Project Proposal

Samuel Leung

Disclaimer

The proposal written here is my best attempt at compiling and understanding all the information and concepts from our talks in office hours, as well as clarifying thoughts provided by Miguel. I am not confident that I have a good grasp of the project specifics, but hopefully the overall aim is there. Please let me know what areas of the project I've misunderstood, and apologies in advance if there are things that don't make sense.

Background

Bayesian inference of a parameter of interest involves deriving a posterior distribution based on prior belief and observed data (likelihood). In many cases, direct computation of the posterior is not possible. Fortunately, Monte-Carlo Markov Chain (MCMC) algorithms make it possible to sample from the posterior, recreating the posterior distribution fully as the number of samples approach infinity. A majority of MCMC methods are based on the Metropolis-Hastings algorithm, but for this project we explore a possible adaptation of another MCMC sampling method: simulated tempering. We restrict our exploration to be on problems with discrete state space, and plan to benchmark the proposed twist against the Metropolis-Hastings algorithm.

The Twist

Consider a problem with discretized state space $X_i; i \in \{1, \cdots, n\}$, where $X_i \sim \pi_i(X_i)$. With traditional simulated tempering, given that we are at state X_i, we consider sampling from only the next neighboring state distributions π_{i-1} or π_{i+1} (neighboring in integer space). Rather than only considering these distributions, we can consider choosing to sample from non-neighboring π_i's (ie. consider jumping to different index i's).

As the goal is to generate the posterior distribution as sample size approaches infinity, we want to make large jumps that also do not make the acceptance probability too small. In addition, as where we want to jump to is dependent on where we are at the current iteration, we can model this as a Markov Chain and as a result has a stationary distribution satisfying the following, with i, j representing the current and next index respectively:

$$\pi(i) = \sum_{j \in \{1, \cdots, n\}} \pi(j) K(i|j)$$

Choosing which index to jump to next can be done deterministically using an objective function. A possible objective to maximize is an adaptation of the invariance property scaled by the distance between the proposed index and our current one:

$$f_{\text{obj}} = \sum_{i,j \in \{1, \cdots, n\}} \pi(i) K(j|i) \cdot d(i, j)$$

From above, $d(i,j)$ represents a measure of distance between indexes i, j. Specifically, we want to find the following:

$$\max_K \mathbb{E}[d(i,j)] \quad \text{such that } \pi K = \pi$$

From above: $i \sim \pi,\ j|i \sim K$, and K is π-invariant. From this lens, the problem of selecting the next index to jump to is cast as an optimal transport problem, which can be solved using linear optimization.

We also note that this maximization will be dependent on the distance measure chosen. At this stage, the plan is to use the absolute difference $|i - j|$ and squared distance $(i - j)^2$, but other distance measures can be considered as the project progresses.

Benchmarking

We consider using the rocket dataset from excercise 2 of the course, as well as a car crash dataset from kaggle. For both datasets, the plan is to estimate the posterior using our proposed modification as well as Metropolis-Hastings, and compare the distributions for each, as well as run-time.

Github link: https://github.com/sleung124/Stat447-Final-Project/tree/main