Sampling with Trusthworthy Constraints: A Variational Gradient Framework

Anonymous Author(s)
Affiliation
Address
email

Abstract

Sampling-based inference and learning techniques, especially Bayesian inference, provide an essential approach to handle uncertainty in machine learning (ML). As these techniques are increasingly used in daily life, it becomes essential to safeguard the ML systems with various trustworthy-related constraints, such as fairness, safety, interpretability. Mathematically, enforcing these constraints in probabilistic inference can be cast into sampling from intractable distributions subject to general nonlinear constraints, for which practical efficient algorithms are still largely missing. In this work, we propose a family of constrained sampling algorithms which generalize Langevin Dynamics (LD) and Stein Variational Gradient Descent (SVGD) to incorporate a moment constraint specified by a general nonlinear function. By exploiting the gradient flow structure of LD and SVGD, we derive two types of algorithms for handling constraints, including a primal-dual gradient approach and the constraint controlled gradient descent approach. We investigate the continuous-time mean-field limit of these algorithms, and show they have 1) $O(1/t)$ convergence under mild conditions; 2) linear convergence if the target density is strongly log-concave. Various numerical experiments are conducted to demonstrate the efficiency of our algorithms in trustworthy settings.

1 Introduction

Efficient approximation and sampling methods of intractable distributions plays a key role in probabilistic machine learning, especially Bayesian inference. Traditionally, this problem has been framed as Monte Carlo sampling. Recently, variational optimization ideas, which frames the sampling problem into functional minimization problem of KL divergence in the space of distributions, have been popularized, as exemplified by Langevin dynamics (through the lens of Wasserstein gradient flow (Jordan et al., 1998)), and Stein variational gradient descent (Liu & Wang, 2016; Liu, 2017). Let p_0^* be a probability density function supported on $\mathbb{R}^d$ from which it is intractable to draw samples exactly; p_0^* can be the posterior distribution in Bayesian inference, or a probabilistic graphical model constructed based on expert knowledge and data information. The variational methods consider to minimize the following KL divergence objective in the space of distributions $\mathcal{P}$,

$$\min_{q \in \mathcal{P}} \mathrm{KL}(q \,||\, p_0^*), \tag{1}$$

which, if solved exactly, recovers the target distribution $q = p_0^*$. Langevin dynamics and SVGD can be viewed as solving (1) following two different types of gradient flow under different metrization of $\mathcal{P}$; in practice, both Langevin dynamics and SVGD yield a particle approximation, a set of points (a.k.a. particles) $\{\theta_i\}_{i=1}^n \subset \mathbb{R}^d$ whose empirical measure $q = \sum_{i=1}^n \delta_{\theta_i}/n$ approximates p_0^* in weak convergence; the difference is that Langevin dynamics obtains the particles with a random diffusion process, while SVGD evolves the particles with a special deterministic repulsive interacting particle system.

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

However, most existing works focus on sampling from unconstrainted domains, reflecting the fact Problem (1) is an unconstrainted optimization on $\mathcal{P}$. In many cases, especially trustworthy machine learning, we would like to impose a constraint of high practical importance in addition to approximating p_0^*, yielding a constrained variant of (1)

$$\min_{q\in\mathcal{P}} \mathrm{KL}(q \,||\, p_0^*), \qquad s.t. \quad \mathbb{E}_q[g(\theta)]\leq 0, \tag{2}$$

where we find the best approximation of p_0^* subject to a moment constraint $\mathbb{E}_q[g(\theta)] \leq 0$, with g a general nonlinear function specified by the users. If p_0^* is the posterior distribution in Bayesian inference, Problem (2) can be viewed as a form of *posterior regularization* (Zhu et al., 2014). Our framework provides a powerful tool for enabling reliable trustworthy machine learning with fairness, safety, interpretability, and other constraints, we illustrate as follows.

Fairness Constraints A motivating example is *Bayesian fairness* (Chakraborty, 2020; Ji et al., 2020; Dimitrakakis et al., 2019), in which, in addition to approximating the posterior distribution, we want to enforce equalized odds or other fairness constraints on different demographic subgroups.

Specifically, assume we are given a dataset $\mathcal{D} = \{x^{(i)}, y^{(i)}, z^{(i)}\}_{i=1}^N$ with a model $y = f(x;\ \theta)$ consisting of the feature vector $x^{(i)}$, a label $y^{(i)}$, and a protected attribute $z^{(i)}$ (e.g., male vs. female), and want to fit a prediction model $\hat{y}(x;\theta)$ with a parameter θ. In typical Bayesian inference, we are interested in finding an q to approximate the Posterior distribution $p_0^* = p(\theta \mid \mathcal{D})$. In fair Bayesian inference, we also hope q to satisfy certain fairness constraints. For example, to contorl disparate impact, one can introduce a constraint of $\mathbb{E}_q[g(\theta)] \leq 0$ with

$$g(\theta) = \ell_{fair}(\theta) - \epsilon, \qquad\qquad \ell_{fair}(\theta) = (\mathrm{cov}_{\mathcal{D}}[z,\ \hat{y}(\theta;\theta)])^2, \tag{3}$$

where we enforce the prediction $\hat{y}(x;\theta)$ is uncorrelated. Therefore, solving (2) allows us to obtain the best approximation of p_0^* subject to the fairness constraint.

Safety and Interpretability Constraints It is difficult to control and make sense of the behavior of large AI models such as deep neural networks. In many applications, such as healthcare, robotics, and AI-based systems, it is essential to add guardrails to constraint behavior of the models in a pre-specified safety region. For example, we may want to ensure that the prediction $\hat{y}(x,\theta)$ must falls inside a pre-specified interval $[\hat{y}_{0-}(x), \hat{y}_{0,+}(x)]$, which can be ensured by $\mathbb{E}_q[g(\theta)] \leq 1$ with

$$g(\theta) = \mathrm{Dist}(\hat{y}(x,\theta),\ \ [y_{0-}(x), y_{0,+}(x)]),$$

where $\mathrm{Dist}(\cdot,\ \cdot)$ is any point-set distance. Similar approach can also used to control and increase the interpretability of deep neural networks (DNNs), by enforcing that the prediction of DNNs is close to simple interpretable models such as linear classifiers and logic rules.

Prior-Agnostic Accuracy Guarantees The posterior of typical Bayesian inference depends on both the likelihood $l(\theta)$ which represents the fitness on data and a prior distribution (say p_0^*). A poor choice of prior may heavily influence the posterior and yield poor fitness. One approach to automatically limit the influence of prior is to find a q to minimize $\mathrm{KL}(q \,||\, p_0^*)$, subject to the moment constraint with $g(\theta) = -l(\theta) + \epsilon$, so that we always have a guaranteed high data fitness regardless of the choice of the prior.

