Boosting Variational Inference With Locally Adaptive Step-Sizes

Gideon Dresdner[1], Saurav Shekhar[1], Fabian Pedregosa[2], Francesco Locatello[1], Gunnar Rätsch[1]

[1] Dept. for Computer Science, ETH Zurich, Universitätsstrasse 6, 8092 Zurich, Switzerland
[2] Google Research
dgideon@ethz.ch

Abstract

Variational Inference makes a trade-off between the capacity of the variational family and the tractability of finding an approximate posterior distribution. Instead, Boosting Variational Inference allows practitioners to obtain increasingly good posterior approximations by spending more compute. The main obstacle to widespread adoption of Boosting Variational Inference is the amount of resources necessary to improve over a strong Variational Inference baseline. In our work, we trace this limitation back to the global curvature of the KL-divergence. We characterize how the global curvature impacts time and memory consumption, address the problem with the notion of local curvature, and provide a novel approximate backtracking algorithm for estimating local curvature. We give new theoretical convergence rates for our algorithms and provide experimental validation on synthetic and real-world datasets.

1 Introduction

The central problem of Bayesian inference is to estimate the posterior distribution $p(z|X)$ of hidden variables z, given observations X, a likelihood model $p(X|z)$, and a prior distribution $p(z)$. The Variational Inference (VI) approach [Jordan *et al.*, 1999; Blei *et al.*, 2017] consists in finding the best approximation in Kullback-Leibler (KL)-divergence to the posterior from within a family of tractable densities $\mathcal{Q}$. This is posed as the following optimization problem:

$$\underset{q \in \mathcal{Q}}{\arg\min} \left\{ D^{KL}(q) \overset{\text{def}}{=} \int q(z) \ln \frac{q(z)}{p(z|X)} dz \right\} \quad (1)$$

We occasionally abuse the notation for D^{KL}: When the target distribution is omitted, as above, it is understood that the target is the true posterior distribution $p(z|X)$.

There is a trade-off between the quality of the approximation and the difficulty of the optimization problem. While a richer family may yield a better approximation of the posterior, finding such a solution requires solving a more complex optimization problem. A growing body of recent work addresses this trade-off by specifying variational families that are richer but still tractable [Rezende and Mohamed, 2015; Saeedi *et al.*, 2017; Salimans *et al.*, 2015; Saxena *et al.*, 2017; Cranko and Nock, 2019]. However, once the VI solver has converged, one cannot spend more compute to improve the approximation. If the approximation is too poor to be useful, it must be abandoned and the inference procedure restarted with a richer variational family.

The recent line of work in Boosting VI takes a different approach. Instead of specifying a richer variational family, Boosting VI greedily constructs one using mixtures of densities from a simpler base family [Guo *et al.*, 2016; Miller *et al.*, 2017; Locatello *et al.*, 2018b; Locatello *et al.*, 2018a; Cranko and Nock, 2019]. The key idea is that one can iteratively build a better approximation to the target posterior by fitting the residual parts which are not yet well-approximated.

Despite advances in making boosting agnostic to the choice of the variational family [Locatello *et al.*, 2018a], this line of work has fallen short of its potential. The reason for this is that Boosting VI does not reliably improve the variational approximation in a reasonable number of iterations [Guo *et al.*, 2016; Locatello *et al.*, 2018b; Locatello *et al.*, 2018a].

In this work, we present a new technique for determining the mixture weights of Boosting VI algorithms which improves the variational approximation in a realistic number of iterations. As we shall see, the previous convergence rates depend on two terms: a term depending on the *global* curvature of the KL and the initial error. Practitioners often focus on decreasing the latter term, but the first one can be arbitrarily large without imposing extra assumptions.

We are able to improve the dependency on the curvature in the convergence rate by tuning the mixture weights according to a quadratic function satisfying a sufficient decrease condition, i.e. is sufficiently tight on the KL divergence objective [Pedregosa *et al.*, 2020].

In the black-box VI setting, checking for exact upper bounds is not feasible due to sampling errors. Therefore, we consider the case where the estimate of the bound is *inexact*. Using this approximate local upper-bound, we develop a fast and memory efficient away-step black-box Boosting VI algorithm.

Our **main contributions** can be summarized as follows:

1. We introduce an *approximate* sufficient decrease condition

A full version of this paper including the appendix is available at https://arxiv.org/abs/2105.09240.

and prove that the resulting backtracking algorithm converges with a rate of $\mathcal{O}(1/t)$ with an improved dependency on the curvature constant.

2. We develop an away-step Boosting VI algorithm that relies on our approximate backtracking algorithm. This enables Boosting VI methods to selectively downweight previously seen components to obtain sparser solutions, thus reducing overall memory costs.

3. We present empirical evidence demonstrating that our method is both faster and more robust than existing methods. The adaptive methods also yield more parsimonious variational approximations than previous techniques.

