Machine Learning and Neural Networks (MATH3431) Epiphany term, 2023

Handout 4: Bayesian Learning with Stochastic gradient Langevin dynamics

Lecturer & author: Georgios P. Karagiannis georgios.karagiannis@durham.ac.uk

Aim. To introduce the stochastic gradient descent (motivation, description, practical tricks, analysis in the convex scenario, and implementation).

Reading list & references:

- Welling, M., & Teh, Y. W. (2011). Bayesian learning via stochastic gradient Langevin dynamics. In Proceedings of the 28th international conference on machine learning (ICML-11) (pp. 681-688).

This is subject to minor changes that will be decided based on the Lecture. It will be finalized around 1 day after the Lecture. It is given as guide before the lecture.

1. Motivations in Bayesian learning

Note 1. Stochastic gradient Langevin dynamics (SGLD) algorithm can be used to facilitate Bayesian learning in high-dimensional and/or large scale dataset problems. SGLD produces samples from a posterior distribution of parameters based on available data, in the limit; like Langevin dynamics. SGLD is an iterative optimization algorithm which uses mini-batches to create a stochastic gradient estimator; like SGD.

Remark 2. Consider a Bayesian statistical model with sampling distribution (statistical model) $f(z|\theta)$ labeled by an unknown parameter $w \in \Theta \subseteq \mathbb{R}^d$ that follows a prior distribution $f(w)$. Assume a dataset $\mathcal{S}_n = \{z_i; i = 1, ..., n\}$ of size n. Let $L_n(w) = f(z_{1:n}|w)$ denotes the likelihood of the observables $\{z_i \in \mathcal{Z}\}_{i=1}^n$ for a parameter value of θ. The Bayesian model is

$$\begin{cases} z_i|w & \sim f(z_i|w) \\ w & \sim f(w) \end{cases} \tag{1.1}$$

Given a posterior distribution

$$f(\theta|z_{1:n}) = \frac{L_n(w) f(w)}{\int L_n(w) f(w) \, \mathrm{d}w} \tag{1.2}$$

with (often) a computationally intractable density function, we are interested to compute the posterior expectation of a function h defined on W

$$\mathrm{E}_f(h(w)|z_{1:n}) = \int h(w) f(w|z_{1:n}) \, \mathrm{d}w \tag{1.3}$$

in order to perform inference. In real world problems, the integral (1.3) cannot be analytically computed, and hence we have to resort to numerical methods.

Remark 3. Monte Carlo integration aims at approximating (1.3), by using Central Limit Theorem or Law of Large Numbers arguments as $\hat{h} \approx \mathrm{E}_f\left(h\left(w\right)|z_{1:n}\right)$ where

$$\hat{h} = \frac{1}{T}\sum_{t=1}^{T} h\left(w^{(t)}\right)$$

where $\left\{w^{(t)}\right\}$ are T simulations drawn (approximately) from the posterior distribution 1.2. This theory is subject to conditions we skip.

Remark 4. Stochastic gradient Langevin dynamics (SGLD) algorithm is able to generate a sample distributed according to the posterior distribution (1.2) (approximately in the limit) as required to perform Monte Carlo. It is particularly suitable in read world problems where the conventionality of w is high and the size of the data-set (number of examples) n is huge (big data).

2. The Algorithm

Note 5. SGLD is a framework for learning from large scale datasets based on iterative learning from small mini-batches. It relies on adding the right amount of noise to a standard stochastic gradient optimization algorithm, such that the produced chain converges to samples from the true posterior distribution as anneal the stepsize properly reduces.

Note 6. We assume that the likelihood $L_n\left(w\right)$ is differentiable for $w \in \Theta$, for simplicity.

Algorithm 7. *Stochastic Gradient Langevin Dynamics (SGLD) with learning rate $\eta_t > 0$, batch size m, and temperature $\tau > 0$ is*

(1) *Compute*

$$w^{(t+1)} = w^{(t)} + \eta_t\left(\frac{n}{m}\sum_{i\in J^{(t)}} \nabla \log f\left(z_i|w\right) + \nabla \log f\left(w\right)\right) + \sqrt{\eta_t}\sqrt{\tau}\epsilon_t$$

where $\epsilon_t \sim \mathrm{N}\left(0,1\right)$

(2) *Iterate until a termination criterion is satisfied; E.g., $t \leq T_{max}$ for a prespecified $T_{max} > 0$.*

Remark 8. SGLD (Algorithm 7) generates a random chain $\left\{w^{(t)}\right\}$ that is asymptotically distributed according to a distribution with density such as

$$f_\tau\left(w|z_{1:n}\right) \propto \exp\left(\frac{1}{\tau}\prod_{i=1}^{n} f\left(z_i|w\right) f\left(w\right)\right) \tag{2.1}$$

$$\propto \exp\left(\frac{1}{\tau}L_n\left(w\right) f\left(w\right)\right) \tag{2.2}$$

Hence for $\tau = 1$ we may simulate from the posterior (1.2).

Remark 9. The output of SGLD (Algorithm 7) is the last few iterations $\left\{w^{(t)}; t \geq T_0\right\}$, because we need to discard the first T_0 iterations from the output as burn in. This is done because at the first stage the chain of w_t generated by SGLD is trying to converge (aka to reach areas where with substantial posterior density).

Remark 10. Expectation (1.3), aka

$$\mathrm{E}_f\left(h\left(w\right)|z_{1:n}\right)=\int h\left(w\right)f\left(w|z_{1:n}\right)\mathrm{d}w$$

can be estimated as an arithmetic average

$$\widehat{h_T\left(w\right)}=\frac{1}{T-T_0}\sum_{t=T_0+1}^{T}h\left(w^{(t)}\right) \tag{2.3}$$

as by LLN $\widehat{h_T\left(w\right)}\to\mathrm{E}_f\left(h\left(w\right)|z_{1:n}\right)$

Remark 11. Another more efficient estimator for the expectation (1.3) is the the weighted arithmetic average

$$\widehat{h\left(w\right)}=\sum_{t=T_0+1}^{T}\frac{\eta_t}{\sum_{t=T_0+1}^{T}\eta_{t'}}h\left(w^{(t)}\right) \tag{2.4}$$

3. Examples