Our Contribution We advance the frontier of sampling with constraints in the following ways: 1) We propose two general approaches to extend SVGD and Langevin dynamics to the constrained setting (1), including a primal-dual gradient method, and a novel constraint controlled steepest descent approach method. 2) We develop novel theoretical analysis on both methods for both convex and non-convex settings. 3) We demonstrate the power of our approaches on a variety of tasks related to trustworthy machine learning, including Bayesian fairness classification, incorporating logic rules into black-box models, and training monotonic neural networks.

2 Review on SVGD and Langevin: Sampling with Steepest Descent

We provide a unified introduction to Langevin dynamics and Stein variational gradient descent (SVGD), which can be viewed as minimizing $\mathrm{KL}(q \,||\, p_0^*)$ with two different types of gradient flow on the space of distributions $\mathcal{P}$. Assume we start from an initial density q_0 and the associated random variable $\theta_0 \sim q_0$. We consider moving the random variable θ_t along a time-dependent

vector field $\phi_t : \mathbb{R}^d \to \mathbb{R}^d$. In other words, θ_t is driven an ordinary differential equation (ODE): $\mathrm{d}\theta_t/\mathrm{d}t = \phi_t(x_t)$. Let q_t be the distribution of θ_t, which is known to follow the Fokker-Planck equation $\mathrm{d}q_t/\mathrm{d}t = -\nabla^\top(\phi_t q_t)$, where $\nabla^\top f(\theta) := \sum_{\ell=1}^d \partial_{\theta^\ell} f^\ell(\theta)$ denotes the divergence operator of a vector-valued function $f\colon \mathbb{R}^d \to \mathbb{R}^d$ with θ^ℓ and f_ℓ the ℓ-th element of θ and f, respectively; it can be viewed as treating ∇ as a vector formally.

To decrease $\mathrm{KL}(q_t \,||\, p_0^*)$ as fast as possible, we want to choose ϕ_t, from a pre-decided candidate function set $\mathcal{F}_t$, so that the decreasing rate $-\frac{\mathrm{d}}{\mathrm{d}t}\mathrm{KL}(q_t \,||\, p_0^*)$ is maximized. Note that

$$-\frac{\mathrm{d}}{\mathrm{d}t}\mathrm{KL}(q_t \,||\, p_0^*) = \mathbb{E}_{\theta\sim q_t}\left[(\nabla \log p_0^*(\theta) - \nabla \log q_t(\theta))^\top \phi_t(\theta)\right] := \mathbf{R}_{q_t,p_0^*}[\phi_t], \tag{4}$$

which is a linear functional $\mathbf{R}_{q_t,p_0^*}$ acting on $\mathcal{F}_t$. Assume $\mathcal{F}_t$ is a Hilbert space, equipped with an inner product $\langle\cdot,\cdot\rangle_{\mathcal{F}_t}$ and a norm $\|\cdot\|_{\mathcal{F}_t}$. By Riesz representation theorem, assuming $\mathbf{R}_{q_t,p_0^*}$ is a continuous linear functional, there exists an element $\mathbf{r}_{q_t,p_0^*} \in \mathcal{F}_t$, such that $\mathbf{R}_{q_t,p_0^*}[\phi] = \langle r_{q_t,p_0^*}, \phi\rangle_{\mathcal{F}_t}$. We call $\mathbf{r}_{q_t,p_0^*}$ the Riesz representation of $\mathbf{R}_{q_t,p_0^*}$ in $\mathcal{F}_t$.

Therefore, the optimal choice of ϕ_t in $\mathcal{F}_t$ can be framed into an optimization:

$$\phi_t = \operatorname*{arg\,min}_{\phi\in\mathcal{F}_t}\left\{-\langle \mathbf{r}_{q_t,p_0^*},\ \phi\rangle_{\mathcal{F}_t} + \frac{1}{2}\|\phi\|_{\mathcal{F}_t}^2\right\} = \mathbf{r}_{q_t,p_0^*}, \qquad \mathbb{D}_{\mathcal{F}_t}(q_t,p_0^*)^2 = \left\|\mathbf{r}_{q_t,p_0^*}\right\|^2, \tag{5}$$

where we add a regularization $\|\phi\|_{\mathcal{F}_t}^2$ to constrain the scale of ϕ, yielding the optimal solution of $\phi_t = \mathbf{r}_{q_t,p_0^*}$; here we defined $\mathbb{D}_{\mathcal{F}_t}(q_t,\ p_0^*)^2$ to be the corresponding maximum descending rate of KL divergence, which can be viewed as a discrepancy measure between q_t and p_0^*. In this work, we always choose $\mathcal{F}_t$ to be sufficiently large, such that $\mathbb{D}_{\mathcal{F}_t}(q_t,\ p_0^*) = 0$ implies $q_t = p_0^*$.

Langevin dynamics and SVGD can be viewed as using different Hilbert spaces $\mathcal{F}_t$, hence yielding different Riesz representation $\mathbf{r}_{q_t,p_0^*}$ for the linear operator $\mathbf{R}_{q_t,p_0^*}$.

Langevin Dynamics Taking $\mathcal{F}_t$ to be $\mathcal{L}_{q_t,2}$, the Hilbert space equipped with inner product $\langle\phi,\phi'\rangle_{\mathcal{L}_{q_t,2}} := \mathbb{E}_{\theta\sim q_t}[\phi^\top\phi']$. Then it is immediate to see from (4) that

$$\phi_t(\theta) = \mathbf{r}_{q_t,p_0^*}(\theta) = \nabla \log p_0^*(\theta) - \nabla \log q_t(\theta). \tag{6}$$

In this case, $\mathbb{D}_{\mathcal{F}_t}(q_t,\ p_0^*)$ reduces to the (square root of) Fisher divergence $\mathbb{D}_{\mathcal{F}_t}(q_t,\ p_0^*)^2 = \mathbb{E}_{\theta\sim q_t}\left[\|\nabla \log p_0^*(\theta) - \nabla \log q_t(\theta)\|^2\right]$.

Related, q_t follows the Fokker Planck equation: $\mathrm{d}q_t/\mathrm{d}t = -\nabla^\top((\nabla \log p_0^* - \nabla \log q_t)q_t) = -\nabla^\top(\nabla \log p_0^*(\theta)q_t(\theta)) + \nabla^\top\nabla q_t(\theta)$, which coincides with the density functions of the Langevin diffusion process $d\theta_t = \nabla \log p_0^*(\theta_t)dt + \sqrt{2}dW_t$, whose time-discretization yields the (unadjusted) Langevin Monte Carlo method; here dW_t is the standard Wiener process.

