distribution and the current approximation, more importance will be placed on points that are in an under-approximated region of the target distribution.

3 Verifying Variational Boosting's Effectiveness

In this section, we use a mixture of two bivariate normal distributions as an example target distribution to verify and critique the effectiveness of the VB algorithm, as well as the two practical considerations. We begin by looking at the weighted EM initialization. From Fig. 1, we see that when approximating $\pi(x) = 0.5N\left(\begin{bmatrix}2\\2\end{bmatrix}, I_2\right) + 0.5N\left(\begin{bmatrix}-2\\-2\end{bmatrix}, I_2\right)$ indeed fails to capture the second mode. In fact, the weight of the second component, initialized through the weighted EM approach with a sample size 40, is exactly zero. While the intuition the weighted EM approach makes sense, it relies on some of the samples from the current mixture approximation producing unlikely samples. Specifically, after we have found the first mode in Fig. 1, all of the sampled observations of the current approximation stays within the mode that is already well-approximated. Then whether we can jump out of this local mode depends on that some of the samples to be used in the weighted EM algorithm comes from the edge of the already-found mode or further away from the mode. As the variance of the first component decreases, it becomes more and more unlikely that we can jump out of this local mode using the weighted EM approach. While it is a difficult problem to tackle, perhaps some improvements need to be done on the weighted EM approach for it to be more reliable.

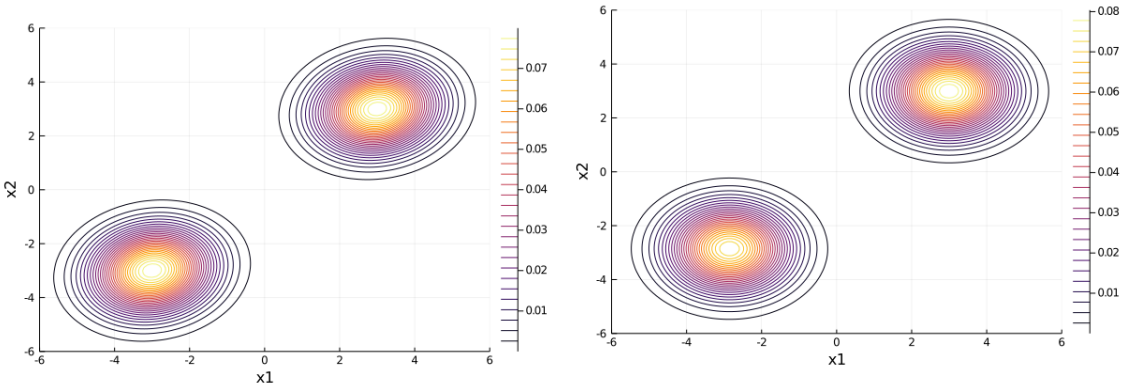
Discarding this component initialization, if initialization works in the target distribution's favour, as shown in Fig. 2, we indeed can very effectively approximate the posterior distribution. It is worth noting that a tiny 0.1 covariance has been added to the posterior in Fig. 2. However, even with just a diagonal covariance structure, the posterior approximation is still very close to the target distribution, hence confirming the intuition on selecting the structure to be imposed on the covariance matrix.



(a) True posterior distritbution

(b) Variational boosting approximating with two components

Figure 1: Comparing the target distribution and the variational boosting approximation using the weighed EM component initialization.



(a) True posterior distritbution

(b) Variational boosting approximating with two components (diagonal covariance)

Figure 2: Comparing the target distribution and the variational boosting approximation without the weighed EM component initialization.

4 Extending to Multiple Variational Families

In this section, we introduce our proposed extension in detail. All of the work so far has focused on increasing the expressibility of a single variational family through mixture approximations. However, the variational family can be further diversified by considering multiple distribution families when fitting the components of the mixture approximation. The key observation is from Eq. (2). Specifically, all of the computations on the gradient can be done component-wise. Therefore, it is possible to use different distribution families for different components. In addition, we know from [KW13] that the reparameterization trick is not limited to the Gaussian distributions. As an example, all distributions with a closed-form

inverse CDF function can be reparameterized through this function using samples from the uniform distribution.

