Sampling Methods

Reda Arab

Contents

1 Introduction

The goal of this document is to describe some sampling methods. In this document you will find : MCMC, Gibbs Sampling and the Metropolis Hasting algorithm.

For MCMC and Metropolis Hasting I used :

- The course Probabilistic Graphical Models (Part II) of Stanford University on Coursera (`https://www.coursera.org/learn/probabilistic-graphical-models-2-inference?`, Week 4)
- A document from MIT `https://ocw.mit.edu/courses/economics/14-384-time-series-analysis-fa lecture-notes/MIT14_384F13_lec25.pdf`

For Gibbs Sampling :

- The course Bayesian Statistics : Techniques and Models of University of California Santa Cruz on Coursera (`https://www.coursera.org/learn/mcmc-bayesian-statistics/lecture/35nWu/multiple-parameter-sampling-and-full-conditiona`
- A handout (`http://nitro.biosci.arizona.edu/courses/EEB596/handouts/Gibbs.pdf`

For more information about Markov Chain, refer to a course/book.

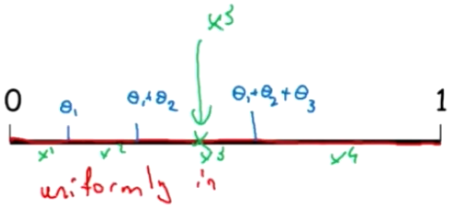
2 Simple sampling

First remark : Most computers has a random number generator function that generates samples uniformly in [0,1] i.e $\mathcal{U}(0,1)$. We have to take this into account when we are exploring sampling methods.

2.1 Sampling from discrete distribution

Suppose $X \sim P$ and takes values on $\text{Val}(X) = \{x^1, x^2, ...x^k\}$, with $\text{P}(x^i) = \theta^i$.

The trick is to divide the segment [0,1] into k segments $[0,\theta^1]$, $[\theta^1, \theta^1 + \theta^2]$, $[\theta^1 + \theta^2, \theta^1 + \theta^2 + \theta^3]$, ..., $[\theta^1 + \theta^2 + ... + \theta^{k-1}, \theta^1 + \theta^2 + ... + \theta^k = 1]$.



The lengths of the intervals correspond to the probability associated. Therebery, when we generate a random number a with $\mathcal{U}(0,1)$, if a is in $[0,\theta^1]$, we assign x^1, if a is in $[\theta^1, \theta^1 + \theta^2]$ we assign x^2 and so on.

2.2 Estimation

Use of Hoeffding and Chernoff Bound *(to complete).*

3 Markov Chain Monte Carlo

Use : Sample from a distribution P intractable to sample from.

[To complete : statistics for proving non mix, how to define a Markov Chain with limiting distribution P : example of Metropolis-Hasting]

3.1 Markov Chains

A Markov Chain is a stochastic process where the distribution of x_{t+1} only depends on $x_t, P\left(x_{t+1} \in \ A \mid x_t, x_{t-1}, \ldots\right) = P\left(x_{t+1} \in A \mid x_t\right) \forall A$

Definition 1. A *transition kernel* is a function, $P(x, A)$, such that, for every x it is a probability measure in the second argument :

$$P(x, A) = P\left(x_{t+1} \in A \mid x_t = x\right)$$

It gives the probability of moving from x into the set A.

The transition kernel may have atoms, in particular, we would be considering cases with non-zero probability of (not moving) staying : $P(x, \{x\}) \neq 0$.
We want to study the behavior of a sequence of draws $x_1 \to x_2 \to \ldots$ where we move around according to a transition kernel. Suppose the distribution of x_t is $P^{(t)}$, then the distribution of $y = x_{t+1}$ is

$$P^{(t+1)}(y)dy = \int_{\mathbb{R}} P^{(t)}(x)P(x, dy)dx$$

Definition 2. A distribution π^* is called an *invariant measure* (with respect to transition kernel $P(x, A)$) if $\pi^*(y)dy = \int_{\mathbb{R}} \pi^*(x)P(x, dy)dx$