Stein Variational Gradient Descent (SVGD) In SVGD, we take $\mathcal{F}_t$ to the reproducing kernel Hilbert space of a positive definite and continuously differentiable kernel $k_t(\theta,\theta')$. By the reproducing property of RKHS and *Stein identity* one can show that in this case (Liu & Wang, 2016)

$$\phi_t(\cdot) = \mathbf{r}_{q_t,p_0^*}(\cdot) = \mathbb{E}_{\theta\sim q_t}[\nabla \log p_0^*(\theta)k_t(\theta,\cdot) + \nabla_\theta k_t(\theta,\cdot)], \tag{7}$$

Related, $\mathbb{D}_{\mathcal{F}_t}(q_t,p_0^*)$ reduces to kernel Stein discrepancy:

$$\mathbb{D}_{\mathcal{F}}(q_t,p_0^*)^2 = \left\|\mathbf{r}_{q_t,p_0^*}\right\|_{\mathcal{F}_t}^2 = \mathbb{E}_{\theta\sim q_t}[(\nabla_\theta \log p_0^*(\theta) + \nabla_\theta)^\top(\nabla_{\theta'} \log p_0^*(\theta') + \nabla_{\theta'})k_t(\theta,\theta')].$$

A nice property of the ϕ_t in (7), compared with the ϕ_t in (6), is that it depends on q_t only through the expectation operator $\mathbb{E}_{q_t}$, which enables a direct particle implementation of $\mathrm{d}\theta_t = \phi_t$, instead of resolving to a diffusion process. Specifically, if we initialize $q_0 = \sum_{i=1}^n \delta_{\theta_{i,0}}/n$ to be the empirical measure of a set of particles $\{\theta_{i,0}\}_i$, then q_t remains to an empirical measure, that is, $q_t = \sum_{i=1}^n \delta_{\theta_{i,t}}/n$, where the particles are evolved by

$$\frac{\mathrm{d}}{\mathrm{d}t}\theta_{i,t} = \mathbb{E}_{\theta\sim q_t}[\nabla \log p_0^*(\theta)k_t(\theta,x_{i,t}) + \nabla_\theta k_t(\theta,\theta_{i,t})], \quad \forall i = 1,\ldots,n,$$

This is a set of differential equations coupled by the *mean field* $q_t = \sum_{i=1}^n \delta_{\theta_{i,t}}/n$. Following Liu & Wang (2016), $k_t(\theta,\theta')$ can be the Gaussian RBF kernel whose bandwidth depends on the current particles $\{\theta_{i,t}\}_i$ via the median trick.

3 Main Methods

We provide two approaches to extend SVGD and Langevin dynamics to solve the constraint optimization in (2), including a primal-dual gradient method (Section 3.1), and constraint controlled steepest descent method Section 3.2). We provide theoretical analysis of both approaches for both non-convex and convex cases.

3.1 Primal-Dual Gradient Method

Introducing a Lagrange multipler λ, Problem (2) is equivalent to the following minimax problem:

$$\min_{q\in\mathcal{P}} \max_{\lambda\geq 0} \left\{ L(q,\lambda) = \mathrm{KL}(q \,||\, p_0^*) + \lambda \mathbb{E}_q[g(\theta)] \right\}. \tag{8}$$

We solve the minimax problem by alternatively updating λ and q with gradient descent. Note that we can rewrite $L(q,\lambda)$ into $L(q,\lambda) = \mathrm{KL}(q \,||\, p_\lambda^*) + \Phi(\lambda)$, with

$$p_\lambda^*(\theta) = p_0^*(\theta)\exp(-\lambda g(\theta) + \Phi(\lambda)), \quad \text{and} \quad \Phi(\lambda) = \min_{q\in\mathcal{P}} L(q,\lambda) = -\log \mathbb{E}_{\theta\sim p_0^*}[\exp(-\lambda g(\theta))],$$

and $p_\lambda^* = \arg\min_{q\in\mathcal{P}} L(q,\lambda)$. Therefore, if we want to minimize $L(q,\lambda_t)$ with a fixed λ_t, we should update q_t to move it towards $p_{\lambda_t}^*$, which can be done by following the functional gradient descent by either SVGD or Langevin dynamics: $\mathrm{d}\theta_t/dt = \mathbf{r}_{q_t,p_{\lambda_t}^*}(\theta_t)$. Fixing q, we perform standard projected gradient ascent on λ,

$$\lambda_t = \max(\lambda_{t-1} + \epsilon \mathbb{E}_{q_t}[g(\theta)],\ 0), \tag{9}$$

where ϵ is the step size. In the continuous time limit, $\epsilon \to 0$, we find the λ_t follows a projected ordinary differential equation. It can be written jointly with the density q_t of θ_t as a primal-dual gradient flow (PDGF):

$$\frac{\mathrm{d}q_t}{\mathrm{d}t} = -\nabla\cdot(\phi_t q_t) = -\nabla\cdot(\mathbf{r}_{q_t,p_{\lambda_t}^*} q_t), \quad \frac{\mathrm{d}\lambda_t}{\mathrm{d}t} = [\eta \mathbb{E}_{q_t}[g]]_{\lambda_t,+}, \tag{10}$$

with $[a]_{\lambda,+} = \mathbb{I}(\lambda \leq 0)\max(a,0) + \mathbb{I}(\lambda > 0)a$. Again, $\mathbf{r}_{q_t,p_{\lambda_t}^*}$ is defined by (6) for the Langevin dynamics, and (7) for the SVGD, as long as p_0^* is replaced by $p_{\lambda_t}^*$.

Numerical Implementation

In order to implement (10) through a particle-based algorithm with step size ϵ, we need to update λ_t with (9). We also need to approximate q_t with empirical measure $\sum_{i=1}^n \delta_{\theta_{i,t}}/n$, where $\{\theta_{i,t}\}_{i=1,\ldots,n}$ are the particles at time t. Following our discussion in Section 2, $\theta_{i,t}$ should be updated by:

$$\text{SVGD:} \quad \theta_{i,t+1} = \theta_{i,t} + \epsilon \mathbb{E}_{\theta\sim q_t}[(\nabla \log p^*(\theta) - \lambda_t \nabla g(\theta))k(\theta,\theta_{i,t}) + \nabla_\theta k_t(\theta,\theta_{i,t})], \tag{11}$$

$$\text{Langevin:} \quad \theta_{i,t+1} = \theta_{i,t} + \epsilon(\nabla \log p_0^*(\theta_{i,t}) - \lambda_t \nabla g(\theta_{i,t})) + \sqrt{2\epsilon}\xi_{i,t}, \tag{12}$$

where $\{\xi_{t,i}\}$ are i.i.d. standard Gaussian noise. Note that in (12), we run n parallel chains of Langevin dynamics, which are coupled through sharing the common λ. The empirical distribution of $\{\xi_{t,i}\}$ is used to approximate $\mathbb{E}_{q_t}[g(\theta)]$ in (19). This is summarized in Algorithm 1. The version containing more detailed formulation can be found in Algorithm 3.

Convergence Analysis

One problem of primal-dual gradient method is that the algorithm does not try to meet constraint directly. Instead, it tries to minimize $\mathrm{KL}(q \,||\, p_\lambda^*)$ which involves the constraint. So in order to prove convergence of algorithm, we need to assume that $\mathrm{KL}(q \,||\, p_\lambda^*)$ and its maximum descending rate $\mathbb{D}_{\mathcal{F}}(q,\ p_\lambda^*)$ dominate the constraint function g in the following sense.

