Markov Chains: DBEs

Caden Hewlett

2024-04-02

Global Balance

Checking π-Invariance, by Global Balance Equation. This gives us MC LOLN when combined with irreducibility.

$$\pi(x') = \sum_{x \in X} \pi(x) K(x' \mid x)$$

Detailed/Local Balance:

Write mathematically "let city 1 be x and city 2 be x', where each pair of cities is assigned a fixed number of planes going back and forth that remains constant over time."

$$\text{Detailed Balance}: \ \pi(x) K(x' \mid x) = \pi(x') K(x \mid x')$$

Basically $\mathbb{P}(x' \to x) = \mathbb{P}(x' \leftarrow x)$.

What is the Relationship

Between Local and Global Balance…

$$\text{Local Balance} \implies \text{Global Balance}$$

However, a system can be globally balanced but not locally balanced.

$$\text{Local Balance} \overset{\text{NOT}}{\Longleftarrow} \text{Global Balance}$$

If you have global balance but not local balance, this is a "non-reversible" setting.

Invariance of Metropolis-Hastings

Specifically, of symmetric proposals. Recall that invariance is a synonym for "satisfying the global balance equation." Further, since detailed balance implies global balance, we can show local balance.

Recall:

$$\alpha(x, x') = \min\big(1, r(x, x')\big), \text{ where } r(x, x') = \gamma(x')/\gamma(x) \equiv \pi(x')/\pi(x)$$

$$\pi(x) K(x' \mid x) = \pi(x)\Big(q(x' \mid x)\alpha(x, x')\Big)$$

$$\pi(x) K(x' \mid x) = \pi(x) q(x' \mid x) \min\big(1, r(x, x')\big)$$

$$\pi(x) K(x' \mid x) = \pi(x) q(x' \mid x) \min\big(1, \pi(x')/\pi(x)\big)$$

$$\pi(x) K(x' \mid x) = \pi(x) q(x' \mid x) \min\big(\pi(x), \pi(x')\big)$$

Assuming a **symmetric proposal**, this is…

$$\pi(x) K(x' \mid x) = \pi(x) q(x' \mid x) \min\big(\pi(x), \pi(x)\big) = \pi(x) q(x \mid x') \min\big(\pi(x'), \pi(x)\big) = \pi(x') K(x \mid x')$$

Which is the guarantee of convergence of M-H, since it is π-invariant by the above.

Further, if we have an **asymmetric** proposal, we now know how to modify MH!

Kernel Mixtures

To prove the invariance of MCMC algorithms, a common strategy is to "break down" the algorithm into simpler parts and to prove invariance of each part.

NOTE: We have used mixtures in the context of model building. Here, we use the same cosntruction but in a different context, i.e. constructing and analyzing MCMC allgorithms instead of model building.

Let us recall the beta-binomial model that we've seen a lot of times now.

```
# prior: Beta(alpha, beta)
alpha = 1
beta = 2

# observations: binomial draws
n_successes = 3
n_trials = 3

gamma_beta_binomial = function(p) {
  if (p < 0 || p > 1) return(0.0)
  dbeta(p, alpha, beta) * dbinom(x = n_successes, size = n_trials, prob = p)
}
```

Intuition

We will write an algorithm for the above but varying standard deviation (as an example for varying algorithms.) Instead of choosing one, we run *both* and have the kernel mixture decided by Bernoulli trials.

So.... we let K_1 be the MH kernel with proposal standard deviation 1, and K_2 the MH kernel with proposal standard deviation 2.

Then, the changing of `proposal_sd` decides which variable we change.

```
kernel = function(gamma, current_point, proposal_sd) {
  dim = length(current_point)
  proposal = rnorm(dim, mean = current_point, sd = proposal_sd)
  ratio = gamma(proposal) / gamma(current_point)
  if (runif(1) < ratio) {
    return(proposal)
  } else {
    return(current_point)
  }
}
```

Putting this together!

```
# simple Metropolis-Hastings algorithm (normal proposal)
mcmc_mixture = function(gamma, initial_point, n_iters) {
  samples = numeric(n_iters)
  dim = length(initial_point)
  current_point = initial_point
  for (i in 1:n_iters) {

    kernel_index_choice = if (runif(1) < 0.5) 1 else 2
```

```
    current_point = kernel(gamma, current_point, kernel_index_choice)

    samples[i] = current_point
  }
  return(samples)
}

#ource("../blocks/plot_traces_and_hist.R")
samples = mcmc_mixture(gamma_beta_binomial, 0.5,  1000)
```

Why does this work?

Instead of having equal probability of K_1 and K_2. Instead, let ρ_1 be $\mathbb{P}(\text{use } K_1)$ and hence $\rho_2 = 1 - K_2$.

$$\underset{\text{mix}}{K}(x' \mid x) = \sum_{i=1}^{2} \rho_i K_i(x' \mid x)$$

Proposition: if K_i are π-invariant, then their mixture is π-invariant.

Question: can you prove this result? Convince yourself at the same time that the argument crucially depends on not varying with the current state.

$$\sum_{x \in X} \pi(x) K(x' \mid x) = \sum_{x \in X} \sum_{i \in N} \pi(x) \rho_i K_i(x' \mid x)$$
$$\sum_{x \in X} \pi(x) K(x' \mid x) = \sum_{i} \rho_i \sum_{x \in X} \pi(x) K_i(x' \mid x)$$
$$\sum_{x \in X} \pi(x) K(x' \mid x) = \sum_{i} \rho_i \pi(x') \quad \text{by invariance of each } K_i$$
$$\sum_{x \in X} \pi(x) K(x' \mid x) = \pi(x') \sum_{i} \rho_i$$
$$\sum_{x \in X} \pi(x) K(x' \mid x) = \pi(x')$$