Under some regularity conditions, a transition kernel $P(x, A)$ has a unique invariant distribution π^*; and a marginal distribution $P^{(t)}$ of x_t- an element in Markov chain with the transitional kernel $P(x, A)$ converges to its invariant distribution π^* as $t \to \infty$. That is, if one would run a Markov chain long enough then the distribution of the draw is close to π^*. Generally, if the transition kernel is *irreducible* (it can reach any point from any other point) and aperiodic (not periodic, i.e. the greatest common denominator of $\{n : y$ can be reached from x in n steps $\}$ is 1), then it converges to an invariant distribution.
A classical Markov chain problem is to find π^* given $P(x, A)$. The MCMC has an inverse problem. Assume we want to simulate a draw from π^* (which we know up to a constant multiplier). We need to find a transition kernel $P(x, dy)$

such that π^* is its invariant measure. Let's suppose that π^* is continuous. We will consider the class of kernels

$$(*) \quad P(x, dy) = p(x, y)dy + r(x)\Delta_x(dy)$$

here $\Delta_x(dy)$ is a unit mass measure concentrated at point $x : \Delta_x(A) = \mathbb{I}\{x \in A\}$. So, the transition kernel $(*)$ says that we can stay at x with probability $r(x)$, otherwise y is distributed according to some pdf proportional to $p(x, y)$. Notice, that $p(x, y)$ isn't exactly a density because it doesn't integrate to 1. $\int P(x, dy) = 1 = \int p(x, y)dy + r(x); \int p(x, y)dy = 1 - r(x)$

Definition 3. A transition kernel is *reversible* if $\pi(x)p(x, y) = \pi(y)p(y, x)$

Theorem 4. If a transition kernel is reversible, then π is invariant.

Proof. We need to check that the definition of invariant distribution is satisfied

$$\begin{aligned}
\int_{\mathbb{R}} \pi(x)P(x, A)dx &= \int_{\mathbb{R}} \left(\int_A p(x, y)dy \right) \pi(x)dx + \int_{\mathbb{R}} r(x)\Delta_x(A)\pi(x)dx \\
&= \int_A \int_{\mathbb{R}} p(x, y)\pi(x)dxdy + \int_A r(x)\pi(x)dx \\
&= \int_A \int_{\mathbb{R}} p(y, x)\pi(y)dxdy + \int_A r(x)\pi(x)dx \\
&= \int_A \pi(y) \left(\int_{\mathbb{R}} p(y, x)dx \right) dy + \int_A r(x)\pi(x)dx \\
&= \int_A \pi(y)(1 - r(y))dy + \int_A r(x)\pi(x)dx = \pi(A)
\end{aligned}$$

In discrete time, we have :

Temporal Dynamics of a Markov Chain :
$P^{(t+1)}\left(X^{(t+1)} = x'\right) = \sum_x P^{(t)}\left(X^{(t)} = x\right) T(x \to x')$, where T is the transition distribution with $\sum_{x'} T(x \to x') = 1$ for all x.

Stationnary Distribution : $\pi(x') = \sum_x \pi(x)T(x \to x')$

Regular Markov Chains : A Markov Chain is said *regular* if there exists k such that for every x, x' the probability of getting from x to x' in exactly k steps is strictly positive (>0).

3.2 Regularity

Theorem : A regular Markov Chain converges to a unique stationnary distribution regardless of start state.

Sufficient conditions for regularity (used in practice) :

— Every two states x, x' are connected with probability >0

— For every state there is a self-transition

Example : If we take k to be the distance between furthest x, x' it works (as we can stay at the same state with self-transition)

3.3 Using Markov Chain in practice

Goal : Compute $\mathrm{P}(x \in S)$ (but P is too hard to sample directly).

Steps :

1. Construct a Markov Chain T whose unique stationnary distribution is P
2. Sample $x^{(0)}$ from some $\mathrm{P}^{(0)}$
3. For t = 0,1,2,... generate $x^{(t+1)}$ from $T(x \to x')$

Issues : We only want samples that are samples from a dsitribution close to P and at every iteration, $\mathrm{P}^{(t)}$ is usually far from P.
We want to start collecting samples only after the chain has run long enough to "**mix**" (i.e $\mathrm{P}^{(t)}$ close enough to π).

Question : How do we know when a chain has mixed ?

Answer : In general we cannot prove that a chain has mixed. However, we can prove in some situations that it has NOT mixed.