Assumption 3.1. *There exists a positive constant $c_1, c_2, c_3 < \infty$, such that for any $t \geq 0$,*

$$(\mathbb{E}_{q_t}[g] - \mathbb{E}_{p_{\lambda_t}^*}[g])^2 \leq c_1 \mathbb{D}_{\mathcal{F}_t}(q_t,\ p_{\lambda_t}^*)^2 \tag{13}$$

$$\mathrm{KL}(q_t \,||\, p_\lambda^*) \leq c_2 \mathbb{D}_{\mathcal{F}_t}(q_t \,||\, p_{\lambda_t}^*)^2 \tag{14}$$

$$(\mathbb{E}_{q_t}[g] - \mathbb{E}_{p_{\lambda_t}^*}[g])^2 \leq c_3 \mathrm{KL}(q_t \,||\, p_{\lambda_t}^*). \tag{15}$$

Algorithm 1 Primal-Dual Method

Initialize the particles $\{\theta_{i,0}\}_{i=1}^n$ and λ_0.
for iteration t **do**
  Update the particles $\{\theta_{i,t}\}_{i=1}^n$ by Eq. (11) (`SVGD`), or Eq. (12) (`Langevin`).
  Update λ_t by Eq. (9).
end for

Algorithm 2 Constraint Controlled Method

Initialize the particles $\{\theta_{i,0}\}_{i=1}^n$.
for iteration t **do**
  If `SVGD`, update $\{\theta_{i,t}\}_{i=1}^n$ by Eq. (11), and then update λ_t by Eq. (20)
  If `Langevin`, update $\{\theta_{i,t}\}_{i=1}^n$ Eq. (12), and then Update λ_t by Eq. (19).
end for

Condition (14) is equivalent to the log-Sobolev inequality in the Langevin case (when $\mathcal{F}_t = \mathcal{L}_{q_t,2}$ and $\mathbb{D}_{\mathcal{F}}(\cdot,\cdot)^2$ is Fisher divergence), which holds when $\log p_\lambda^*$ is a bounded perturbation of a strongly concave function (Holley & Stroock, 1986). In particular, it holds when p_0^* is strongly log-concave and g is convex. So it can be seen as a convexity (concavity) assumption.

Condition (15) is a mild condition, which always hold when g is a bounded function as implied by Pinsker's inequality (See Lemma A.1 of Cui & Tong (2021)). Condition (13) is implied by (14)+(15), and also holds if g can be written into a form of $g = a + \nabla \log p_\lambda^*(\theta)^\top \psi + \nabla^\top \psi$ for some $\psi \in \mathcal{F}_t$ and $\|\psi\|_{\mathcal{F}_t}^2 \le c_1$ (see Lemma B.1 in Appendix).

Assumption 3.2. *There exists* $0 < v_0 < \infty$ *such that* $v_0 \le \mathrm{var}_{\theta\sim p_{\lambda_t}^*}[g(\theta)]$ *for* $\forall t \ge 0$.

This is a mild assumption, and holds when g is bounded and is not a constant on the support of $p_{\lambda_t}^*$.

Our analysis is based on the following Lyapunov function for the minimax problem on $L(q,\lambda)$:

$$E(q,\lambda) = L(q,\lambda) - 2\Phi(\lambda) + \max_{\lambda\ge 0} \Phi(\lambda).$$

It has the following properties.

Lemma 3.3. *We have* $E(q,\lambda) \ge 0$*, and* $E(q,\lambda) = 0$ *iff* (q,λ) *is a saddle point of* $L(q,\lambda)$.

Under mild conditions, we can show $\mathbb{E}(q_t,\lambda_t)$ is decaying along PDGF. Moreover, we can find a certain solution that meets the KKT requirement approximately.

Theorem 3.4. *If* (13) *holds and* $0 < \eta < 1/(2c_1)$*, then* $E(q_t,\lambda_t)$ *decreases monotonically following the primal-dual gradient flow in* (10)*,*

$$-\frac{\mathrm{d}}{\mathrm{d}t}E(q_t,\lambda_t) \ge \Delta(q_t,\lambda_t) := (1-2c_1\eta)\mathbb{D}_{\mathcal{F}_t}(q_t,p_{\lambda_t}^*)^2 + \frac{1}{2}\eta(\mathbb{E}_{q_t}[g])^2 \times \mathbb{I}(\lambda > 0 \ \ or \ \ \mathbb{E}_{q_t}[g] > 0).$$

Also (10) *finds an approximate solution with an* $O(1/T)$ *rate:* $\min_{t\in[0,T]} \Delta(q_t,\lambda_t) \le \frac{1}{T}E(q_0,\lambda_0), \forall T \ge 0$.

Note that $\Delta(q,\lambda)$ is a measure of optimality. This is because $\Delta(q,\lambda) = 0$ implies the KKT condition holds, that is $q = p_\lambda^*$, $\mathbb{E}_q[g] \le 0$ and $\lambda\mathbb{E}_q[g] = 0$. If strong convexity (or strong log-concavity) conditions are assumed, we can further show linear convergence to the optimal solution:

Theorem 3.5. *Suppose problem* (8) *has a solution* (q^*,λ^*)*, and Assumptions 3.1 and 3.2 hold. The PDGF following* (10) *with* $0 < \eta < 1/(2c_1)$ *will converge to* q^* *linearly in* KL *divergence:*

$$\mathrm{KL}(q_t||q^*) \le \alpha_1 E(q_t,\lambda_t) \le \alpha_1 \exp(-\alpha_2 t)E(q_0,\lambda_0),$$

where $\alpha_1 = \max\{1+\frac{1}{2}c_3, 1+\frac{1}{2v_0}\}$ *and* $\alpha_2 = \min\{\frac{1}{c_2}(1-c_1\eta), \frac{1}{2}\eta v_0\}$.

3.2 Constraint Controlled Gradient Descent

The results of the primal-dual gradient method can be sensitive to the initialization λ_0 and the learning rate η of λ. We now propose a "constraint controlled" method, which finds a constrained variant of steepest descent direct ϕ_t that yields the steepest descent on the KL objective like Section 2, while ensuring that the solution converges to the feasible set rapidly when it is violated.

