Marcov Chain Monte Carlo (MCMC)

Giorgio Corani - SUPSI

Bayesian Data analysis and Probabilistic Programming

Credits

- J. Krushke, “Doing Bayesian Data Analysis”, Chapter 7

Bayes' rule

- The computationally challenging part is the denominator:

$$f(\theta \mid D) = \frac{\underbrace{f(\theta)}_{\text{prior}} \underbrace{\mathcal{L}(D \mid \theta)}_{\text{likelihood}}}{\underbrace{f(D)}_{\text{marg lik}}}$$

- where D denotes the observed data and

$$f(D) = \int f(D, \theta) d\theta = \int f(\theta)\mathcal{L}(D \mid \theta) d\theta$$

Bayes' rule

- In the conjugate case, we have an analytical expression of $f(\theta \mid D)$ but generally the likelihood is **not** conjugate to the prior.

- Is gridding a solution?
  - define a grid of values of the parameters
  - for each point of the grid, compute the product (prior x likelihood)
  - normalize the obtained density

Gridding does not scale

- The joint distribution of 6 parameters, each represented by 1000 states, has 1000^6 states, too much for any computer.

MCMC

- MCMC methods address this type of problems.
- It approximates the posterior of θ by returning many samples.
- Doing so, it avoids computing the difficult integral in the denominator of Bayes' rule.
- For simplicity, we show a single-parameter problem.

Approximating a distribution with a (large) sample

- By randomly sampling a subset of people from a population, we can estimate the underlying tendencies in the entire population.
- The larger the sample, the better the estimation.
- The population from which we want to sample is the **posterior distribution of θ**.

Approximating a distribution by a large random sample

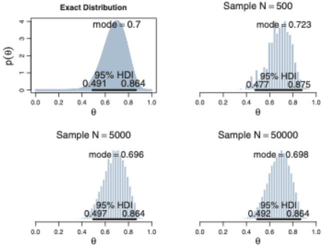

Figure 7.1 Large representative samples approximate the continuous distribution in the upper-left panel. The larger the sample, the more accurate the approximation. (This happens to be a beta(θ|15, 7) distribution.)

- Larger samples yield a more accurate histogram;
- The exact values (top-right) were obtained from the mathematical formula of the beta distribution.

The Metropolis algorithm: a simple example.

- We live in a chain of 7 islands.
- We have to do many travels, visiting each island proportionally to its population.
- We can move from an island to a neighboring one, or remain on the same island.
- The population of the different islands is 1000, 2000, 3000, ...,7000.
- This is the unnormalized distribution we want to sample from.

Making decisions

- Flip a coin to decide whether the *proposed* island is east or west of the current one.

- If it has a larger population than the current one, visit it.

- Otherwise, visit it with probability

$$p_{\text{move}} = \frac{\text{pop}_{\text{proposed}}}{\text{pop}_{\text{current}}}$$

- $\text{pop}_{\text{proposed}}$: population of the proposed island
- $\text{pop}_{\text{current}}$: population of the current island.

- In the long run, each island is visited proportionally to its population.

Discussion

- The proposed direction and its acceptance are random.
- If the process were started over again, the specific trajectory would be different.
- Yet, in the long run the relative frequency of visits mimics in every case the target distribution.

Proposal

- The possible moves and the probability of proposing each constitute the *proposal distribution*.
- Our proposal distribution has only two values (left and right) with probability 0.5 each.

Accepting the proposal

- If the target distribution is greater at the proposed position, we accept the proposal: *we always move higher if we can*.
- If the target distribution is less at the proposed position than at our current position, we accept the move with probability:

$$p_{\text{move}} = \frac{\text{pop}_{\text{proposed}}}{\text{pop}_{\text{current}}}$$

- More in general, we move to the proposed position with probability:

$$p_{\text{move}} = min\left(\frac{f(\theta_{\text{proposed}})}{f(\theta_{\text{current}})}, 1\right)$$

where $f(\theta)$ is the unnormalized posterior (see later).

We use the *unnormalized* posterior

- The algorithm evaluates the ratio $\frac{f(\theta_{\text{proposed}})}{f(\theta_{\text{current}})}$, without computing the normalizing constant (i.e., the denominator) of Bayes rule.
- It does not require the absolute value of $f(\theta|D)$, but only the ratio between the density in different locations.
- We sample from $f(\theta)f(D|\theta)$ without normalizing by the (often) untractable marginal likelihood $p(D)$.
- In the example of islands-hopping, the target distribution was the unnormalized population of each island, not a normalized probability.