2 Related Work

We refer to [Blei *et al.*, 2017] for a review of Variational Inference (VI). Our focus is to use boosting to increase the complexity of a density, similar to the goal of Normalizing Flows [Rezende and Mohamed, 2015], MCMC-VI hybrid methods [Saeedi *et al.*, 2017; Salimans *et al.*, 2015], distribution transformations [Saxena *et al.*, 2017], and boosted density estimation [Cranko and Nock, 2019; Locatello *et al.*, 2018c]. Our approach is in line with several previous ones using mixtures of distributions to improve the expressiveness of the variational approximation [Jaakkola and Jordan, 1998; Jerfel, 2017] but goes further to draw connections to the optimization literature. While our method does not leverage classical weak learners as in [Cranko and Nock, 2019], it does return a solution which is sampleable and is therefore more amenable to downstream Bayesian analysis.

While boosting has been well studied in other settings [Meir and Rätsch, 2003], it has only recently been applied to the problem of VI. Related works of [Guo *et al.*, 2016; Miller *et al.*, 2017] developed the algorithmic framework and conjectured a convergence rate of $\mathcal{O}(1/t)$. Later, [Locatello *et al.*, 2018b] identified sufficient conditions for convergence and provided explicit constants to the $O(1/t)$ rate. They based their analysis on the smoothness of the KL-divergence when using carefully constructed variational base families which are restrictive in practice.

In [Locatello *et al.*, 2018a], these assumptions were reduced to a simple entropy regularizer which allows for a black-box implementation of the boosting subroutine. The promise of their work is to make Boosting VI useful in practice while retaining an $\mathcal{O}(1/t)$ convergence rate.

Recent work by [Campbell and Li, 2019] and [Lin *et al.*, 2019] also explore the relationship between Boosting VI and curvature. In this paper, we present our unique view on curvature which does not require sacrificing the KL-divergence for a smooth objective as in [Campbell and Li, 2019] or fixing the number of components in the mixture [Lin *et al.*, 2019]. Finally, note that [Locatello *et al.*, 2018b] also suggests performing line-search on a quadratic approximation of the KL-divergence. Crucially, they suggest using a *global* approximation which renders their step-size arbitrarily small. See Table 1 for comparison to prior work.

Backtracking line-search and related variants are well-understood for deterministic objectives with projectionable constraints [Boyd and Vandenberghe, 2004]. However, backtracking techniques have only recently been applied to Frank-Wolfe by [Pedregosa *et al.*, 2020]. Our work lies at the intersection of these developments in VI and backtracking line-search.

3 Boosting Variational Inference

Boosting Variational Inference (VI) aims to solve an expanded version of the problem defined in Equation (1) by optimizing over the convex hull of $\mathcal{Q}$ defined as,

$$\operatorname{conv}(\mathcal{Q}) \overset{\text{def}}{=} \{\textstyle\sum_i \alpha_i q_i \mid q_i \in \mathcal{Q}, \sum_i \alpha_i = 1, \alpha_i > 0\}$$

Boosting VI algorithms take a greedy, two-step approach to solving this problem. At each iteration, first, the posterior residual p_X/q_t is approximated with an element of $\mathcal{Q}$ and added to the list of components. Then, the weights of the mixture are updated. Previous research on Boosting VI has focused on the first step — selecting the best fitting density of the residual — whereas our work takes into consideration the second step — adjusting the weights. As we shall see, step-size choice has a significant impact on both the constant in the convergence rate as well as the observed speed in practice.

Selecting the next component. Greedily approximating the residual p_X/q_t is equivalent to solving a linear minimization problem as first realized by [Guo *et al.*, 2016] and later formalized by [Locatello *et al.*, 2018b]. Our contribution (Sec. 4) can be combined with any of these approaches.

Without imposing additional constraints on the boosted Variational Inference problem, the greedy subproblem has degenerate solutions, i.e. Dirac delta located at the maximum of the residual [Locatello *et al.*, 2018b; Locatello *et al.*, 2018a; Guo *et al.*, 2016; Miller *et al.*, 2017]. This is the central challenge addressed by existing work on Boosting Variational Inference. The approach taken in [Locatello *et al.*, 2018a] is to use a generic entropy regularizer as the additional constraint. This conveniently reframes the subproblem as another KL-minimization problem which can then be fed into existing black-box VI solvers.

In their work, the greedy step can be formulated as a constrained linear minimization problem over the variational family,

$$\underset{\substack{s\in\mathcal{Q}\\ H(s)\geq -M}}{\arg\min}\ \langle s, \nabla D^{KL}(q_t)\rangle \tag{2}$$

where H is the differential entropy functional. This results in a modified overall Variational Inference objective over the entropy constrained mixtures, rather than all of $\operatorname{conv}(\mathcal{Q})$:

$$\underset{\operatorname{conv}(\overline{\mathcal{Q}})}{\arg\min}\, D^{KL}(q) \tag{3}$$

where $\overline{\mathcal{Q}} \overset{\text{def}}{=} \{s \in \mathcal{Q} \mid H(s) \geq -M\}$.

