MiniBatch Monte Carlo simulation

Chernov Andrey

May 9, 2021

Bayesian Inference for black box models

- Compute posterior using Bayes Rule:

$$p(\mathbf{w} \mid \mathcal{D}) = \frac{p(\mathbf{w})p(\mathcal{D} \mid \mathbf{w})}{p(\mathcal{D})}$$

  Using MCMC to calculate $p(w|D)$.

- Make predictions using the posterior predictive distribution:

$$p(t_{test} \mid x_{test}, \mathcal{D}_{train}) = \int p(\mathbf{w} \mid \mathcal{D}_{train}) p(t_{test} \mid x_{test}, \mathbf{w}) \mathrm{d}\mathbf{w}$$

$$\approx \sum_i w_i p(t|x, w_i)$$

  Where $w_i \sim p(\mathbf{w} \mid \mathcal{D}_{train})$

MCMC

- Markov chain Monte Carlo (MCMC) is a family of methods comprise a class of algorithms for sampling from the desired probability distribution.
- How it works in practice
  - Construct a Markov chain that has the desired distribution as its stationary distribution
  - Sample from a Markov chain while It wiil not converge to stationary distribution (Warm up the chain)
  - Now, you got the "black box", which samples from the desired distribution

Metropolis-Hastings Algorithm

- **The task** is a generate samples from distribution $p(T)$, known up to a normalization constant.

$$p(T) = \frac{1}{Z}\tilde{p}(T)$$

- **The algorithm**
  - The sample θ_{t+1} at time $t+1$ is generated using a candidate θ' from a (simpler) proposal distribution $q\left(\theta' \mid \theta_t\right)$, filtered by an acceptance test.
  - The acceptance test has acceptance probability: $\alpha\left(\theta_t, \theta'\right) = \frac{p\left(\theta'\right)q\left(\theta_t|\theta'\right)}{p(\theta_t)q(\theta'|\theta_t)} \wedge 1$
  - if $u < \alpha\left(\theta_t, \theta'\right)$ $\theta_{t+1} = \theta'$, otherwise set $\theta_{t+1} = \theta_t$. Where $u \sim \mathcal{U}(0, 1)$

We must pass each point from dataset for one acceptance test

For Bayesian inference, the target distribution is $p\left(\theta \mid x_1, \ldots, x_N\right)$. The acceptance probability is now:

$$\alpha\left(\theta, \theta'\right) = \frac{p_0\left(\theta'\right) \prod_{i=1}^{N} p\left(x_i \mid \theta'\right) q\left(\theta \mid \theta'\right)}{p_0(\theta) \prod_{i=1}^{N} p\left(x_i \mid \theta\right) q\left(\theta' \mid \theta\right)} \wedge 1$$

It is not scalable for big datasets.

Let us introduce the following notation

$$\Lambda_i(\theta, \hat{\theta}) = \log \frac{p\left(x_i|\hat{\theta}\right)}{p(x_i|\theta)} \quad \Lambda(\theta, \hat{\theta}) = \sum_{i=1}^{n} \Lambda_i$$
$$\psi(\theta, \hat{\theta}) = \log \frac{p(\theta)q(\hat{\theta}|\theta)}{p(\hat{\theta})q(\theta|\hat{\theta})} \quad \Delta(\theta, \hat{\theta}) = \Lambda(\theta, \hat{\theta}) - \psi(\theta, \hat{\theta})$$

The unbiased estimation for Λ и Δ

$$\Lambda^*(\theta, \hat{\theta}) = \frac{n}{b} \sum_{i=1}^{b} \log \frac{p\left(x_i^* \mid \hat{\theta}\right)}{p\left(x_i^* \mid \theta\right)}$$
$$\Delta^*(\theta, \hat{\theta}) = \Lambda^*(\theta, \hat{\theta}) - \psi(\theta, \hat{\theta})$$
$$x_i^* \sim \mathcal{U}\left[\{x_1, \ldots, x_n\}\right]$$

Barker Lemma

Lemma 2. If $g(s)$ is any function such that $g(s) = \exp(s)g(-s)$, then the sampling with acceptance function $\alpha(\theta, \theta') \triangleq g(\Delta(\theta, \theta'))$ get desired distribution.

The sigmoid function satisfies this lemma. Indeed,

$$\exp(s)g(-s) = \frac{\exp(s)}{1 + \exp(-s)} = \frac{1}{\exp(-s) + 1} = g(s)$$

MiniBatch Metropolis-Hastings Algorithm

$$\alpha(\theta, \hat{\theta}) \wedge V, \quad V \sim \mathcal{U}[0, 1]$$

Apply Barker's lemma

$$g(\Delta(\theta, \hat{\theta})) \wedge V, \quad V \sim \mathcal{U}[0, 1]$$

Require the monotonicity of the function g.

$$\Delta(\theta, \hat{\theta}) \wedge g^{-1}(V) = X_{log}$$

Using CLT for Δ

$$\Delta^* \sim \mathcal{N}\left(\Delta, \sigma^2\left(\Delta^*\right)\right) = \Delta + \mathcal{N}\left(0, \sigma^2\left(\Delta^*\right)\right)$$

If $\sigma^2(\Delta^*)) < Var(X_{log}) = \dfrac{\pi^2}{3}$, we can decompose X_{log} as

$$X_{log} = \mathcal{N}\left(0, \sigma^2(\Delta^*))\right) + X_{corr}$$

MiniBatch Metropolis-Hastings Algorithm

Thus, we accept the new point, if:

$$\Delta + X_{log} > 0$$

$$\Delta + X_{\mathrm{log}} = (\Delta + X_{\mathrm{norm}}) + X_{\mathrm{corr}} = \Delta^* + X_{\mathrm{corr}} > 0$$

where $X_{norm} \sim \mathcal{N}\left(\Delta, \sigma^2\left(\Delta^*\right)\right)$