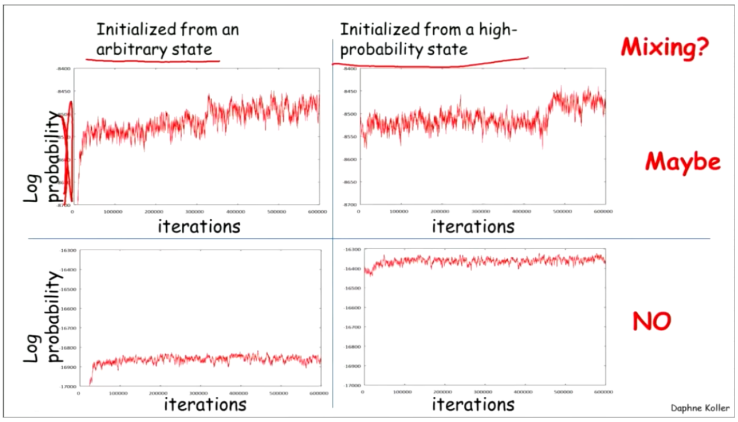
3.4 Proving a chain has not mixed

— Compare chain statistics in different windows within a single run of the chain

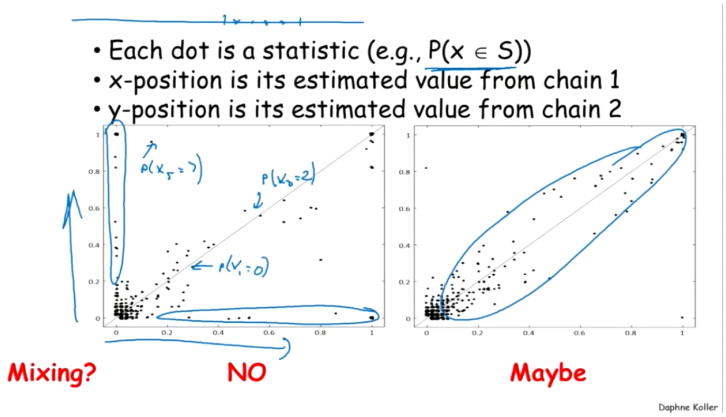
— Across different runs initialized differently (in case for example there are two clusters with low probability of transition between them such as below)

Question : What statistics can we use ?

Answer : Probability of a particular state (estimate $\mathrm{P}(x^i)$ for all i by calculating the fraction of sample in the window associated to the state), log probability (or even unnormalized) [TO COMPLETE]



We can also plot chain 1 vs chain 2 as show below.



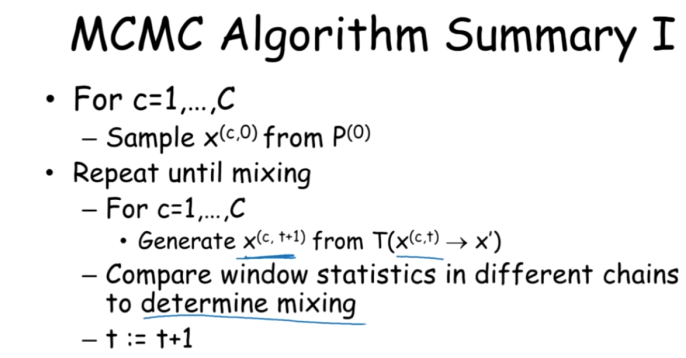
3.5 How to choose samples ?

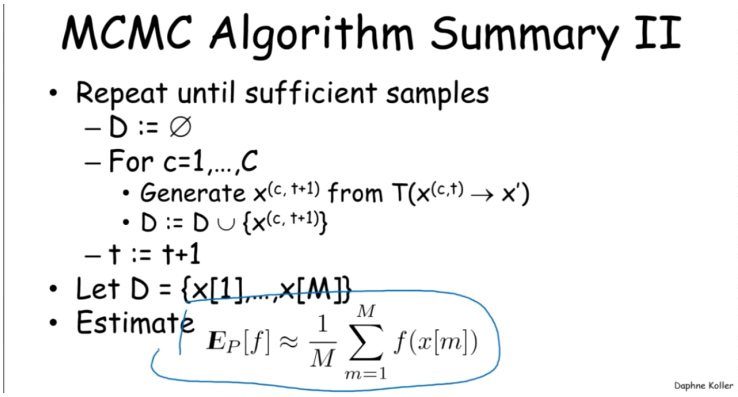
Once the chain mixes, all samples $x^{(t)}$ are from the stationnary distribution π (or close to).