By reformulating the differential entropy constraint in Equation (2) using a Lagrange multiplier and then setting the multiplier to one, one arrives at a convenient form for the greedy subproblem (Alg. 1 line 3):

$$\underset{s\in\mathcal{Q}}{\arg\min}\, D^{KL}\left(s \,\Big\|\, \frac{p_X}{q_t}\right) = \underset{s\in\mathcal{Q}}{\arg\min}\, \underset{s}{\mathbb{E}}[\ln s] - \underset{s}{\mathbb{E}}[\ln \frac{p_X}{q_t}] \tag{4}$$

	conv. to true post.	black-box	ada. weight tuning	KL obj.	flexible num. of comps.	gen. post.
[Guo *et al.*, 2016]	✗	✗	✗	✓	✓	✓
[Miller *et al.*, 2017]	✗	✗	✓	✓	✓	✓
[Locatello *et al.*, 2018a]	✓*	✓	✗	✓	✓	✓
[Locatello *et al.*, 2018b]	✓*	✗	✗	✓	✓	✓
[Campbell and Li, 2019]	✓	✗	✓	✗	✓	✓
[Cranko and Nock, 2019]	✓	✗	✓	✓	✓	✗
[Lin *et al.*, 2019]	✗	✗	✓	✓	✗	✓
This work	✓	✓	✓	✓	✓	✓

Table 1: Comparison to previous work. *Conv. to true post.*: the paper provides asymptotic convergence guarantees ($*$) under mild conditions on the variational family $\mathcal{Q}$ such as clipped tails and initialization in the neighborhood of the optimum. *black-box*: agnostic to the form of the variational family or target distribution. *ada. weight tuning*: provides methods for tuning the mixture weights based on the quality of the components. *KL obj.*: minimizes the traditional KL-divergence VI objective. *flexible num. comps.*: does not require the user to specify the number of components in advance. *gen. post.*: the approximate posterior can be sampled from and not merely used to evaluate the probability density.

Intuitively, this objective encourages the next component to be close to the target posterior $\ln p_X$ while simultaneously being different from current iterate $\ln q_t$ and also being non-degenerate via the negative entropy term $\ln s$.

Predefined step-size. To update the mixture, [Locatello *et al.*, 2018a] take a convex combination between the current approximation and the next component (Alg. 1 line 7) with a *predefined* step-size of $\gamma_t = \frac{2}{t+2}$:

$$q_{t+1} = \left(1 - \frac{2}{t+2}\right) q_t + \frac{2}{t+2} s_t \tag{5}$$

where v_t is set to q_t.

4 Local Boosting Variational Inference

We now describe our main algorithmic contribution.

Notation. We view expectations as a special case of functional inner-products. Given two functionals, $a, b : z \mapsto \mathbb{R}$, their inner-product is $\langle a, b\rangle \overset{\text{def}}{=} \int a(z)b(z)dz$. Practically, we only encounter these integrals when one of the arguments is a density that can be sampled from thus allowing us to use Monte-Carlo: $\langle a, b\rangle \approx \widehat{\langle a, b\rangle} \overset{\text{def}}{=} \frac{1}{n}\sum_{i=1}^{k} b(\zeta_i)$ where $\zeta_i \sim a$.

Assumption. We assume that for all $\varepsilon > 0$ there exists an $n \in \mathbb{N}$ number of Monte-Carlo samples such that for all $s \in \mathcal{Q}$ and $q \in \text{conv}(\mathcal{Q})$, the Monte-Carlo approximation $\widehat{\langle s, \nabla D^{KL}(q)\rangle}$ is ε-close to the true value:

$$|\langle s, \nabla D^{KL}(q)\rangle - \widehat{\langle s, \nabla D^{KL}(q)\rangle}| \leq \varepsilon \tag{6}$$

This assumption states that we can approximate the value of the objective in Equation (2) up to a predefined tolerance.

Now, suppose we are at iteration t of the boosting algorithm (Alg. 1). q_t is the current variational approximation containing at most t components. The next component s_t is provided by line 3. v_t is then returned from the corrective components procedure (described in Sec. 4.1 and App. A). Let $d_t = s_t - v_t$ be the update direction.

Our goal is to solve the following one dimensional problem,

$$\gamma_t \in \underset{\gamma\in[0,1]}{\arg\min}\, D^{KL}(q_t + \gamma d_t) \tag{7}$$

Then we can set $q_{t+1} = q_t + \gamma_t d_t$ as described in line 7 of Algorithm 1. Solving this problem — often termed "line-search" — may be hard when no closed-form solution is available e.g. in the case of black-box VI [Locatello *et al.*, 2017; Locatello *et al.*, 2018b; Locatello *et al.*, 2018a; Pedregosa *et al.*, 2020]. In practice, general approaches such as gradient descent struggle to handle the changing curvature of the KL-divergence throughout $\text{conv}(\mathcal{Q})$ (c.f. Sec. 6).

Instead, [Pedregosa *et al.*, 2020] uses so-called Deminov-Rubinov line-search. Rather than solving the line-search problem directly, their technique defines a surrogate objective:

$$Q_t(\gamma, C) \overset{\text{def}}{=} D^{KL}(q_t) + \gamma\langle d_t, \nabla D^{KL}(q_t)\rangle + \frac{C\gamma^2}{2} \tag{8}$$

Importantly, there exists $C_t > 0$ such that for all $\gamma \in [0, 1]$, $Q_t(\gamma, C_t)$ is a valid upper bound on the line-search problem (Eq. (7)). In this case, we say that C_t satisfies the sufficient decrease condition:

$$D^{KL}(q_t + \gamma d_t) \leq Q_t(\gamma, C_t) \tag{9}$$

Essentially, we are bounding the first-order Taylor expansion of the KL-divergence at $q_t + \gamma d_t$ (as in Eq. (7)).

