METROPOLIS HASTING AND SIMULATED ANNEALING

PUN WAI TONG

Contents

0. Goal 1
1. Metropolis-Hasting Algorithm 1
2. Simulated Annealing 2
References 3

0. Goal

This note is a draft of a brief introduction of Metropolis-Hasting Algorithm and stimulated annealing. Reader should read this note with caution. There are many references about these two topics, e.g. [1] and [2]

1. Metropolis-Hasting Algorithm

It is well known that there is a curse of dimensionality when we sample states following a probability distributions π in a high dimensional sample space of states S. Metropolis-Hasting is one of popular approaches to solve this issue. The main idea to make Metropolis-Hasting Algorithm work is to find a transition kernel density $p(x, y)$ on $S \times S$ such that the the Markov chain induced by the transition kernel density $p(x, y)$ has the limiting probability $\pi_L = \pi$. If we can find such a transition kernel density, then we can start the Markov chain with a random state s_0 and let the Markov chain proceed. Then, we can start to collect states induced by the Markov chain after sufficiently many iterations in the chain and the states we collected follow the limiting probability distribution $\pi_L = \pi$.

The key to implement Metropolis-Hasting Algorithm successfully is to find another transition kernel density $q(\cdot, \cdot)$ on $S \times S$ such that it follows the following properties:

(1) Given $x \in S$,it is easy to sample states from $q(x, \cdot)$.
(2) Let f be a density of a probability distribution π with respect to a probability measure μ on S, i.e. $\pi(d\mu) = f d\mu$. For all $x \in S$, the support of $q(x, \cdot)$ contains the support of $f(\cdot)$, i.e. if $f(y) > 0$, then $q(x, y) > 0$. [Actually, we can weaken this condition in practice.]

States $\{X_t\}$ is denoted to be a Markov chain in the Metropolis-Hasting Algorithm. The following is Metropolis-Hasting Algorithm to samples states in S following a probability distribution π:

Date: 00:01, Wednesday 13th September, 2017 *File:"Metropolis Hasting and Simulated Annealing_ver2".tex*.

(1) For $t = 0$, pick an initial state $X_0 = s_0 \in S$ such that $f(s_0) > 0$ and assign the last step T to be a large positive number.
(2) Generate a candidate Y follows from the probability distribution $q(X_t, \cdot)$
(3) Generate a random number β following a uniform distribution between 0 and 1
(4) Update the state X_{t+1}as $\begin{cases} Y & \text{if } \alpha(X_t, Y) > \beta \\ X_t & \text{otherwise} \end{cases}$ where

$$\alpha(x, y) := min\left\{\frac{f(y)q\,(y, x)}{f(x)q\,(x, y)}, 1\right\}. \tag{1.1}$$

(5) $t \leftarrow t + 1$ and go to Step 2 until $t > T$.

Metropolis-Hasting Algorithm builds a Markov chain $\{X_t\}$. Then we pick $T_{eqm} < T$ large enough such that all states generated after T_{eqm} follow very closely to a probability distribution π and we collect states $\{X_t\}_{T_{eqm}<t<T}$ as a sample on S following probability distribution π.

Remark 1.1. There are three remarks when we implement Metropolis-Hasting Algorithm in practice.

(1) Both probability distributions $q(x, \cdot)d\mu(\cdot)$ and $\pi(\cdot)$ are not necessarily to be normalized, that we do not require $\int_{y\in S} q(x, y)d\mu(y)$ and $\pi(S) = \int_{y\in S} f(y)dy$ equals to 1. Especially the normalizing constant to normalize $q(x, \cdot)$ or $\pi(\cdot)$ are unknown in some cases.
(2) The transition kernel density $q(x, y)$ is not necessarily to be symmetric, i.e. $q(x, y) = q(y, x)$.
(3) If the initial state X_0 we start at has $f(X_0) = 0$, then the update in Step 4 becomes invalid as the denominator of Eq. 1.1 becomes zero.