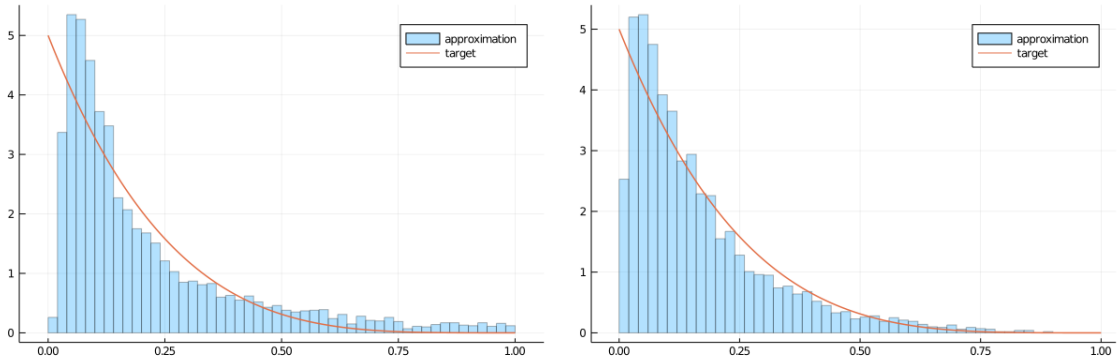
One thing to note, though, is that in the $\mathbb{E}_{q_c}\left[\ln q^{(C)}(x;\psi,\lambda) - \ln \pi(x)\right]$ term from Eq. (2), $q^{(C)}(x;\psi,\lambda)$ is the overall mixture approximation. If we were to consider multiple distribution families, this mixture approximation may contain components from multiple distribution families. As a result, the samples from the reparameterized distribution for component c needs to be able to transform to all of the distribution families in the mixture. Therefore, one requirement for extending VB to multiple variational families is to ensure that the reparameterized reference distributions have the same support. Once this is satisfied, at each iteration, we can fit a component from each of the distribution families considered, and add the one that makes the resulting mixture yield the smallest divergence to the target. Then ideally, the characteristisc of the different distribution families can be exploited to achieve the best possible fit. As a result, we may be able to obtain a good approximation of the targe using fewer components. Note that although this extension introduces more computation before adding each component, the multiple component optimization problems can be easily parallelized to remain computationally competitive against the standard variational boosting algorithm.

5 Experiments

In this section, we put our proposed extension to practice. Specifically, we use the log-normal and exponentional components to approximate mixtures of Beta distributions. The log-normal and exponential components are selected because they have the same support on non-negative real values, and both have their respective inverse CDF functions available in closed-form. From Figs. 3 and 4, we see that in this particular setting, having two competing distribution families do not lead to significantly better approximations of the posterior. We now provide some analysis on what might have caused this.

In Fig. 3, standard ADVI's are run to approximate the Beta$(1, 5)$ target using an exponential distribution and using a log-normal distribution. We see that both approximations approximates the target roughly equally well. In fact, the log-normal approximation seems to capture the tail a little bit better. However, note that both approximations fail to capture the increasing trend as the input approaches zero. This is not surprising given the shape of the log-normal distribution. However, it might have been unexpected to see the exponential distribution also failing to capture this shape, especially given that the target has a very similar shape to the exponential distribution. It is believed that this is likely due to the mapping used to convert the support of the exponential distribution from $(0, \infty)$ to $(0, 1)$. Specifically, to map $(0, \infty)$ to $(0, 1)$, the transformation $T_0(t) = \frac{1}{t+1}$ is used. As a result, the higher density region where the input is close to zero actually gets mapped to values that are among the smallest. This dip close to zero is clearly reflected in Fig. 3.