Unlike the finite dimensional setting described in [Pedregosa *et al.*, 2020], in black-box VI we are unable to validate the sufficient decrease condition directly because we can only approximate Q_t within a precision of ε_t,

$$|Q_t(\gamma, C) - \widehat{Q}_t(\gamma, C)| \leq \varepsilon_t \tag{10}$$

where $\widehat{Q}_t$ is the Monte-Carlo approximation to Q_t. This leads us to define an *approximate* sufficient decrease condition (c.f. Appendix B.3):

$$\widehat{Q}'_t(C, \gamma) \overset{\text{def}}{=} \widehat{D^{KL}}(q_t) + \gamma\widehat{\langle \nabla D^{KL}(q_t), d_t\rangle} + \frac{C}{2}\gamma^2 + \boldsymbol{2\varepsilon_t} \tag{11}$$

where $\varepsilon_t = \mathcal{O}(1/t^2)$. If we assume that the number of samples is large enough, then we can guarantee that

$$D^{KL}(q_t + \gamma d_t) \leq Q_t(C, \gamma) \leq \widehat{Q}'_t(C, \gamma) \tag{12}$$

Intuitively, we are adding an offset of ε_t to compensate for errors in the Monte-Carlo approximation of Q_t. As we will

Algorithm 1 Template for Boosting VI algorithms

```
Input: $q_0 \in \mathcal{Q}$, $C_{-1} > 0$
for $t = 0, 1 \ldots$ do
    $s_t = \arg\min_{s \in \mathcal{Q}} D^{KL}(s \,\|\, \frac{p_X}{q_t})$        {next component}
    $v_t, \gamma_t^{max} =$ CORRECT_COMPONENTS$(s_t, q_t)$
    $\gamma_t, C_t =$ FIND_STEP_SIZE$(q_t, s_t - v_t, C_{t-1}, \gamma_t^{max})$
    Update: $q_{t+1} = q_t + \gamma_t(s_t - v_t)$
end for
```

Algorithm 2 Find step-size with approximate backtracking

```
1:  function FIND_STEP_SIZE$(q_t, d_t, C, \gamma_t^{max})$
2:      Choose: $\tau > 1$, $\eta \leq 1$, $\varepsilon_0 > 0$, IMAX $\in \mathbb{Z}_+$
3:      Let: $g_t = -\langle \nabla D^{KL}(q_t), d_t \rangle$
4:      Set: $C \leftarrow C/\eta$, $\gamma = \min\{g_t/C, 1\}$, $i = 0$
5:      while $\widehat{D^{KL}}(q_t + \gamma d_t) > \widehat{Q}_t'(\gamma, C)$ do
6:          if $i >$ IMAX then
7:              return $\frac{2}{t+2}, C$                {predefined step-size}
8:          end if
9:          $C \leftarrow \tau C$
10:         $\gamma = \min\{g_t/C, \gamma_t^{max}\}$
11:         $i \leftarrow i + 1$
12:     end while
13:     return $\gamma, C$                        {adaptive step-size}
14: end function
```

see in Section 5, to obtain an overall convergence rate we require that ε_t decreases at each iteration. Equivalently, this requires increasing the number of samples at each step of the algorithm.

Suppose that, for some C_t, $\widehat{Q}_t'$ satisfies the approximate sufficient decrease condition. Then, $\widehat{Q}_t'(\gamma, C_t)$ can easily be minimized with respect to γ_t by setting the derivative with respect to γ equal to zero and solving:

$$\gamma_t = \min\left\{ -\frac{\widehat{\langle \nabla D^{KL}(q_t), d_t \rangle}}{C_t}, \gamma_t^{max} \right\} \tag{13}$$

where $\gamma_t^{max} \in (0, 1]$ depends on the corrective algorithm variant (c.f. Sec. 4.1). Equation (13) shows that C_t and γ_t are inversely proportional. Intuitively, the new component should be aggressively favored when it is highly correlated with the gradient of the KL-divergence since this gradient is the optimal decrease direction. We want to take advantage of these low curvature opportunities to take more aggressive steps towards improving the posterior approximation.

Therefore, our goal is to find a C_t which satisfies the approximate sufficient decrease conditions of Equation 9 while also being as small as possible. This is achieved using approximate backtracking on C (Alg. 2).

The cost of this procedure is the cost of estimating the sufficient decrease condition for each proposal of C_t. The Monte-Carlo estimates used to compute $\widehat{Q}_t'$ can be reused but the approximation to $D^{KL}(q_t + \gamma d_t)$ must be re-estimated.

