Stochastic Gradient Descent Hamiltonian Monte Carlo Applied to Bayesian Logistic Regression

Sta663 Final Project

Gilad Amitai and Beau Coker

Abstract

Hamiltonian Monte Carlo (HMC) is a Markov chain Monte Carlo algorithm for drawing samples from a probability distribution where proposed values are computed using Hamiltonian dynamics to find values of high acceptance probabilities. They allow us to explore sample states more efficiently than random walk proposals, but are limited by the expensive computation of the gradient of the potential energy function. Chen, Fox, and Guestrin propose the method Stochastic Gradient Hamiltonian Monte Carlo (SGHMC), a HMC algorithm that uses a subset of the data to compute the gradient. The authors find that the stochastic gradient is noisy and correct this with a friction term.

In this project, we adapt the SGHMC to be used for Bayesian Logistic regression, implement this method in Python, optimize the code for computational efficiency, validate our approach using simulated data, and apply the algorithm to real world classification problems.

1 Background

Because this is a project about Hamiltonian Monte Carlo, imagine a frictionless puck on an icy surface of varying heights. The state of this puck is given by it's momentum $\boldsymbol{q}$ and position $\boldsymbol{p}$. The potential energy of the puck U will be a function of only its height, while the kinetic energy will be $K(q) = \frac{|\boldsymbol{p}|^2}{2m}$. If the ice is flat, the puck will move with a constant velocity. If the ice slopes upwards, the kinetic energy will decrease as the potential energy increases until it reaches zero, at which point it will slide back down.

2 Description of Algorithm

Stochastic Gradient Hamiltonian Monte Carlo proposes using a subset $\tilde{\mathcal{D}}$ of the entire dataset $\mathcal{D}$ to compute the stochastic gradient

$$\Delta\tilde{U}(\theta) = -\frac{|\mathcal{D}|}{|\tilde{\mathcal{D}}|}\sum_{x\in\tilde{\mathcal{D}}}\Delta\log p(x|\theta) - \Delta\log p(\theta)$$

which can then be used in the Hamiltonian Monte Carlo equations in the stead of the gradient $\Delta U(\theta)$. Logistic regression assigns the probability of success to a dichotomous response variable

$$\Pr\left(y_i = 1|\boldsymbol{x}_i, \boldsymbol{\beta}\right) = \frac{\exp\left\{\boldsymbol{x}_i^T \boldsymbol{\beta}\right\}}{1 + \exp\left\{\boldsymbol{x}_i^T \boldsymbol{\beta}\right\}}$$

where $\boldsymbol{x}_i$ is a vector of length p covariates for data point i and $\boldsymbol{\beta}$ is a vector of regression coefficients of length p. In a Bayesian framework, we would assign the a prior distribution on our unknown parameters $P(\boldsymbol{\beta}) \sim \mathcal{N}(0, \sigma^2)$ where, for the purposes of our project, σ^2 is known. The corresponding posterior will be proportional to $P(\theta) \prod_{i=1}^{n} \Pr\left(y_i|\boldsymbol{x}_i, \boldsymbol{\beta}\right)$, which would give us the potential energy function

$$U(\boldsymbol{\beta}) = -\log\left[P(\boldsymbol{\beta})\right] - \sum_{i=1}^{n} \log\left[\Pr\left(y_i|\boldsymbol{x}_i, \boldsymbol{\beta}\right)\right] = \sum_{j=1}^{p} \frac{\beta_j^2}{2\sigma^2} - \sum_{i=1}^{n} \left[y_i(\boldsymbol{x}_i^T \boldsymbol{\beta} - \log(1 + \exp\left\{\boldsymbol{x}_i^T \boldsymbol{\beta}\right\})\right]$$

and gradient components

$$\frac{\partial U}{\partial \beta_j} = \frac{\beta_j}{\sigma^2} - \sum_{i=1}^{n} x_{ij} \left[y_i - \frac{\exp\left\{\boldsymbol{x}_i^T \boldsymbol{\beta}\right\}}{1 + \exp\left\{\boldsymbol{x}_i^T \boldsymbol{\beta}\right\}}\right].$$

Input: Starting position $\theta^{(1)}$ and step size ϵ.
for *t=1, 2, . . .* **do**
  Sample momentum $r^{(t)} \sim \mathcal{N}(0, M)$;
  Set alsk;akfsl;kafl
end

3 Optimization

blah blah blah

4 Application to Simulated Data

blah blah blah

5 Application to Real Data

blah blah blah

6 Comparative Analysis

blah blah blah

7 Discussion and Conclusion

blah blah blah

8 Bibliography

References

[1]

[2]

[3]