Following Section 2, assume we update q_t by $\mathrm{d}\theta_t = \phi_t(\theta_t)\mathrm{d}t$, and want to decide the optimal ϕ_t. To solve the constrained optimization (2), in additional to maximizing the descending rate of KL divergence as Section 2, we also want to ensure that constraint $\mathbb{E}_{q_t}[g]$ is properly controlled, such that 1) if the constraint is not met (i.e., $\mathbb{E}_{q_t}[g] \geq 0$), we should monotonically decrease $\mathbb{E}_{q_t}[g]$, and 2) after the constraint is met (i.e., $\mathbb{E}_{q_t}[g] \leq 0$), we should monotonically descend the loss $\mathrm{KL}(q_t, p_0^*)$, while ensuring that the solution stays within the constraint set. Note that

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbb{E}_{q_t}[g] = \mathbb{E}_{q_t}[\nabla g(\theta)^\top \phi_t(\theta)] := \mathbf{S}_{q_t,g}[\phi_t] = \langle \mathbf{s}_{q_t,g},\ \phi_t\rangle_{\mathcal{F}_t},$$

where $\mathbf{S}_{q_t,g}$ is a linear operator on $\mathcal{F}_t$, whose Riesz representation is assumed to be $\mathbf{s}_{q_t,g}$. Generalizing (5), we propose to set ϕ_t to be the solution of

$$\phi_t = \operatorname*{arg\,min}_{\phi\in\mathcal{F}_t} \left\{ -\langle \mathbf{r}_{q_t,p_0^*},\ \phi\rangle_{\mathcal{F}_t} + \frac{1}{2}\|\phi\|_{\mathcal{F}_t}^2 \quad s.t. \quad \langle \mathbf{s}_{q_t,g},\ \phi\rangle_{\mathcal{F}_t} \leq -\alpha\mathbb{E}_{q_t}[g] \right\}, \tag{16}$$

where $\alpha \geq 0$ is a control coefficient. This ensures that, if the constraint is not met (i.e. $\mathbb{E}_{q_t}[g] > 0$), the constraint is descending ($\frac{\mathrm{d}}{\mathrm{d}t}\mathbb{E}_{q_t}[g] \leq -\alpha\mathbb{E}_{q_t}[g] < 0$); If the constraint is met ($\mathbb{E}_{q_t}[g] \leq 0$), then it allows $\mathbb{E}_{q_t}[g]$ to increase, but with a rate smaller than $-\alpha\mathbb{E}_{q_t}[g]$, which decreases towards zero when the solution approaches the constraint boundary $\{q\colon \mathbb{E}_q[g] = 0\}$; this on one hand allows us the have the flexibility to choose ϕ to decrease $\mathrm{KL}(q \,||\, p_0^*)$), and on the other hand ensures the solution is confined inside the constraint set. The high-level idea here is similar to the control barrier functions in control theory (Ames et al., 2019), which works in completely different settings. Note that the constraint in (16) can be viewed as a linearization of the constaint $\mathbb{E}_q[g] \leq 0$ around q_t, as $\alpha\mathbb{E}_{q_{t+1/\alpha}}[g] = \alpha\mathbb{E}_{q_t}[g] + \frac{\mathrm{d}}{\mathrm{d}t}\mathbb{E}_{q_t}[g] = \alpha\mathbb{E}_{q_t}[g] + \langle \mathbf{s}_{q_t,g},\ \phi\rangle_{\mathcal{F}_t}$, while the loss in (Ames et al., 2019) can be viewed as a simple qudratic approximation of $\mathrm{KL}(q \,||\, p_0^*)$. Therefore, (16) can be viewed as a functional variant of sequential qudratic programming (Nocedal & Wright, 2006).

Using the Lagrange duality of (16), it is easy to derive that (the details can be found in Lemma B.2) $\phi_t = \mathbf{r}_{q_t,p_0^*} - \lambda_t \mathbf{s}_{q_t,g} = \mathbf{r}_{q_t,p_{\lambda_t}^*}$, with

$$\lambda_t = \max\left(\frac{\alpha\mathbb{E}_{q_t}[g] + \langle \mathbf{r}_{q_t,p_0^*},\ \mathbf{s}_{q_t,g}\rangle_{\mathcal{F}_t}}{\|\mathbf{s}_{q_t,g}\|_{\mathcal{F}_t}^2},\ 0 \right). \tag{17}$$

Therefore, the density following the constraint controlled gradient flow (CCGF) should satisfy:

$$\frac{\mathrm{d}q_t}{\mathrm{d}t} = -\nabla\cdot(\phi_t q_t) = -\nabla\cdot(\mathbf{r}_{q_t,p_{\lambda_t}^*} q_t). \tag{18}$$

This is similar to the PDGF (10) as q_t is driven towards $p_{\lambda_t}^*$. The difference is that the λ_t in PDGF is learnt by the algorithm; but CCGF has the explicit formula for λ_t. Intuitively, CCGF will be more efficient.

Next, we obtain Langevin and SVGD variants by taking $\mathcal{F}_t$ to be $\mathcal{L}_{q_t,2}$ and RKHS, respectively.

Langevin Dynamics and its implementation Assume $\mathcal{F}_t = \mathcal{L}_{q_t,2}$. We have $\mathbf{r}_{q_t,p_0^*} = \nabla\log p_0^* - \nabla\log q_t$ and $\mathbf{s}_{q_t,g} = \nabla g$. Hence $\phi_t = \nabla\log q - \lambda_t\nabla g - \nabla\log q_t$, where

$$\lambda_t = \max\left(\frac{\alpha\mathbb{E}_{q_t}[g] + \mathbb{E}_{q_t}[(\nabla\log p_0^*)^\top\nabla g + \nabla^\top\nabla g]}{\mathbb{E}_{q_t}[\|\nabla g\|^2]},\ 0 \right). \tag{19}$$

The evolution of q_t associated with $\mathrm{d}\theta = \phi_t(\theta)\mathrm{d}t$ is standard Fokker Planck equation associated with $p_{\lambda_t}^*$, which can be realized by the same parallel Langevin particle dynamics in (12), when setting $q_t = \sum_{i=1}^n \delta_{i,t}/n$. Note that λ_t depends on q_t only through the expectation $\mathbb{E}_{q_t}[\cdot]$, which can be replaced by the empirical average of the particles. See Algorithm 2 and the detailed version Algorithm 4

The SVGD Case Let $\mathcal{F}_t$ be the RKHS of kernel $k_t(\theta,\theta')$. Then it is easy to show that $\mathbf{s}_{q_t,g}(\cdot) = \mathbb{E}_{\theta\sim q_t}[\nabla_\theta g(\theta)k_t(\theta,\cdot)]$. Therefore, $\phi_t(\cdot) = \mathbb{E}_{\theta\sim q_t}[(\nabla\log p(\theta) - \lambda_t\nabla g(\theta))k_t(\theta,\cdot) + \nabla_\theta k_t(\theta,\cdot)]$,

$$\lambda_t = \max\left(\frac{\alpha\mathbb{E}_{\theta\sim q_t}[g(\theta)] + \mathbb{E}_{\theta,\theta'\sim q_t}[\nabla_{\theta'} g(\theta')^\top(\nabla\log p_0^*(\theta) + \nabla_\theta)k_t(\theta,\theta')]}{\mathbb{E}_{\theta,\theta'\sim q_t}[\nabla g(\theta)^\top\nabla g(\theta')k_t(\theta,\theta')]},\ 0 \right), \tag{20}$$

Similar to regular SVGD, we set $q_t = \sum_{i=1}^n \delta_{\theta_{i,t}}/n$, which is iteratively updated with SVGD updates with $p_{\lambda_t}^*$ as the target distribution as (11). Also, λ_t again only depends on $\mathbb{E}_{q_t}[\cdot]$, which should be evaluated with the empirical mean of the particles. See Algorithm 2 and the detailed version Algorithm 4.