Observe that in order to guarantee convergence, there must exist a global bound on C_t. This quantity is known as the *global* curvature and is defined directly as the supremum over all possible C_t's [Jaggi, 2013]:

$$C_{\mathcal{Q}} \overset{\text{def}}{=} \sup_{\substack{s \in \mathcal{Q} \\ q \in \operatorname{conv}(\mathcal{Q}) \\ \gamma \in [0,1] \\ y = q + \gamma(s - q)}} \frac{2}{\gamma^2} D^{KL}(y \,\|\, q) \tag{14}$$

Section 5.1 provides theoretical results clarifying the relationship between C_t and $C_{\mathcal{Q}}$.

We make a slight modification to the algorithm to circumvent this problem, summarized in lines (6-8) of Algorithm 2. When the adaptive loop fails to find an approximation within IMAX number of steps, it exits and simply performs a predefined step-size update. We find that this trick allows us to quickly escape from curved regions of the space. After just one or two predefined update steps, we can efficiently estimate the curvature and apply the approximate backtracking procedure. See Appendix C for results on early termination.

4.1 Correcting the Current Mixture

Not only does our work analyze and solve the problem of varying curvature in Boosting VI, it also enables the use of more sophisticated, corrective variants of the boosting algorithm. Corrective variants aim to address the problem of downweighting or removing suboptimally chosen components. This is important in the approximate regime of VI where most quantities are estimated using Monte-Carlo. However, it is impossible to apply these corrective methods to boosting without a step-size estimation method such as we describe in Section 4.

In the optimization literature, there are two corrective methods (c.f. App. A). Either one of these variants can be substituted for the CORRECT_COMPONENTS procedure. Both corrective methods begin by searching for the worst previous component, $\bar{v}$ in the sense of most closely aligning with the positive gradient of the current approximation:

$$\bar{v} = \underset{v \in \mathcal{S}_t}{\arg\max} \left\{ \langle v, \nabla D^{KL}(q_t) \rangle = \mathop{\mathbb{E}}_{v} \ln \frac{p_X}{q_t} \right\} \tag{15}$$

where $\mathcal{S}_t = \{s_1, s_2, \ldots, s_k\}$ is the current set of components. $\bar{v}$ is found by estimating each element of $\mathcal{S}_t$ using Monte-Carlo samples and selecting the argmax.

Implementation. Using the work of [Locatello *et al.*, 2018a], we perform the greedy step using existing Variational Inference techniques. For example, if $\mathcal{Q}$ is a reparameterizable family of densities, then the reparameterization trick can be used in conjugation with gradient descent methods on the parameters of $s \in \mathcal{Q}$. All the quantities necessary to compute $\widehat{Q}_t'$, γ_t, and $\bar{v}$, are estimated using Monte-Carlo.

5 Convergence Analysis

The following theorem shows a convergence rate for our algorithm. It extends the work of [Pedregosa *et al.*, 2020] to the case of Variational Inference

Theorem 1. *Let q_t be the t-th iterate generated by Algorithm 1. Let $\varepsilon_t = \frac{\varepsilon_0}{t^2}$ bound the Monte-Carlo approximation error, described in Equation* (6)*, with some initial approximation error $\varepsilon_0 > 0$. Let $\overline{C}_t \overset{def}{=} \frac{1}{t}\sum_{i=0}^{t-1} C_t$ be the average of the curvature constant estimates. Then we have:*

$$D^{KL}(q_t) - D^{KL}(q^*) \leq \frac{4(1-\delta)}{t\delta(t\delta+1)} E_0 + \frac{2\overline{C_t}}{\delta(t\delta+1)} + \frac{2\varepsilon_0}{t}$$

	Train LL	Test AUROC	Time (in s)
AdaAFW	**-0.669 ± 5.390e-04**	0.787 ± 4.599e-03	48.240 ± 2.384e+01
AdaPFW	-0.672 ± 6.340e-04	**0.791 ± 2.398e-03**	107.247 ± 1.675e+02
AdaFW	-0.671 ± 9.700e-04	0.789 ± 7.985e-03	40.870 ± 1.647e+01
Predefined*	-0.676 ± 7.435e-04	0.788 ± 7.401e-03	**7.296 ± 1.777e+00**
Line-search	**-0.669 ± 1.011e-03**	**0.791 ± 7.514e-03**	265.608 ± 1.655e+02

Table 2: Comparison of different step-size selection methods on CHEMREACT dataset. Adaptive variants have comparable AUROC values while taking less time and having less variance across multiple runs. (*) Predefined is the method proposed by [Locatello *et al.*, 2018a].

	Train LL	Test AUROC	Time (in s)
AdaAFW	**-0.169 ± 1.111e-03**	**0.859 ± 2.565e-03**	74.634 ± 3.369e+01
AdaPFW	-0.172 ± 1.361e-03	**0.857 ± 1.011e-03**	149.721 ± 1.088e+02
AdaFW	**-0.170 ± 1.774e-03**	**0.859 ± 3.672e-03**	54.334 ± 2.632e+01
Predefined*	-0.181 ± 2.983e-03	0.853 ± 3.693e-03	**21.369 ± 9.631e+00**
Line-search	-0.181 ± 2.651e-03	0.854 ± 3.473e-03	145.725 ± 1.347e+02

Table 3: Comparison of different step-size selection methods on EICU dataset. Adaptive away-steps variant gives the best test AUROC as well as training log-likelihood. (*) Predefined is the method proposed by [Locatello *et al.*, 2018a].

where $E_0 \overset{def}{=} D^{KL}(q_0) - \psi(\nabla D^{KL}(q_0))$ *is the initialization error (*ψ *denotes the dual objective) and* $\delta > 0$ *bounds the error of estimating the greedy subproblem defined in Equation* (4)*. See Appendix B for the proof.*

It is clear that the Monte-Carlo estimates must be increasingly accurate at each iteration to guarantee convergence. This can be achieved by increasing the number of samples.