INFORMALLY, the transition kernel density in the Markov chain of the Metropolis-Hasting Algorithm can be written as

$$p(x, y) = q(x, y)\alpha(x, y) + r(x)\delta_x(y) \tag{1.2}$$

where $\delta_x(y)dy$ is a Dirac measure, i.e.

$$\int_{y\in A} f(y)\delta_x(y)dy = \begin{cases} f(x) & \text{if } x \in A \\ 0 & \text{otherwise} \end{cases}$$

and

$$r(x) = 1 - \int_{y\in S} q(x, y)\alpha(x, y)d\mu(y).$$

The first term on R.H.S. in Eq. 1.2 stands for the probability of all accepted move from a state x to a state y ($X_{t+1}$is updated as Y) while the second term on R.H.S. in Eq. 1.2 stands for the probability of all unaccepted move from a state x to a state y (X_{t+1} is updated as X_t).

2. Simulated Annealing

One application of Metropolis-Hasting Algorithm is to search for an optimal point(global minimum/ global maximum) point of a function. The function needs not to be continuous. Suppose S is a sample spaces of states (it can be a high dimensional space) and $E : S \to \mathbb{R}$ is an objective function. Let's say we search

for the global minimum point of E. The main idea of simulated annealing is to find a probability distribution π on S such that π is extremely large at the global minimum point and almost zero elsewhere. Then, we apply Metropolis-Hasting to generate a Markov chain whose states follows the probability distribution π at the very end. In other words, the last state in the Markov chain is very close to the global minimum point. One common example to construct such a probability distribution π is

$$f(x) = \exp\left(-E\left(x\right)\right). \tag{2.1}$$

where f is the density function corresponding to π, i.e. $\pi(d\mu) = fd\mu(x)$. However, in order to increase the ability to get rid of local minimum point and stabilize the convergence of states in the Markov chain, we also introduce another parameter, temperature $\mathscr{T}$ to Eq. 2.1 and the probability distribution becomes

$$\pi(x) = \exp(-\frac{E(x)}{\mathscr{T}})$$

where $\mathscr{T}$ decreases slowly along steps in the Markov chain.

Let $\{X_t\}$ be a Markov chain in simulated annealing algorithm for minimizing $E : S \to \mathbb{R}$. The simulated annealing algorithm is as follows:

(1) For $t = 0$, pick an initial state $X_0 = s_0 \in S$ and initialize both the last step T and the temperature $\mathscr{T} = \mathscr{T}$ to be a large positive number.
(2) Generate a candidate Y follows from the probability distribution $q(X_t, \cdot)$
(3) Generate a random number β following a uniform distribution between 0 and 1
(4) Update the state X_{t+1}as $\begin{cases} Y & \text{if } \alpha(X_t, Y) > \beta \\ X_t & \text{otherwise} \end{cases}$ where

$$\alpha(x, y) := min\left\{\exp(-\frac{E(y) - E(x)}{\mathscr{T}})\frac{q\left(y, x\right)}{q\left(x, y\right)}, 1\right\}.$$

(5) $t \leftarrow t + 1$ and $\mathscr{T} \leftarrow \mathscr{T}(0.5^{\frac{t}{1000}})$ and go to Step 2 until $t > T$.

The last state X_T should be very close to the global minimum point of the function E. There is a remark of the simulated annealing algorithm.

Remark 2.1. There are two remarks about the simulated annealing algorithm

(1) In the step 2 of the algorithm, without further information, we have no ideas which direction a state should move so it can get closer to the global minimum point. Therefore, all proposed directions in the step 2 should be equally likely. Hence, the transition kernel q should be symmetric, i.e.$q(x, y) = q\left(y, x\right)$. Common examples of such a $q(x, y)$ is a Gaussian distribution, random walk or a uniform distribution. Furthermore, if $q(\cdot, \cdot)$ is symmetric, then $\alpha(x, y) = min\left\{\exp(-\frac{E(y)-E(x)}{\mathscr{T}}), 1\right\}$.
(2) The update formular of temperature $\mathscr{T}$ in the step 5 means $\mathscr{T}$ decreases by half in every 1000 iteration.

References

[1] Charles J Geyer. Markov chain monte carlo lecture notes. *http://www.stat.umn.edu/geyer/f05/8931/n1998.pdf*, 2005. 0

[2] Stephen Sauchi Lee. Markov chains on continuous state space. http://www.webpages.uidaho.edu/ stevel/565/lectures/5d0