Convergence Analysis

Since CCGF tries to meet the constraint explicitly, we have the following attractive properties:

Theorem 3.6. *Suppose* $\|\mathbf{s}_{q_t,g}\|_{\mathcal{F}_t} \neq 0$ *in the CCGF* (18)*. Then with* ϕ_t *defined in* (17)*. We have*

$$\mathbb{E}_{q_t}[g] \leq \exp(-\alpha t)\mathbb{E}_{q_0}[g], \quad \forall t \geq 0.$$

If $\mathbb{E}_{q_{t_0}}[g] \leq 0$ *for some* $t_0 \geq 0$*, then* $\mathbb{E}_{q_t}[g] \leq 0$ *and* $\frac{d}{dt}\mathrm{KL}(q_t \,||\, p_0^*) \leq 0$ *for all* $t \geq s$*.*

Note that the requirement $\|\mathbf{s}_{q_t,g}\|_{\mathcal{F}_t} \neq 0$ is necessary for the λ_t in CCGF to be well defined. If fact, if $\|\mathbf{s}_{q_t,g}\|_{\mathcal{F}_t} = 0$, $\frac{d}{dt}\mathbb{E}_{q_t}[g] = 0$ along any descent direction ϕ. And if $\mathcal{F}_t = \mathcal{L}_{q_t,2}$, one can show g is a constant function q_t-a.s.. In other words, this is an ill-posed scenario where local method like CCGF has no chance to solve. An assumption slightly stronger than $\lambda_t \neq \infty$ would be assuming it is bounded from above:

Assumption 3.7. *There exists an upper bound* $\lambda_+ < \infty$*, such that for all time* t *when the constraint is not satisfied (i.e.,* $\mathbb{E}_{q_t}[g] > 0$*), we have*

This is mild regularity condition, which holds, for example, if $\mathbb{E}_{q_t}[g]$ and $\|\mathbf{r}_{q_t}\|$ are upper bound, and $\|\mathbf{s}_{q_t,g}\| > c_- > 0$. We show that, under a mild regularity condition, our algorithm meets the KKT condition with a $O(1/t)$ rate. Further, if Condition (14) holds, then $\mathrm{KL}(q_t \,||\, q^*)$ converges to zero exponentially fast. Note that the conditions here are milder than that of the primal-dual method (which additionally requires Condition (13), (15) and Assumption 3.2).

Theorem 3.8. *Let* $\{q_t\}$ *be the density function of* $\mathrm{d}\theta_t = \phi_t(\theta)\mathrm{d}t$ *with* ϕ_t *defined in* (17)*. Under Assumption 3.7, we have*

$$\min_{t\in[0,T]} \Delta_2(q_t, \lambda_t) := \mathbb{D}_{\mathcal{F}_t}(q_t,\ p_{\lambda_t}^*)^2 + \alpha(-\lambda_t \mathbb{E}_{q_t}[g])_+ \leq \frac{1}{T}\left(\mathrm{KL}(q_0, p_0^*) + \lambda_{\max,+}(\mathbb{E}_{q_0}[g])_+\right).$$

Note that, given that $\mathbb{E}_{q_t}[g] \leq 0$ (see Theorem 3.6), $\Delta_2(q_t, \lambda_t) = 0$ implies the rest of KKT condition (i.e., $q_t = p_{\lambda_t}^*$ and $(-\lambda_t \mathbb{E}_{q_t}[g])_+ = 0$).

Similar to Theorem 3.5, we can also show linear convergence to optimal solution under convexity assumption, Condition (14):

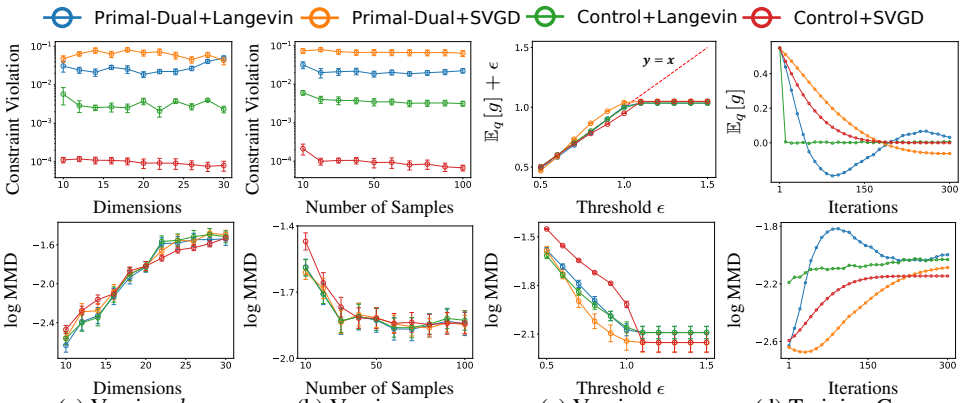
Theorem 3.9. *Suppose problem* (8) *has a solution* (q^*, λ^*)*,* $\|\mathbf{s}_{q_t,g}\|_{\mathcal{F}_t} \neq 0$*, and Condition* (14) *also holds. Then the CCGF following* (18) *with* $\alpha \leq 1/c_2$ *will converge to* q^* *linearly in* KL *divergence:*

$$\mathrm{KL}(q_t \,||\, q^*) \leq e^{-\alpha t}\mathrm{KL}(q_0 \,||\, q^*).$$

4 Experiments

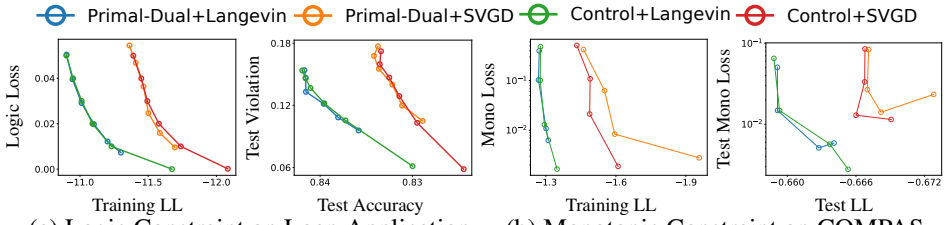
We start with testing our method on a Gaussian mixture toy, and then proceed to a number of thrustworthy-related applications, including incorporating logic rules in Bayesian logistic regression, introducing monotonicity constraints in Bayesian neural networks, and learning fair Bayesian neural networks. In all cases, we find that our methods can reliable enforce the constraints while achieving high prediction performance. The controlled-based methods tend enforce the constraint more strictly and faster.