The average $\overline{C}_t$ will be kept much smaller than the global curvature $C_{\mathcal{Q}}$. Previous results give rates in terms of global curvature [Locatello *et al.*, 2018a; Locatello *et al.*, 2018b; Guo *et al.*, 2016; Campbell and Li, 2019]. Without making additional assumptions, the global curvature is in principle unbounded. This explains why the number of iterations t must be large before observing the decrease in the error expected from prior work.

5.1 Discussion

The authors of [Locatello *et al.*, 2018a; Locatello *et al.*, 2018b] also provide a convergence rate for their algorithm. However, in practice this rate is never achieved. This is due to a dependence on the global curvature (Eq. (14)) of the KL-divergence which can be huge in certain areas of $\mathrm{conv}(\mathcal{Q})$. Since the inner loop of the algorithm is essentially a full run of Variational Inference, it is impossible to run it enough times to beat the global curvature and decrease the initial error as $\mathcal{O}(1/t)$.

We reduce the rate by focusing on the curvature constant which, as shown by Equation (13), is directly related to the estimation of the mixture weights. In practice, our approximate backtracking approach on the local curvature is a viable middle ground between exact line-search, which is expensive for the KL divergence, and predefined step-size which requires an infeasibly large number of iterations.

Recognize that the backtracking approach of [Pedregosa *et al.*, 2020] cannot be applied directly to the VI setting because Q_t cannot be computed exactly. Overestimation of Q_t due to approximation errors results in slower convergence. However, if C_t is underestimated, then the step-size will be overestimated which breaks the convergence proof of [Pedregosa *et al.*, 2020]. By introducing the parameter ε_t, we can provide some control over this underestimation problem as well as fully characterize the theoretical consequences.

Limitations. Determining the number of samples needed to satisfy the assumption of Section 4, namely, the number of samples needed to guarantee the convergence rate provided in Theorem 1, is a long-standing open problem. There is work on this subject ranging from pure optimization [Paquette and Scheinberg, 2018] to Importance Weighted Autoencoders [Burda *et al.*, 2015]. In this paper, our goal is to characterize the problem theoretically in terms of convergence rates. We believe that this clarifies the ongoing gap between theory and practice. Despite this gap, we found empirically that `100` Monte-Carlo samples was sufficient.

6 Empirical Validation

6.1 Instability of Line-Search

We tested the behavior of gradient-based line-search methods in a well-understood synthetic setup in which the target posterior is a mixture of two Gaussians. We set the first variational component to be the first component of the target density assuming a perfect first iteration. We then compare the performance of step $t = 1$ of the algorithm, with that of line-search and our adaptive method at step $t = 2$, over multiple runs. To measure overall performance, we use (approximate) KL-divergence to the true posterior.

Line-search becomes unstable as the dimensionality increases. This is because line-search oscillates between the two extreme values of zero and one, regardless of the quality of the next component. The quality of the next random component decreases with the dimensionality due to the curse of dimensionality. Even in medium dimensional cases, line-search is unable to distinguish between these cases. Results are summarized in Figure 1 (left).

Source code to reproduce our experiments is available here: https://github.com/ratschlab/adaptive-stepsize-boosting-bbvi

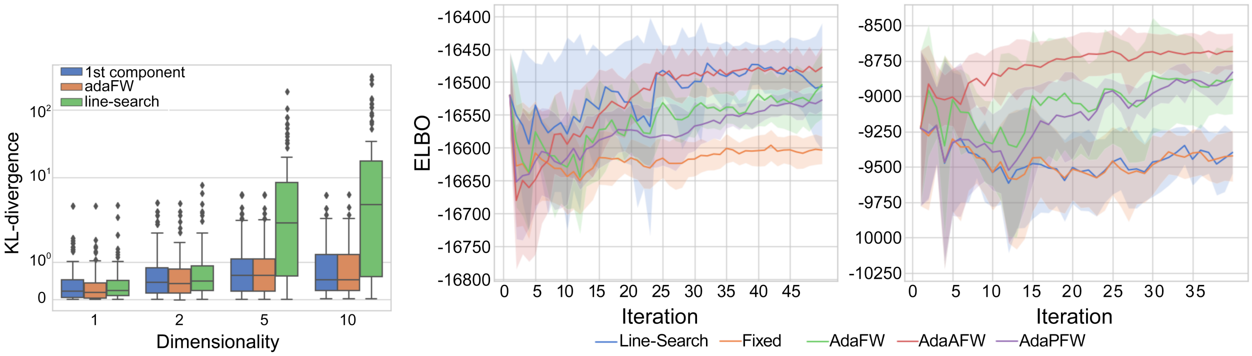

Figure 1: (left) KL-Divergence of mixture to the target distribution for different step-size variants with random LMO (lower is better). ELBO values vs Frank-Wolfe iteration for different step-size selection methods on Bayesian logistic regression task for ChemReact (center) and eICU (right) datasets (higher is better). Solid lines are mean values and shaded regions are standard deviations over different parameter configurations and 10 replicates. In both the cases, adaptive variants achieve higher ELBO and are more stable than line-search.

