MO435 — Probabilistic Machine Learning
INSTITUTO DE COMPUTAÇÃO — UNICAMP

Markov Chain Monte Carlo

Lesson No. 12

Thiago Gomes Marçal Pereira *

1 Introduction

When doing Bayesian Statistics, we calculate the posterior probability of some set of data as

$$P(w_i|D) = \frac{P(D|w_i)P(w_i)}{\sum_{j=1}^{N} P(D|w_j)P(w_j)} \tag{1}$$

and for continuous parameters

$$P(\tau_i, t_i, \alpha|data) = \frac{P(data|\tau_i, t_i\alpha)P(\tau_i, t_i, \alpha)}{\sum_{j=1}^{C} \int_{t_j} \int_{\alpha'} P(data|\tau_i, t_j, \alpha')P(\tau_j, t_j, \alpha')dt_j d\alpha'} \tag{2}$$

And this can be extremely difficult and sometimes even impossible to compute analytically, or even numerically using many different techniques.
The solution for this problem is to use Markov Chain Monte Carlo (MCMC).

2 Pre-definitions

Considering the parameters in our model

- For each parameter value, we can compute its likelihood
- For each parameter value, we already know it's prior probability
- We can easily compute prior x likelihood for any given point in parameter space

3 MCMC

Unlike Monte Carlo sampling, that is capable of getting random independent samples from the distribution, a Markov Chain draws samples that are dependent on existing samples.
By combining both approaches, the Markov Chain Monte Carlo is capable of drawing random samples from the distribution probability, based on the probability of the last sample that was drawn.

There are many different algorithms to reach the MCMC approach, and we'll talk about some of them next.

3.1 Metropolis-Hastings

The Metropolis-Hastings algorithm is appropriate for cases where the next probability distribution cannot be directly calculated. So, instead, it uses a proposal probability distribution that is sampled and based on an acceptance criteria, the sample is either added to the chain, or discarded.
This criterion is probabilistic and based on how likely the proposed distribution is to the next state probability distribution.
The algorithm will usually follow:

*RA: 189691 - t189691@g.unicamp.br

- Define a random start position and compute $P_x = priorxlikelihood$
- Randomly draw a sample from the proposal distribution for a new position Y, so that we have $q(Y|X)$
- Compute $P_y = priorxlikelihood$ for this new position
  - if $P_y > P_x$ acceps
  - otherwise we can calculate the probability of acceptance as

$$P(accept) = \frac{P_Y xq(X|Y)}{P_X xq(Y|X)} \tag{3}$$

- if the value is accepted, store the sample value, if not, store the previous value again

After repeating these steps many times, we tend to get an empirical approximation of the distribution.
If there are many peaks at the distribution, we can easily get stuck in one of them. A solution to that would be the Metropolis-coupled Markov Chain Monte Carlo, that would allow more peaks to be visited. The MC^3 won't be treated here.

3.2 Gibbs Sampling

This algorithm approach is to construct a Markov chain where the probability of the next sample is calculated as the conditional probability given the prior sample. And this is done by changing only one variable at a time, allowing to search for samples in the near space. But this risks that the chain can get stuck.
Gibbs sampling is more appropriate to discrete distributions, because the conditional probability can be easily calculated.

Considering a distibution $p(z) = p(z_1, ..., z_M)$ at each step θ of the algorithm, we'll replace the value $z_i^{(\theta)}$ by a value $z_i^{(\theta+1)}$ obtained by sampling from the conditional distribution, cycling for each variable i from our distribution in turn.

3.3 Slice Sampling

As seen, the step size can be a huge problem for the Metropolis algorithms. So, the Slice Sampling intends to provide an adaptive step size, adjusted to the characteristics of the distribution.
The goal is to sample uniformly from the distribution

$$\hat{p}(z,u) = \begin{cases} 1/Z_p & if 0 \leq u \leq \widetilde{p}(z) \\ 0 & otherwise \end{cases} \tag{4}$$

In practice, it can be difficult to sample from a slice through the distribution, so, instead we try to sample the next value from a region $z_min \leq z \leq z_max$ that contains our current value $z^{(\tau)}$. One approach is to try a region of width w and test its boundaries to see if they lie within the slice. If either of the is not, the region is extended. If they lie, the region can be shrunk.
Then another candidate point is drawn uniformly from this new region.

Slice sampling can be used by repeatedly sampling each variable in turn, just like Gibbs Sampling.

4 Conclusion

The Markov Chain Monte Carlo algorithms provide a good way of sampling in situations where the simple Monte Carlos would not fit. Specially in problems with high dimensions.
They also provide a wide range of algorithms to specific situations like discrete or continuous distributions. And even though they also have some limitations, it's worth considering them and their variations, that try to minimize the effects of those limitations.