Algorithms and Settings We summarize the tested methods and the unconstrained baselines here: `Langevin`: Vanilla Langevin dynamics. We run n parallel chains of Langevin dynamics for sampling; `SVGD`: Vanilla SVGD with codes from (Liu & Wang, 2016); `Primal-Dual+Langevin`: Langevin dynamics with Primal-dual gradient descent; `Primal-Dual+SVGD`: SVGD with Primal-dual gradient descent; `Control+Langevin`: Langevin dynamics with constraint controlled gradient descent ; `Control+SVGD`:SVGD with constraint controlled gradient descent. For the SVGD methods, we follow the configuration in (Liu & Wang, 2016). We use the RBF kernel with bandwidth chosen by the standard median trick, and adopt Adagrad (Duchi et al., 2011) for optimization. The step size for the Lagrangian multiplier in primal-dual methods is chosen from $\{0.001, 0.01, 0.1, 1, 10, 100, 1000\}$ to reach the smallest constraint loss in each experiment.



(a) Varying d Fixed $n = 50, \epsilon = 0.5$ (b) Varying n Fixed $d = 20, \epsilon = 0.5$ (c) Varying ϵ Fixed $d = 20, n = 50$ (d) Training Curve Fixed $n = 50, \epsilon = 0.5$

Figure 1: Results on Gaussian mixture models. Averaged over 10 runs



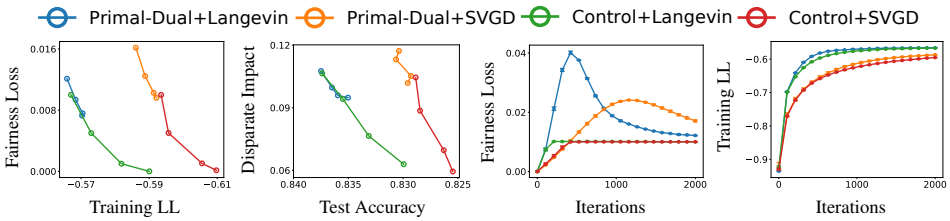
(a) Logic Constraint on Loan Application (b) Monotonic Constraint on COMPAS

Figure 2: Trade-off curve with different ϵ. 'LL': log-likelihood. 'Mono': monotonicity. Note that the x-axis are flipped so larger values are on the left.

Gaussian Mixtures We set p_0^* to be randomly generated $\mathbb{R}^d$-Gaussian mixtures (see Appendix), and $g(\theta) = \|\theta\|^2 - \epsilon$, which constrains the second order moment of the samples . Figure 1 shows the result as we vary the dimension d of the domain (Figure 1(a)), the number of particles n (Figure 1(b)), the threshold value ϵ (Figure 1(c)), and iteration step ((Figure 1(d))). The violation of constraint is measured by $(\mathbb{E}_q[g])_+$; the optimization result is measured by the maximum mean discrepancy, $\mathrm{MMD}(q,\ p_0^*)$, between q and p_0^*, because the KL divergence $\mathrm{KL}(q \,||\, p_0^*)$ is not computable when q is an empirical measure of particles. We can see that all the methods achieve similar $\mathrm{MMD}(q,\ p_0^*)$, but `Control+SVGD` tends to satisfy the constraint best on (a) and (b).

Embedding Logic Rules into Black-box Model ML models can provide accurate predictions but are difficult to interpret and control explicitly. Our method can be used to enforce the ML model to be consistent with a set of logic rules to improve the interpretability and safety. We illustrate this with a loan classification problem of predicting whether to lend loans to a specific applicant. We impose the two logic rules: (1) an applicant must be denied if she has the lowest credit rank and not employed; (2) an applicant must be approved if she has the highest credit rank and has been employed over 15 years. To apply our method, we set p_0^* to be the typical posterior distribution of Bayesian logistic regression, and define the constraint to $g(\theta) = \ell_{logic}(\theta) - \epsilon$, with $\ell_{logic}(\theta) = \mathbb{E}_{(y,x)\sim\mathcal{D}_{logic}}[\mathrm{Loss}(y, \hat{y}(x;\theta))]$, where $\mathcal{D}_{\text{logic}}$ is the uniform distribution on the (x, y) that satisfy the logic constraints, Loss is the classification loss and $\hat{y}(x;\theta)$ is the prediction by the Bayesian logistic regression model. We vary the threshold ϵ in $\{0.001, 0.01, 0.02, 0.03, 0.04, 0.05\}$ and find the corresponding training log-likelihood (LL). In combination, they are plotted as a trade-off curve shown in in Figure 2(a, left). In Figure 2(a, right), we plot the trade-off curve of the testing accuracy vs. the constraint violation $((\mathbb{E}_q[\ell_{logic}] - \epsilon)_+)$ on the testing data. We can see that the controlled-based methods tend to enforce the constraints better (reaching lower values in logic loss) in both training and testing set, and the Langevin methods achieves higher test accuracy.

Training Monotonic Bayesian Neural Networks In some applications, it can be desirable to enforce the ML prediction to be monotonic w.r.t certain attributes (Karpf, 1991; Sill, 1998). For example, when predicting admission decisions, a fair ML system must admit a student with higher



(a) Trade-off Curve with Different ϵ (b) Training Curve

Figure 3: Experiment results on learning fair Bayesian neural networks. 'LL': log-likelihood. Note that in (a) the x-axis are flipped so larger values are on the left.

GPA over the one with lower GPA, given that they are identical in the other features. We apply our method to enforce monotonicity in Bayesian neural networks. We use the COMPAS dataset following the setting in (Liu et al., 2020). In this case, p_0^* is the posterior of a Bayesian neural network on the data, and g is defined as $g(\theta) = \ell_{mono}(\theta) - \epsilon$ with $\ell_{mono}(\theta) = \mathbb{E}_{x\sim\mathcal{D}}\left[(-\nabla_x \hat{y}(x;\theta))_+\right]$. By varying ϵ in $\{0.0001, 0.01, 0.1, 0.5\}$, we plot in Figure 2(b) the trade-off curve of LL vs. the monotonic loss on training/testing set, which follow similar trends as Figure 2(a).

Training Fair Bayesian Neural Networks Following the discussion in the introduction, we apply our method to train fair Bayesian neural networks. We use the constraint defined in (3), and use the *Adult Income* dataset (Kohavi, 1996), which is a classification problem of predicting whether the annual income of a person is $\geq$ \$50,000, with gender as the protected attribute. For the experiment, we follow the setting in Martinez et al. (2020); Liu & Vicente (2020). With $\epsilon = \{0.0001, 0.001, 0.005, 0.01\}$, Figure 3(a, left) shows the trade-off curve of training LL vs. fairness loss, and Figure 3(a, right) shows the testing accuracy vs. CV score (Calders & Verwer, 2010), a standard measure of disparate impact. Figure 3(b) shows an example of training curves. We can again see that the control based methods enforce the constrain more strictly and in a faster speed.