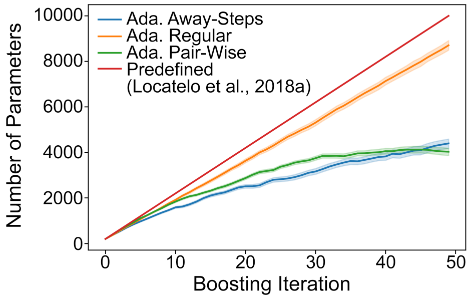

Figure 2: Comparing the number of parameters per iteration to previous work on ChemReact.

6.2 Bayesian Logistic Regression

We consider two real-world binary-classification tasks: predicting the reactivity of a chemical and predicting mortality in the intensive care unit (ICU). For both tasks we use Bayesian logistic regression. Bayesian logistic regression is a conditional prediction model with prior $p(\mathbf{w}) = \mathcal{N}(0,1)$ and conditional likelihood $p(\mathbf{y}|\mathbf{X}) = \text{Bernoulli}(p = \text{sigmoid}(\mathbf{X}^\top \mathbf{w}))$. This model is commonly used as an example of a simple model which does not have a closed-form posterior [Blei *et al.*, 2017]. We set the base family to be the Laplace distributions following [Locatello *et al.*, 2018a].

Chemical reactivity. For this task we used the ChemReact[1] dataset which contains 26,733 chemicals, each with 100 features. We ran our algorithm multiple times for 50 iterations. We selected the iterate with the best median train log-likelihood over 10 replicates in the first 50 boosting iterations (Table 2).

Mortality prediction. For this task we used the eICU Collaborative Research database [Goldberger *et al.*, 2000]. Following [Fortuin *et al.*, 2019], we selected a subset of the data with 71,366 patient stays and 70 relevant features ranging from age and gender to lab test results. We ran boosting for 40 iterations and, for each algorithm, chose the iteration which gave best median train log-likelihood over 10 replicates (Tab. 3).

In both datasets, we observe that adaptive variants achieve a better ELBO and are more stable than line-search (Fig. 1 (center and right)). This results in better AUROC and train log-likelihood (Tab. 2, 3).

Naturally, predefined step-size is the fastest since it simply sets the step-size to $2/(t+2)$. But, this results in suboptimal performance and unnecessarily large variational approximations. Our approach results in at least a 2x speed-up over line-search as well as better performance.

Adaptive variants are also faster than line-search (Tables 2 and 3). Step-size adaptivity is only 2-5 times slower than predefined step-size as opposed to line-search which is 7-39 times slower. Overall, we observe that step-size adaptivity is faster, more stable, and yields more accurate models than line-search.

6.3 Memory Efficiency

Our corrective methods discussed in Section 4.1 not only yield superior solutions in terms of accuracy, but also yield more parsimonious models by removing previously selected components. Figure 2 demonstrates this behavior on the ChemReact dataset. Appendix C has similar results on the eICU dataset.

7 Conclusion

In this paper, we traced the limitations of state-of-the-art boosting variational inference methods back to the global curvature of the KL-divergence. We characterized how the global curvature directly impacts both time and memory consumption in practice, addressed this problem using the notion of local curvature, and provided a novel approximate backtracking algorithm for estimating the local curvature efficiently. Our convergence rates not only provide theoretical guarantees, they also clearly highlight the trade-offs inherent to boosting variational inference. Empirically, our method enjoys improved performance over line-search while requiring significantly less memory consumption than a predefined step-size.

Applying this work to more complex inference problems such as Latent Dirichlet Analysis (LDA) is a promising direction for future work.

[1] http://komarix.org/ac/ds/

References

[Blei *et al.*, 2017] David M. Blei, Alp Kucukelbir, and Jon D. McAuliffe. Variational Inference: A Review for Statisticians. *Journal of the American Statistical Association*, 2017.

[Boyd and Vandenberghe, 2004] Stephen P Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge University Press, 2004.

[Burda *et al.*, 2015] Yuri Burda, Roger Grosse, and Ruslan Salakhutdinov. Importance weighted autoencoders. *arXiv:1509.00519*, 2015.

[Campbell and Li, 2019] Trevor Campbell and Xinglong Li. Universal Boosting Variational Inference. *NeurIPS*, 2019.

[Cranko and Nock, 2019] Zac Cranko and Richard Nock. Boosted Density Estimation Remastered. *ICML*, 2019.

