COS 513 LECTURE 17 SCRIBE NOTES

DAN LI, ANDREI UNGUREANU

1. Review of Metropolis algorithm

We are going to start by reviewing the Metropolis algorithm. We will assume that the proposal transition kernel is symmetric $q(x'|x) = q(x|x')$. Starting with an initial random state $x^{(0)}$ the procedure at time t is the following:

- Sample x^* from the proposal $q(x|x^{(t-1)})$.
- Accept x^* with probability $A(x^{(t-1)}, x^*) = \min(1, p(x^*)/p(x^{(t-1)}))$. If accepted, set $x^{(t)} = x^*$, otherwise set $x^{(t)} = x^{(t-1)}$.

Remark 1.1. As long as $q(x'|x) > 0 \ \forall x, x'$ then $p_m(x^{(t)}) \to p(x)$ as $t \to \infty$; here we have denoted by $p_m(x^{(t)})$ as the marginal probability of $x^{(t)}$ obtained by running the sampling algorithm.

2. Theoretical aspects of Markov chains

In general an MCMC algorithm defines a Markov chain whose unique stationary distribution is the distribution of interest. Notice that consecutive samples obtained in an MCMC algorithm are correlated so to get independent samples from the distribution of interest we need to collect samples at some lag. The main question we need to address is when such a Markov chain is going to converge to the stationary distribution, the distribution of interest for us. Let us now shortly present the underlying theory of Markov chains and some convergence results.

Definition 2.1. A *Markov chain* is defined by an initial probability distribution $p_0(x)$ and transition probability kernels for each time step m: $T_m(x_m, x_{m+1}) = p(x_{m+1}|x_m)$.

Definition 2.2. A Markov chain is *homogenous* if $T_m = T \ \forall m$.

For now we will work with a homogenous Markov chain with a discrete state space.

Remark 2.3. A recursion formula for the marginal probability of x_{m+1} is immediate: $p_{m+1}(x_{m+1}) = \sum_{x_m} p(x_{m+1}, x_m) = \sum_{x_m} p(x_{m+1}|x_m) p_m(x_m)$.

Definition 2.4. A distribution p^* is *invariant* (*stationary*) with respect to a Markov chain if each step leaves the distribution invariant: $p^*(x) = \sum_{x'} T(x', x)p^*(x')$.

Remark 2.5. In general a Markov chain can have 0 or more invariant distribution.

Proposition 2.6. *A sufficient (but not necessary) condition for a distribution to be invariant is detailed balance:* $p^*(x)T(x, x') = p^*(x')T(x', x)$.

Proof.

$$\sum_{x'} p^*(x')T(x', x) = \sum_{x'} p^*(x)T(x, x') = p^*(x) \sum_{x'} p(x'|x) = p^*(x)$$

□

Finally, another desirable property of Markov chains is ergodicity, in the sense that the Markov chain converges to some invariant distribution regardless of the initial distribution.

Definition 2.7. A Markov chain is said to be *ergodic* if for all $p_0(x)$, $p_m(x) \to p^*(x)$ as $m \to \infty$ (where p^* does not depend on p_0).

Theorem 2.8. (Neal 1993) *If a homogeneous Markov chain on a finite state space with transition probabilities T has p^* as a stationary distribution, and*

$$\nu = \min_x \left\{ \min_{x':\, p^*(x')>0} \frac{T(x, x')}{p^*(x')} \right\} > 0,$$

then the Markov chain is ergodic.

In fact the following bound on the rate of convergence holds (for all x):

$$|p^*(x) - p_n(x)| \leq (1 - \nu)^n.$$

3. MCMC ALGORITHMS

3.1. **Overview and considerations.** Our goal now is to construct homogeneous Markov chains with stationary distribution equal to our target distribution, while minimizing the computational effort required to sample from the stationary distribution. The computational effort comes from three main sources:

(1) Computation to simulate each transition.
(2) Time for the distribution to converge to p^* (this is called the "burn in").
(3) Number of samples needed to go from one draw of p^* to the next ("lag"). This is required in order to get independent samples from the stationary distribution.

Remark 3.1. Clearly 2 and 3 are related but lag tends to be less than burn in. These generally need to be determined empirically.

3.2. **Metropolis-Hastings algorithm.** Suppose our state space has K components $X = \{X_1, \ldots, X_K\}$. We consider K transition matrices B_1, ..., B_K such that each B_k affects only X_k (and holds X_j fixed for $j \neq k$), and we apply each transition in turn to go from $x^{(t)}$ to $x^{(t+1)}$.

Remark 3.2. A key fact is that if detailed balance holds for each of T_1, T_2, $\ldots T_K$, then it holds for their product $T_1 T_2 \ldots T_K$.

The algorithm works as follows. Suppose we are currently in state $x^{(t)}$. Transitioning to $x^{(t+1)}$ involves updating each component of $x^{(t)}$ in turn. The following gives the procedure to update the k^{th} component to go from a state x to x':

- Draw x^* from $B_k(x, x^*)$ (note that this can only change x_k).
- Accept x^* (i.e. set $x' = x^*$) with some probability $A_k(x, x^*)$; otherwise, reject x^* and set $x' = x$.

The acceptance probability A_k is given by

$$A_k(x, x') = \min\left\{1, \frac{p(x')B_k(x', x)}{p(x)B_k(x, x')}\right\}.$$

Remark 3.3. Note that since we are only interested in the ratio $p(x')/p(x)$, we only need to know p up to a normalization constant. We do need to be able to compute and sample from B_k, however.

Remark 3.4. When the B_k are symmetric, this is called the Metropolis algorithm.

We only need to verify detailed balance for the transition step:

$$\begin{aligned}
p(x)B_k(x, x')A_k(x, x') &= p(x)B_k(x, x') \min\left\{1, \frac{p(x')B_k(x', x)}{p(x)B_k(x, x')}\right\} \\
&= \min\left\{p(x)B_k(x, x'), p(x')B_k(x', x)\right\} \\
&= p(x')B_k(x', x) \min\left\{\frac{p(x)B_k(x, x')}{p(x')B_k(x', x)}, 1\right\} \\
&= p(x')B_k(x', x)A_k(x', x),
\end{aligned}$$

as desired.

Remark 3.5. When designing B_k, there is a trade-off between how big one's moves tend to be and how likely one is to accept each move, both of which can affect the speed of convergence.

3.3. **Gibbs sampling.** Again we are working with K components $X = \{X_1, \ldots, X_K\}$. At each iteration, if we are at state x, we draw the components for the next state like this:

- x_1 according to $X_1|x_2, \ldots, x_K$,
- x_2 according to $X_2|x_1, x_3, \ldots, x_K$,
- ...
- x_K according to $X_K|x_1, \ldots, x_{K-1}$.

Note that we only need to be able to compute the distribution of one component conditioned on all the other components (in a graphical model, this corresponds to one variable conditioned on its Markov blanket).