STCS 6702: Foundations of Graphical Models: Reading 3

Maxime TCHIBOZO (MT3390)

September 2020

1 Neal (1993) – Probabilistic Inference Using Markov Chain Monte Carlo Methods – Chap 1-4

This paper describes the foundations of Markov Chain Monte Carlo methods, and illustrates their use in Statistical Physics as well as Artificial Intelligence.

In "Artificial Intelligence" challenges, MCMC are used in conjunction with belief networks which specify relations between variables, with neural networks which approximate functions, with latent variables to infer unobservable parameters, and with hierarchical models, in which the prior set of latent parameters is itself parametrized by hyperparameter.

The main challenges in probabilistic Machine Learning are the complexity of distributions at play, their high dimensionality which leads to the inability to solve equations analytically in many cases. Sampling points from these distributions to approximate statistics can solve these challenges, and Monte Carlo methods allow us to estimate the distributions by generating samples when the above challenges arise.

Markov Chain based Monte Carlo sampling generates individual samples of n observations which converge to the true distribution of the n observations. Statistics such as expectations are then measured on the sampled data to approximate statistics of the true data. Usually, MCMC algorithms utilize "locality", or transitions between states where one variable (or state) is kept fixed.

Interestingly, Neal weighs into the discussion over "frequentist" and "Bayesian" interpretations of statistics in this paper:

In frequentist statistics, probabilities exist as the limits of frequencies of events in experiments which can be repeatable. In this context, probabilities can not

be used to interpret unique events which are never reproduced. Neal gives the following example: "We cannot for example ask what is the probability that Alexander the Great played the flute". On the other hand, Bayesian statistics introduce "degrees of belief" and use probability to quantify how much the evidence/observations fits/fit with our prior belief. In "Artificial Intelligence", the Bayesian framework is widely used. Bayesian predictions are preferable to frequentist ones in cases where there is not enough data for frequentist approaches to identify clear parameters (example, if we only flip a coin 3 times and it yields heads on each toss). Bayesian predictions also perform well when the posterior predictive distribution is strongly peaked (i.e modal).

Using Bayes' rule, the integration of marginal distribution is necessary when computing the distribution of observations, which is the denominator of the posterior(probability of the observations is conditioned over latent variable). Integrating exact marginal distributions over high dimensional spaces is infeasible, but Monte Carlo methods allow us to approximate them. This relies on reliable estimates of the conditional probabilities.

In cases where the number of variables over which to marginalize is small, or if the factorization is convenient, a "brute force" computation of the integral using numerical methods is possible, and there is no need for MC methods. However, these are not appropriate for high dimensional spaces.

Neal details three cases in which Monte Carlo integration would yield different results: if the probability density function f associated to variable a has no modes (is equally distributed across the domain), MC will estimate the mean of a well. If f is strongly modal and peaks for a specific value of the domain (and is close to 0 elsewhere), MC will not be effective. Likewise, if f has a strong peak for a certain value of the domain (and close to 0 elsewhere) at which a is close to 0, MC will not work well.

MCMC relies on the implicit assumption the variable or statistic of interest a(x) does not vary strongly from one likely region (high probability region) to another. In some cases, generating a number of independent realization of the Markov Chain alleviates this problem.

Rejection sampling is also an interesting option for sampling according to a distribution, however this is not feasible for complicated problems such as those seen in Machine Learning. Neal does, however, mention "adaptive rejection sampling", which can be incorporated into MCMC methods. Importance sampling algorithms are another class of Monte Carlo estimation methods, but they do not incorporate any exploration of high probability regions, and thus do not perform well in our context.

The sampling method to be used must take into account the location of high-probability states. One solution can consist of parametrizing a local maxima – which we hope will be a mode – with a gaussian of unknown mean and variance

(or using this gaussian as the importance sampling distribution). MCMC are combine sampling (exploitation) with a search for large high probability regions (exploration).

Neal explains the foundations of MCMC: the state of the Markov chain approximating the distribution of interest at time t_i is nearly independent of the state at time t_j if i »j. At large time intervals, the Markov Chain distribution is close to its invariant distribution. While Markov Chain convergence properties are convenient in finite countable state spaces, in continuous or infinite state spaces many of these properties no longer hold. Such is the case for unconstrained random walks. Random walk sampling procedures therefore perform poorly, but their behavior can be improved by constraining them to a finite number of steps.

To sample from the joint distribution, the Gibbs sampler uses a transition matrix for each component of the joint, which leaves all other components fixed and conditions on them. The base transitions are then applied in sequence. Gibbs sampling is appropriate for discrete variable spaces which take values from a small set, and for continuous variables whose conditional distributions can be sampled from easily. Gibbs sampling has proved effective on Bayesian belief networks.

Gibbs sampling is not appropriate for Bayesian Multi-Layer Perceptrons, as they generate complex and multimodal conditional distributions. Metropolis algorithm is preferable in this case.

The Metropolis algorithm is a generalization of Gibbs sampling, and avoids any need to sample from difficult distributions. The Metropolis algorithm has no constraints on whether the state space is continuous or discrete, and only requires the possibility of computing ratios of densities of two states. For each sample of the joint X containing n variables, we generate random changes to the components of X using a symmetric proposal distribution, and accept or reject them depending on how they affect the probability of each changed component state. For continuous components, a parametric distribution centred on the current value can be used. More generally, the primary condition of Metropolis algorithm is that the probability distributions involved are never 0.

Components can be updated in turn (local Metropolis algorithm) or updated simultaneously using a distribution over all components (global Metropolis algorithm).

When conditional distributions can be sampled from easily, Gibbs sampling is preferable. Gibbs sampling is also effective when some components of the state are almost independent. When decomposing conditional probabilities on a fixed state is complicated or when correlations are high, we prefer Metropolis.

Local Metropolis algorithm scales linearly with the number of components,

while the global Metropolis algorithm scales exponentially if the proposal distribution is fixed over iterations (this can be alleviated with an adaptive proposal distributions – but always remains more than linear). Variations of the Metropolis algorithm include trying different acceptance functions, using different distributions to generate the changes, modifying the rejection system or removing it altogether.