[Fortuin *et al.*, 2019] Vincent Fortuin, Matthias Hüser, Francesco Locatello, Heiko Strathmann, and Gunnar Rätsch. SOM-VAE: Interpretable Discrete Representation Learning on Time Series. *ICLR*, 2019.

[Goldberger *et al.*, 2000] Ary L. Goldberger, Luis A. N. Amaral, Leon Glass, Jeffrey M. Hausdorff, Plamen Ch. Ivanov, Roger G. Mark, Joseph E. Mietus, George B. Moody, Chung-Kang Peng, and H. Eugene Stanley. PhysioBank, PhysioToolkit, and PhysioNet. *Circulation*, 2000.

[Guo *et al.*, 2016] Fangjian Guo, Xiangyu Wang, Kai Fan, Tamara Broderick, and David B Dunson. Boosting Variational Inference. *arXiv:1611.05559*, 2016.

[Jaakkola and Jordan, 1998] Tommi S. Jaakkola and Michael I. Jordan. Improving the Mean Field Approximation via the Use of Mixture Distributions. *Learning in Graphical Models*, 1998.

[Jaggi, 2013] Martin Jaggi. Revisiting Frank-Wolfe: Projection-Free Sparse Convex Optimization. *International Conference on Machine Learning (ICML)*, 2013.

[Jerfel, 2017] Ghassen Jerfel. Boosted Stochastic Backpropagation for Variational Inference. *Masters Thesis*, 2017. [Online; accessed 11. Sep. 2019].

[Jordan *et al.*, 1999] Michael I Jordan, Zoubin Ghahramani, Tommi S Jaakkola, and Lawrence K Saul. An introduction to variational methods for graphical models. *Machine learning*, 1999.

[Lin *et al.*, 2019] Wu Lin, Mohammad Emtiyaz Khan, and Mark Schmidt. Fast and Simple Natural-Gradient Variational Inference with Mixture of Exponential-family Approximations. *Proceedings of the 36th International Conference on Machine Learning (ICML)*, 2019.

[Locatello *et al.*, 2017] Francesco Locatello, Rajiv Khanna, Michael Tschannen, and Martin Jaggi. A Unified Optimization View on Generalized Matching Pursuit and Frank-Wolfe. In *Proc. International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2017.

[Locatello *et al.*, 2018a] Francesco Locatello, Gideon Dresdner, Rajiv Khanna, Isabel Valera, and Gunnar Rätsch. Boosting black box variational inference. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2018.

[Locatello *et al.*, 2018b] Francesco Locatello, Rajiv Khanna, Joydeep Ghosh, and Gunnar Rätsch. Boosting Variational Inference: an Optimization Perspective. *Proceedings of the 21st International Conference on Artificial Intelligence and Statistics (AISTATS 2018)*, 2018.

[Locatello *et al.*, 2018c] Francesco Locatello, Damien Vincent, Ilya Tolstikhin, Gunnar Rätsch, Sylvain Gelly, and Bernhard Schölkopf. Competitive Training of Mixtures of Independent Deep Generative Models. *Workshop at the 6th International Conference on Learning Representations (ICLR)*, 2018.

[Meir and Rätsch, 2003] Ron Meir and Gunnar Rätsch. An introduction to boosting and leveraging. In *Advanced lectures on machine learning*. Springer, 2003.

[Miller *et al.*, 2017] Andrew C Miller, Nicholas Foti, and Ryan P Adams. Variational Boosting: Iteratively Refining Posterior Approximations. *Proceedings of the 34th International Conference on Machine Learning (ICML)*, 2017.

[Nesterov, 2018] Yurii Nesterov. *Lectures on convex optimization*. Springer, 2018.

[Paquette and Scheinberg, 2018] Courtney Paquette and Katya Scheinberg. A Stochastic Line Search Method with Convergence Rate Analysis. *arXiv:1807.07994*, 2018.

[Pedregosa *et al.*, 2020] Fabian Pedregosa, Armin Askari, Geoffrey Negiar, and Martin Jaggi. Linearly Convergent Frank-Wolfe with Backtracking Line-Search. *Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics*, 2020.

[Rezende and Mohamed, 2015] Danilo Jimenez Rezende and Shakir Mohamed. Variational Inference with Normalizing Flows. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, 2015.

[Saeedi *et al.*, 2017] Ardavan Saeedi, Tejas D. Kulkarni, Vikash K. Mansinghka, and Samuel J. Gershman. Variational Particle Approximations. *Journal of Machine Learning Research*, 2017.

[Salimans *et al.*, 2015] Tim Salimans, Diederik P. Kingma, and Max Welling. Markov Chain Monte Carlo and Variational Inference: Bridging the Gap. *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, 2015.

[Saxena *et al.*, 2017] Siddhartha Saxena, Shibhansh Dohare, and Jaivardhan Kapoor. Variational Inference via Transformations on Distributions. *CoRR*, 2017.

[Tran *et al.*, 2016] Dustin Tran, Alp Kucukelbir, Adji B. Dieng, Maja Rudolph, Dawen Liang, and David M. Blei. Edward: A library for probabilistic modeling, inference, and criticism. *arXiv:1610.09787*, 2016.