5 Related Works, Limitations, Conclusion

Related Works Although sampling on unconstrained domain has been extensively studied, efficient algorithms for constrained settings are largely lacking. A body of works have developed to extend Langevin dynamics and SVGD on Riemannian manifold (Girolami & Calderhead, 2011; Patterson & Teh, 2013; Liu & Zhu, 2018); these methods, however, rely on computationally efficient characterization of the manifolds and as a result does not work efficiently with constraints specified by general nonlinear inequalities. Projection-based methods (e.g, Sen et al., 2018) can also be developed for sampling on constrained spaces, but again only work for simple constraints with efficient projection maps. Note that our method considers a different setting when the constraint is applied on the population (rather than individual particles) and works for general nonlinear inequality constraints. In terms of problem formulation, our constrained optimization in (2) coincides with the posterior regularization framework of Zhu et al. (2014); but we focus on particle-based inference based on SVGD and Langevin, which was not studied before.

Limits, Impacts, Future Directions Our new methods cast a full spectrum of theoretical questions that can not be completed with a single work. In the current work, we mainly focus on the infinite particle limit $n \to +\infty$ when q_t is a smooth probability density. Future works will study the guarantees with finite particle size n.

On the practical side, since our methods leverage SVGD and Langevin dynamics, they necessarily inherent their limits, such as the sensitivity on the choice of kernel and step sizes. Another practical issue is that we can only enforce constraints on the training set, and hence generalization error need to be considered if we need to strictly enforce constraints on the testing set. Also, this work focuses on a single constraint for simplicity, which is sufficient for most practical applications since different constraints can be combined into a single constraint easily. Extensions to multiple constraints are straightforward but will be studied separately.

In terms of social impact, our method aims to impose trustworthy constraints in Bayesian sampling, and can help increase the reliability, fairness, and interpretability of ML systems applied to daily life. We will release code free online to promote the applications of our approach, although it does open the possibility of malicious use of our techniques for adversarial purposes.

References

Ames, A. D., Coogan, S., Egerstedt, M., Notomista, G., Sreenath, K., and Tabuada, P. Control barrier functions: Theory and applications. In *2019 18th European Control Conference (ECC)*, pp. 3420–3431. IEEE, 2019.

Calders, T. and Verwer, S. Three naive bayes approaches for discrimination-free classification. *Data Mining and Knowledge Discovery*, 21(2):277–292, 2010.

Chakraborty, I. Bayesian sampler: Fairness in sampling. In *2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 246–251. IEEE, 2020.

Cui, T. and Tong, X. T. A unified performance analysis of likelihood-informed subspace methods. *arXiv preprint arXiv:2101.02417*, 2021.

Dimitrakakis, C., Liu, Y., Parkes, D. C., and Radanovic, G. Bayesian fairness. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 509–516, 2019.

Duchi, J., Hazan, E., and Singer, Y. Adaptive subgradient methods for online learning and stochastic optimization. *The Journal of Machine Learning Research*, 12, 2011.

Girolami, M. and Calderhead, B. Riemann manifold langevin and hamiltonian monte carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(2):123–214, 2011.

Holley, R. and Stroock, D. W. Logarithmic sobolev inequalities and stochastic ising models. 1986.

Ji, D., Smyth, P., and Steyvers, M. Can i trust my fairness metric? assessing fairness with unlabeled data and bayesian inference. *arXiv preprint arXiv:2010.09851*, 2020.

Jordan, R., Kinderlehrer, D., and Otto, F. The variational formulation of the Fokker–Planck equation. *SIAM journal on mathematical analysis*, 29(1), 1998.

Karpf, J. Inductive modelling in law: example based expert systems in administrative law. In *Proceedings of the 3rd international conference on Artificial intelligence and law*, pp. 297–306, 1991.

Kohavi, R. Scaling up the accuracy of naive-bayes classifiers: A decision-tree hybrid. In *Kdd*, volume 96, pp. 202–207, 1996.

Liu, C. and Zhu, J. Riemannian stein variational gradient descent for bayesian inference. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Liu, Q. Stein variational gradient descent as gradient flow. *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Liu, Q. and Wang, D. Stein variational gradient descent: A general purpose Bayesian inference algorithm. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2016.

Liu, S. and Vicente, L. N. Accuracy and fairness trade-offs in machine learning: A stochastic multi-objective approach. *arXiv preprint arXiv:2008.01132*, 2020.

Liu, X., Han, X., Zhang, N., and Liu, Q. Certified monotonic neural networks. *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Martinez, N., Bertran, M., and Sapiro, G. Minimax pareto fairness: A multi objective perspective. In *International Conference on Machine Learning*, pp. 6755–6764. PMLR, 2020.

Nocedal, J. and Wright, S. *Numerical optimization*. Springer Science & Business Media, 2006.

Patterson, S. and Teh, Y. W. Stochastic gradient riemannian langevin dynamics on the probability simplex. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2013.

Sen, D., Patra, S., and Dunson, D. Constrained inference through posterior projections. *arXiv preprint arXiv:1812.05741*, 2018.

Sill, J. Monotonic networks. In *Advances in neural information processing systems*, pp. 661–667, 1998.

Zhu, J., Chen, N., and Xing, E. P. Bayesian inference with posterior regularization and applications to infinite latent svms. *The Journal of Machine Learning Research*, 15(1):1799–1847, 2014.

Checklist

The checklist follows the references. Please read the checklist guidelines carefully for information on how to answer these questions. For each question, change the default **[TODO]** to [Yes] , [No] , or [N/A] . You are strongly encouraged to include a **justification to your answer**, either by referencing the appropriate section of your paper or providing a brief inline description. For example:

- Did you include the license to the code and datasets? [Yes] See Section **??**.
- Did you include the license to the code and datasets? [No] The code and the data are proprietary.
- Did you include the license to the code and datasets? [N/A]

Please do not modify the questions and only use the provided macros for your answers. Note that the Checklist section does not count towards the page limit. In your paper, please delete this instructions block and only keep the Checklist section heading above along with the questions/answers below.

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] See section 2.
   (b) Did you describe the limitations of your work? [Yes] See section 5.
   (c) Did you discuss any potential negative societal impacts of your work? [N/A] Our work provides new ways to improve fairness.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] They are listed in Section 3.
   (b) Did you include complete proofs of all theoretical results? [Yes] The proofs are in the appendix.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] See supplementary.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See supplementary.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] See supplementary.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See supplementary.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] See section 4.
   (b) Did you mention the license of the assets? [Yes]
   (c) Did you include any new assets either in the supplemental material or as a URL? [No] We found these assets online.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A] The data does not contain identifiable information or offensive content

5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A] Our research does not involve human subjects
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A] Our research does not involve human subjects
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A] Our research does not involve human subjects