Papers showed that it is better to collect every single sample $x^{(t)}$ for $t > T_{mix}$. Collecting all the samples might be expensive memory-wise (for example, if all samples need to be stored and passed to another program).

However, nearby samples are *correlated*. So we do not have i.i.d samples ! (Some papers indicate to take every hundred sample to avoid this).

3.6 Summary





Pros :

— Very general purpose

— Often easy to implement

— Good theoretical guarantees as $t \to \infty$

Cons :

— Lots of tunable parameters/design choices (mixing time, statistics to measure, number of samples, windows' size,...)

— Can be quite slow to converge

— Difficult to tell whether it is working

4 Metropolis-Hasting Algorithm

A good reference is Chib and Greenberg (The American Statistician 1995). Recall that the key object in Bayesian econometrics is the posterior distribution :

$$p\left(\theta \mid \mathcal{Y}_T\right) = \frac{f\left(\mathcal{Y}_T \mid \theta\right) p(\theta)}{\int f\left(\mathcal{Y}_T \mid \tilde{\theta}\right) d\tilde{\theta}}$$

It is often difficult to compute this distribution. In particular, the integral in the denominator is difficult. So far, we have gotten around this by using conjugate priors - classes of distributions for which we know the form of the posterior. Generally, it's easy to compute the numerator, $f\left(\mathcal{Y}_T \mid \theta\right) p(\theta)$, but it is hard to compute the normalizing constant, the integral in the denominator, $\int f\left(\mathcal{Y}_T \mid \vec{\theta}\right) d\bar{\theta}$. One approach is to try to compute this integral in some clever way. Another, more common approach is Markov Chain Monte-Carlo (MCMC). The goal here is to generate a random sample $\theta_1, \ldots, \theta_N$ from $p\left(\theta \mid \mathcal{Y}_T\right)$. We can then use moments from this sample to approximate moments of the posterior distribution. For example,

$$E\left(\theta \mid \mathcal{Y}_T\right) \approx \frac{1}{N} \sum \theta_n$$

There are a number of methods for generating random samples from an arbitrary distribution.

4.1 Acceptance-Rejection Method (AR)

We start with the simplest one. The goal is to simulate $\xi \sim \pi(x)$. What we know :

(1) A function, $f(x)$, such that $\pi(x) = \frac{f(x)}{k}$ The constant k is unknown (that is, f is a pdf up to an unknown normalization).

(2) We can simulate draws from some candidate pdf $h(x)$

(3) There is a known constant c such that $f(x) \leq \mathrm{ch}(x)$

We simulate draws from $\pi(x)$ as follows :

1. Draw $z \sim h(x), u \sim U[0,1]$
2. If $u \leq \frac{f(z)}{ch(z)}$, accept the draw $\xi = z$. Otherwise discard the draw and repeat (1)

The intuition of the procedure is the following :

Let $v = uch(z)$ and imagine the joint distribution of (v, z). It has support under the graph of $ch(z)$ with a uniform density (it is uniform on $\{(v,z) : z \in \text{ Support } (h), 0 \leq v \leq ch(z)\}$). Then, it is fairly easy to see that if we accept $\xi = z$, the joint distribution of (v, ξ) is uniform over the support $\{(v,\xi) : \xi \in \text{ Supportt } (\pi), f(\xi) \geq v \geq 0\}$. Then (for the same reason that $h(z)$ is the marginal density of (v, z)), the marginal density of ξ will be $\frac{f(\xi)}{k}$.