Random walk

- The samples from the posterior are generated by a *random walk*
- The walk starts from a randomly chosen point where the distribution is non zero.
- At each time step we propose the move to a new position θ_{proposed}.
- We accept the move with probability

$$p_{\text{move}} = min\left(\frac{f(\theta_{\text{proposed}})}{f(\theta_{\text{current}})}, 1\right)$$

Metropolis algorithm applied to the beta-binomial model

$$f(\theta \mid D) \propto f(D \mid \theta) f(\theta) = \theta^{a+y-1}(1-\theta)^{b+n-y-1}$$

- θ is a continuous parameter
- Proposal distribution: $\Delta\theta \sim N(0, \sigma)$.
- θ_{proposed} lies in an interval of $\pm 3\sigma$ around θ_{current}.
- Hence σ controls how far θ_{proposed} can be from θ_{current}.

Sampling the beta-binomial

- Draw the starting point $\theta_0 \in (0,1)$.

At each iteration:

- Draw $\Delta\theta \sim N(0,\sigma)$
- $\theta_{\text{proposed}} = \theta_{\text{current}} + \Delta\theta$

- Make sure the θ_{proposed} is within the feasible interval; the density is only defined in the open interval (0,1).

$$\theta_{\text{proposed}} = \text{min}(\theta_{\text{proposed}}, 0.9999)$$
$$\theta_{\text{proposed}} = \text{max}(\theta_{\text{proposed}}, 0.0001)$$

Probability of the move

We move to θ_{proposed} with probability:

$$\begin{aligned} p &= \min\left(1, \frac{f(\theta_{\text{proposed}} \mid D)}{f(\theta_{\text{current}} \mid D)}\right) \\ &= \min\left(1, \frac{\theta_{\text{proposed}}^{a+y-1}(1-\theta_{\text{proposed}})^{b+n-y-1}}{\theta_{\text{current}}^{a+y-1}(1-\theta_{\text{current}})^{b+n-y-1}}\right) \end{aligned}$$

Application

- Consider the prior $p(\theta) = Beta(1, 1)$
- The Bernoulli likelihood $\theta^{14}(1 - \theta^{6})$ corresponds to 20 tosses and 14 tails.
- The three columns use three different σ in the proposal distribution.

Results

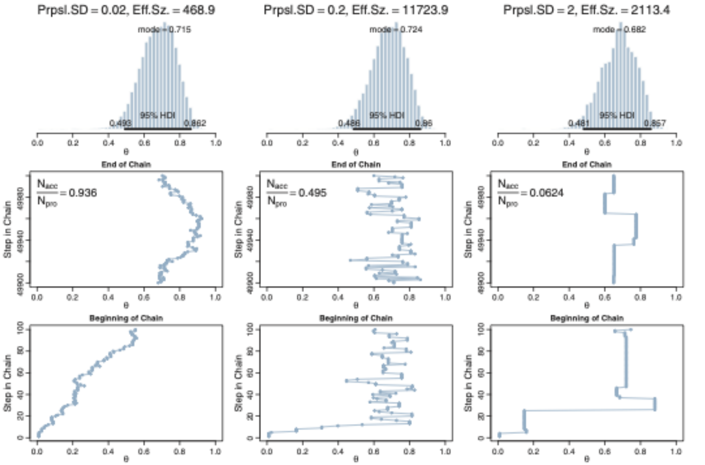

Figure 7.4 Metropolis algorithm applied to Bernoulli likelihood with beta($\theta|1,1$) prior and $z = 14$ with $N = 20$. For each of the three columns, there are 50,000 steps in the chain, but for the left column, the proposal standard deviation (SD) is 0.02, for the middle column SD = 0.2, and for the right column SD = 2.0.

Discussion

- $\frac{N_{\text{acc}}}{N_{\text{pro}}}$ denotes the probability of accepting a proposal.
- It decreases when σ gets larger.

Discussion

- The sequence of sampled values is called *trace*. The value of the trace are generally auto-correlated (i.e., the next sampled value is similar to the previous sampled value).

- Autocorrelation is stronger in the first column, medium in the second column, and strong again in third column.

- The *effective samples size* is is the equivalent number of samples if they were sampled independently of each other (as the sampling is a sequential process, some auto-correlation in the samples is generally present).

Left column (small σ)

- The left column uses a the small σ = 0.02.
- The successive steps in the chain make small moves
- This requires a very long chain to explore the posterior distribution.
- The effective size of this 50,000 step chain is only 468.9.

Right column (large σ)

- The proposed jumps are often far away from the bulk of the posterior distribution; the proposals are often rejected.
- The process rarely accepts new values, producing a clumpy chain.
- The effective size of this 50,000 step chain is only 2113.4.

Discussion

- In no case we have 50,000 *independent* samples of the posterior.
- The effective sample sizes are 468, 11723, 2113 in the three different scenarios.
- Tuning the width of the proposal distribution has a major impact on the effectivness of the sampling.
- In general good sampling is obtained when about 50% of the proposals is accepted (central column).
- Advanced implementations of the Metropolis algorithm automatically adjust the width of the proposal distribution.

Your turn

You can choose between two implementation tasks:

- Implement the island hopping algorithm and check that your visits are proportional to the population of the islands
- Implement the sampling of the posterior Beta and check that the mean of your samples is close to the actual posterior mean.