MiniBatch Metropolis-Hastings Algorithm

Algorithm 2 Minibatch Metropolis Hastings acceptance test

Input: $\hat{\theta}$, sampled from $q\left(\hat{\theta} \mid \theta^{t}\right)$. and previous point θ^{t}

Output: New point θ^{t+1} from the desired distribution.

$$\Delta^{*}=0, \sigma^{2}\left(\Delta^{*}\right)=\infty$$

while $\sigma^{2}\left(\Delta^{*}\right)>\sigma^{2}$ do
Sample new batch $\left\{x_{i}^{*}\right\}_{1}^{b}$ from $\mathcal{D}: x_{i}^{*} \sim \mathcal{U}\left[\left\{x_{1}, \ldots, x_{n}\right\}\right]$. Recalculated $\Delta^{*}, \sigma^{2}\left(\Delta^{*}\right)$, using $\left\{x_{i}^{*}\right\}_{1}^{b}$ end while

$$X_{\text{corr}} \sim C_{\sigma}^{*}, X_{\text{norm}} \sim \mathcal{N}\left(0, \sigma^{2}-\sigma^{2}\left(\Delta^{*}\right)\right)$$
$$\theta^{t+1}=\begin{cases}\hat{\theta}, & \text{if } \Delta^{*}+X_{\text{norm}} \leq X_{\text{corr}} \\ \theta^{t}, & \text{if } \Delta^{*}+X_{\text{norm}}>X_{\text{corr}}\end{cases}$$

Approximate logit distribution using normal distribution

logit PDF:

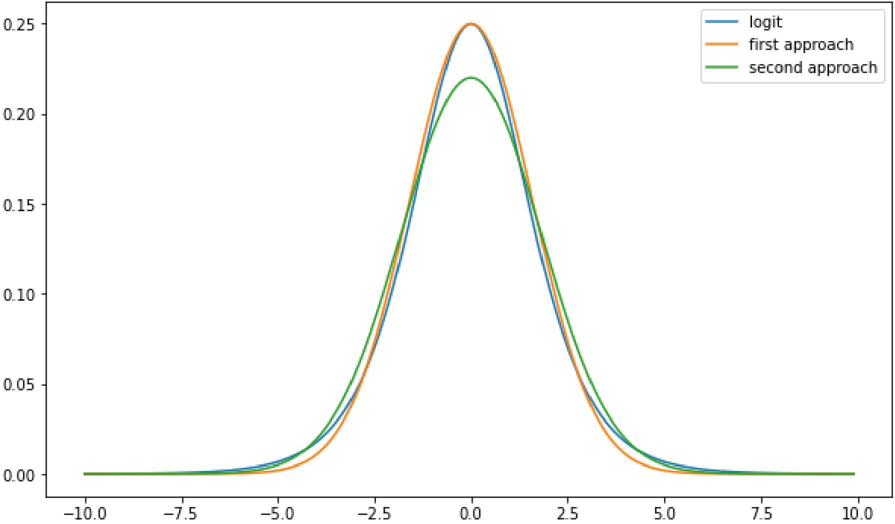
$$f(x) = \frac{\exp(-x)}{(1 + exp(-x))^2}$$

$$f(0) = \frac{1}{4}$$

$$\phi(0) = \frac{1}{\sqrt{2\pi}\sigma_{prox}}$$

First Idea: Find σ_{prox} from equation $\phi(0) = f(0)$, $\sigma_{prox} = \frac{4}{\sqrt{2\pi}}$

Second Idea: $argmin_{\sigma_{prox}} KL(f(x)||\phi(x))$; $\sigma_{prox} = \frac{\pi}{\sqrt{3}}$



0.25
0.20
0.15
0.10
0.05
0.00
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0
logit
first approach
second approach

OK, It does not work anyway

```
// (Part 3.2) Abnormally good or bad minibatches.
else if (math.abs(numStd) > 5.0) {
  if (opts.verboseMH) {
    println("\tCASE 1: math.abs(numStd) = " +math.abs(numStd))
  }
  newMinibatch = true
  if (numStd > 0) {
    accept = true
  }
}
// (Part 3.3) If sample variance is too large, don't do anything.
else if (sampleVariance >= targetVariance) {
  if (opts.verboseMH) {
    println("\tCASE 2: sample >= target = "+targetVariance)
  }
}
```

For MNIST problem with one fully connected layer $\sigma(\Delta^*) > 500$ on **full** dataset

The Idea to fix problem with variance

Let's $g(s) = \dfrac{K}{1 + exp(-s)}$. It is still satisfy Barker Lemma.
But It is incorrect $g(s)$, because $g(+\inf) = K$

$$\Delta + K * X_{log} > 0$$

$$\frac{\Delta}{K} + X_{\mathrm{log}} = \left(\frac{\Delta}{K} + X^{*}_{\mathrm{norm}}\right) + X_{\mathrm{corr}} = \Delta^{*} + X_{\mathrm{corr}} > 0$$

where $X^{*}_{norm} \sim \mathcal{N}\left(\Delta, \dfrac{\sigma^{2}\left(\Delta^{*}\right)}{K^{2}}\right)$.
If $\sigma^2 \approx K^2$ and use Langevin Sampler it works fine.