More formally,

Proof. Let ρ be the probability of rejecting a single draw. Then,

$$\begin{aligned} P(\xi \leq x) &= P\left(z_1 \leq x, u_1 \leq \frac{f(z_1)}{ch(z_1)}\right)\left(1+\rho+\rho^2+\ldots\right) \\ &= \frac{1}{1-\rho}P\left(z_1 \leq x, u_1 \leq \frac{f(z_1)}{ch(z_1)}\right) = \frac{1}{1-\rho}E_z\left[P\left(u \leq \frac{f(z)}{ch(z)} \mid z\right)\mathbf{1}_{\{z\leq x\}}\right] \\ &= \frac{1}{1-\rho}\int_{-\infty}^{x}\frac{f(z)}{ch(z)}h(z)dz = \int_{-\infty}^{x}\frac{f(z)}{c(1-\rho)}dz = \int_{-\infty}^{x}\pi(z)dz \end{aligned}$$

The last line is due to the fact that there exists the unique constant that normalizes f to be a pdf. since the left hand side is a cdf, then $\frac{1}{c(1-\rho)}$ is this constant. A major drawback of this method is that is may lead us to reject many draws before we finally accept one. This can make the procedure inefficient. If we choose c and $h(z)$ poorly, then $\frac{f(z)}{ch(z)}$ could be very small for many z. It will be especially difficult to choose a good c and $h()$ when we do not know much about $\pi(z)$.

4.2 The algorithm

The goal : we want to simulate a draw from the distribution π which we know up to a constant. That is, we can compute a function proportional to $\pi, f(x) = k\pi(x)$. We will generate a Markov chain with transition kernel of the form $(*)$, that will be reversible for π. Then if the chain will run long enough the element of the chain will have distribution π. The main question is how to generate such a Markov chain ?

Suppose we have a Markov chain in state x. Assume that we can draw $y \sim q(x,y)$, a pdf with respect to y (so $\int q(x,y)dy = 1$) . Consider using this q as a transition kernel. Notice that if

$$\pi(x)q(x,y) > \pi(y)q(y,x)$$

then the chain won't be reversible (we would move from x to y too often). This suggests that rather than always moving to the new y we draw, we should only move with some probability, $\alpha(x,y)$. If we construct $\alpha(x,y)$ such that

$$\pi(x)q(x,y)\alpha(x,y) = \pi(y)q(y,x)\alpha(y,x)$$

then we will have a reversible transition kernel with invariant measure π. We can take :

$$\alpha(x,y) = \min\left\{1, \frac{\pi(y)q(y,x)}{\pi(x)q(x,y)}\right\}$$

We can calculate $\alpha(x,y)$ because although we do not know $\pi(x)$, we do know $f(x) = k\pi(x)$, so we can compute the ratio.

In summary, the Metropolis-Hastings algorithm is : given x_t we move to x_{t+1} by :

1. Generate a draw, y, from $q(x_t, \cdot)$
2. Calculate $\alpha(x_t, y)$
3. Draw $u \sim U[0,1]$
4. $u < \alpha(x_t, y)$, then $x_{t+1} = y$. Otherwise $x_{t+1} = x_t$

This produces a chain with

$$P(x, dy) = q(y,x)\alpha(y,x)dy + r(x)\Delta_x(dy), \quad r(x) = 1 - \int q(y,x)\alpha(y,x)dy$$

Then the marginal distribution of x_t will converge to π. In practice, we begin the chain at an arbitrary x_0, run the algorithm many, say M times, then use the last $N < M$ draws as a sample from π. Note that although the marginal distribution of the x_t is π, the x_t are autocorrelated. This is not a problem for computing moments from the draws (although the higher the autocorrelation, the more draws we need to get the same accuracy), but if we want to put standard errors on these moments, we need to take the autocorrelation into account.

4.3 Choice of $q()$

- **Random walk chain** : $q(x,y) = q_1(y-x)$, i.e. $y = x + \epsilon, \epsilon \sim q_1$. This can be a nice choice because if q_1 is symmetric, $q_1(z) = q_1(-z)$, then $\frac{q(x,y)}{q(y,x)}$ drops out of $\alpha(x,y) = \min\left\{1, \frac{\pi(y)}{\pi(x)}\right\}$. Popular such q_1 are normal and $U[-a,a]$. Note that there is a tradeoff between step-size in the chain and rejection probability when choosing $\sigma^2 = E\epsilon^2$. Choosing σ^2 too large will lead to many draws of y from low probability areas (low π), and as a result we will reject lots of draws. Choosing σ^2 too small will lead us to accept most draws, but not move very much, and we will have difficulty covering the whole support of π. In either case, the autocorrelation in our draws will be very high and we'll need more draws to get a good sample from π.