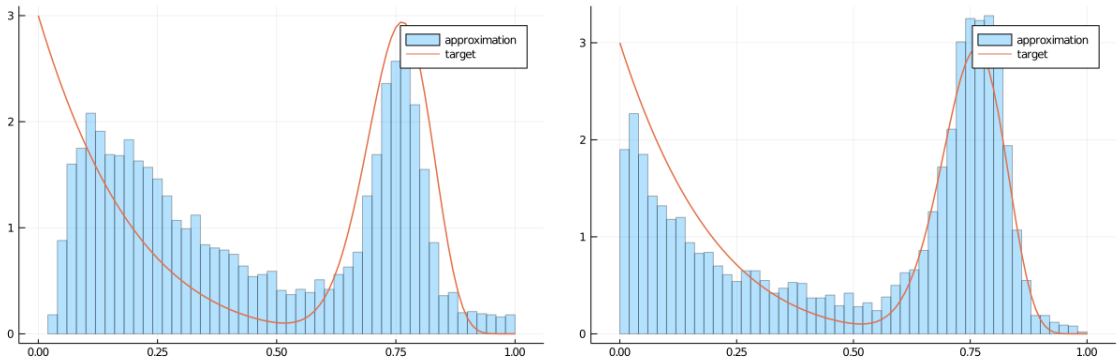
Similarly in Fig. 4, when approximating $0.5\text{Beta}(1, 6) + 0.5\text{Beta}(30, 10)$ using both types of components and only log-normal components, the two methods yield similar results. Again, using only log-normal components seem to approimxate the target distribution a little bit better. While the exponential component somewhat captures the upward trend as x approaches zero, the quality of this approximation is hindered by the transformation T_0.



(a) Approximating $\text{Beta}(1, 5)$ using a single exponential component

(b) Approximating $\text{Beta}(1, 5)$ using a single log-normal component

Figure 3: Comparing the single component approximations on a Beta distribution.



(a) Approximating a mixture of two Beta distributions using both an exponential and a log-normal component

(b) Approximating a mixture of two Beta distributions using two log-normal components

Figure 4: Comparing the two-component approximations on a Beta mixture distribution.

6 Discussion

Overall, while variational boosting generally increases the expressibility of the variational family and allows practitioners to trade more computation for better posterior approximations, it is important to address how each of the components are initialized before being

optimized. As shown in this report, failure to properly initialize components may cause the mixture approximation to be stuck in a local mode.

On the other hand, our proposed extension that further increases the expressibility of the variational family through introducing multiple component distributions do not appear to be superior compared to the standard variational boosting algorithm. However, it is believed that this is due to the poor choice of the mapping that transforms the component distribution's support. It is possible that this idea of considering multiple competing mixture components from different families is advantageous than the standard variational boosting algorithm in other settings. A direction of future work would be to identify the cases where our proposed extension outperforms the standard variational boosting. Further, it is also worth exploring the effect of the different maps that transform the support of the component distributions on our proposed extension. Finally, if considering multiple competing variational families is indeed desirable in some cases, investigations can be made on allowing the different reparameterized distributions to have non-overlapping support.

References

[KW13] D. P. Kingma and M. Welling. "Auto-encoding variational bayes". In: *arXiv preprint arXiv:1312.6114* (2013).

[Kuc+17] A. Kucukelbir et al. "Automatic differentiation variational inference". In: *The Journal of Machine Learning Research* 18.1 (2017), pp. 430–474.

[Loc+18] F. Locatello et al. "Boosting variational inference: an optimization perspective". In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2018, pp. 464–472.

[MFA17] A. C. Miller, N. J. Foti, and R. P. Adams. "Variational boosting: Iteratively refining posterior approximations". In: *International Conference on Machine Learning*. PMLR. 2017, pp. 2420–2429.

[RGB14] R. Ranganath, S. Gerrish, and D. Blei. "Black box variational inference". In: *Artificial intelligence and statistics*. PMLR. 2014, pp. 814–822.