- **Independence chain** : $q(x,y) = q_1(y)$

- If there is an additional information that $\pi(y) \propto \psi(y)h(y)$ where ψ is bounded and we can sample from $q(x,y) = h(y)$. This also simplifies $\alpha(x,y) = \min\left\{1, \frac{\psi(y)}{\psi(x)}\right\}$

- Autocorrelated $y = a + B(x-a) + \epsilon$ with $B < 0$, this leads to negative autocorrelation in y. The hope is that this reverses some of the positive autocorrelation inherent in the procedure.

5 Gibbs sampling

We use Gibbs sampling when we want to use MCMC to sample from posterior distribtions with multiple parameters.

One option is to perform Metropolis Hastings by sampling candidates for all the parameters at once. And accepting or rejecting all of those candidates together. While this is possible, it can get complicated.

Another simpler option is, to sample the parameters one at a time. The key to the Gibbs sampler is that one only considers univariate conditional distributions — the distribution when all of the random variables but one are assigned fixed values

5.1 Idea

The idea in Gibbs sampling is to generate posterior samples by sweeping through each variable (or block of variables) to sample from its conditional distribution with the remaining variables fixed to their current values.
Such conditional distributions are far easier to simulate than complex joint distributions and usually have simple forms (often being normals, inverse chi-2, or other common prior distributions). Thus, one simulates n random variables sequentially from the n univariate conditionals rather than generating a single n-dimensional vector in a single pass using the full joint distribution

To introduce the Gibbs sampler, consider a bivariate random variable (x, y), and suppose we wish to compute one or both marginals, $p(x)$ and $p(y)$. The idea behind the sampler is that it is far easier to consider a sequence of conditional distributions, $p(x \mid y)$ and $p(y \mid x)$, than it is to obtain the marginal by integration of the joint density $p(x, y)$, e.g., $p(x) = \int p(x, y)dy$. The sampler starts with some initial value y_0 for y and obtains x_0 by generating a random variable from the conditional distribution $p\left(x \mid y = y_0\right)$. The sampler then uses x_0 to generate a new value of y_1, drawing from the conditional distribution based on the value x_0 $p\left(y \mid x = x_0\right)$. The sampler proceeds as follows

$$x_i \sim p\left(x \mid y = y_{i-1}\right)$$
$$y_i \sim p\left(y \mid x = x_i\right)$$

Repeating this process k times, generates a Gibbs sequence of length k, where a subset of points (x_j, y_j) for $1 \leq j \leq m < k$ are taken as our simulated draws from the full joint distribution.

This process continues until "convergence" (the sample values have the same distribution as if they were sampled from the true posterior joint distribution). It produces a Markov chain, whose stationary distribution is the target or posterior distribution

When more than two variables are involved, the sampler is extended in the obvious fashion. In particular, the value of the k th variable is drawn from the distribution $p\left(\theta^{(k)} \mid \Theta^{(-k)}\right)$ where $\Theta^{(-k)}$ denotes a vector containing all off the variables but k. Thus, during the i th iteration of the sample, to obtain the value of $\theta_i^{(k)}$ we draw from the distribution

$$\theta_i^{(k)} \sim p\left(\theta^{(k)} \mid \theta^{(1)} = \theta_i^{(1)}, \cdots, \theta^{(k-1)} = \theta_i^{(k-1)}, \theta^{(k+1)} = \theta_{i-1}^{(k+1)}, \cdots, \theta^{(n)} = \theta_{i-1}^{(n)}\right)$$

For example, if there are four variables, (w, x, y, z), the sampler becomes

$$\begin{aligned}
w_i &\sim p\left(w \mid x = x_{i-1}, y = y_{i-1}, z = z_{i-1}\right) \\
x_i &\sim p\left(x \mid w = w_i, y = y_{i-1}, z = z_{i-1}\right) \\
y_i &\sim p\left(y \mid w = w_i, x = x_i, z = z_{i-1}\right) \\
z_i &\sim p\left(z \mid w = w_i, x = x_i, y = y_i\right)
\end{aligned}$$