Lecture Notes for "Stochastic Modeling and Computations"

M. Chertkov (lecturer), S. Belan and V. Parfeneyv (recitation instructors)
M.Sc. and Ph.D. level course at Skoltech
Moscow, March 28 - May 28, 2016
`https://sites.google.com/site/mchertkov/courses`

The course offers a soft and self-contained introduction to modern applied probability covering theory and application of stochastic models. Emphasis is placed on intuitive explanations of the theoretical concepts, such as random walks, law of large numbers, Markov processes, reversibility, sampling, etc., supplemented by practical/computational implementations of basic algorithms. In the second part of the course, the focus shifts from general concepts and algorithms per se to their applications in science and engineering with examples, aiming to illustrate the models and make the methods of solution, originating from physics, chemistry, machine learning, control and operations research, clear.

I. THEME # 2. STOCHASTIC PROCESSES

A. Lecture #6. Monte-Carlo Algorithms: General Concepts and Direct Sampling.

This lecture should be read in parallel with the respective IJulia notebook file. Monte-Carlo (MC) methods refers to a broad class of algorithms that rely on repeated random sampling to obtain results. Named after Monte Carlo -the city- which once was the capital of gambling (playing with randomness). The MC algorithms can be used for numerical integration, e.g. computing weighted sum of many contributions, expectations, marginals, etc. MC can also be used in optimization.

Sampling is a selection of a subset of individuals/configurations from within a statistical population to estimate characteristics of the whole population.

There are two basic flavors of sampling. Direct Sampling MC - mainly discussed in this lecture and Markov Chain MC. DS-MC focuses on drawing **independent** samples from a distribution, while MCMC draws correlated (according to the underlying Markov Chain) samples.

Let us illustrate both on the simple example of the 'pebble' game - calculating the value of π by sampling interior of a circle.

1. Direct-Sampling by Rejection vs MCMC for 'pebble game'

In this simple example we will construct distribution uniform within a disk from the distribution uniform over a square containing the circle. We will use direct product of two rand() to generate samples within the square and then simply reject samples which are not in the interior of the disk.

In the respective MCMC we build a sample (parameterized by a pair of coordinates) by taking previous sample and adding some random independent shifts to both variables, also making sure that when the sample crosses a side of the square it reappears on the opposite side. The sample "walks" the square, but we count (to compute area of the disk) only for samples which are within the disk (rejection again).

See IJulia notebook for an illustration.

2. Direct Sampling by Mapping

Direct Sampling by Mapping consists in application of the deterministic function to samples from a distribution you know how to sample from efficiently. The method is exact, i.e. it produces independent random samples distributed according to the new distribution. (We will discuss formal criteria for independence in the next lecture.)

For example, suppose we want to generate exponential samples, $y_i \sim \rho(y) = \exp(-y)$ – one dimensional exponential distribution over $[0,\infty]$, provided that one-dimensional uniform oracle, which generates independent samples, x_i from $[0,1]$, is available. Then $y_i = -\log(x_i)$ generates desired (exponentially distributed) samples.

Another example of DS MS by mapping is given by the Box-Miller algorithm which is a smart way to map two-dimensional random variable distributed uniformly within a box to the two-dimensional Gaussian (normal) random variable:

$$\int_{-\infty}^{\infty}\frac{dxdy}{2\pi}e^{-(x^2+y^2)/2}=\int_0^{2\pi}\frac{d\varphi}{2\pi}\int_0^{\infty}rdr\,e^{-r^2/2}=\int_0^{2\pi}\frac{d\varphi}{2\pi}\int_0^{\infty}dz\,e^{-z}=\int_0^1 d\theta\int_0^1 d\psi=1.$$

Thus, the deisred mapping is $(\psi, \theta) \to (x, y)$, where $x = \sqrt{-2\log\psi}\cos(2\pi\Theta)$ and $y = \sqrt{-2\log\psi}\sin(2\pi\theta)$.
See IJulia notebook for numerical illustrations.

3. *Direct Sampling by Rejection (another example)*

Let us now show how to get positive Gaussian (normal) random variable from an exponential random variable through rejection. We do it in two steps

- First, one samples from the exponential distribution: $x \sim \rho_0(x) = \begin{cases} e^{-x} & x > 0, \\ 0 & \text{otherwise} \end{cases}$
- Second, aiming to get a sample from the positive half of Gaussian, $x \sim \rho_0(x) = \begin{cases} \sqrt{2/\pi}\exp(-x^2/2) & x > 0, \\ 0 & \text{otherwise} \end{cases}$, one accepts the generated sample with the probability

$$p(x) = \frac{1}{M}\sqrt{2/\pi}\exp(x - x^2/2)$$

where M is a constant which should be larger than, $\max(\rho(x)/\rho_0(x)) = \sqrt{2/\pi}\, e^{1/2} \approx 1.32$, to guarantee that $p(x) \leq 1$ for all $x > 0$.

Note that the rejection algorithm has the advantage that it can be applied even when the probability densities are known only up to a multiplicative constant. (We will discuss issues related to this constant, also called in the multivariate case the partition function extensively)

4. *Importance Sampling*

As we saw MC is useful for computing sums/integrals/expectations. Suppose we want to compute an expectation of a function, $f(x)$, over the distribution, $\rho(x)$, i.e. $\int dx\rho(x)f(x)$, in the regime where $f(x)$ and $\rho(x)$ are concentrated around very different x. In this case overlap of f and $\rho(x)$ is small and as a result a lot of MC samples will be 'wasted'.

Importance Sampling is the method which helps to fix the small-overlap problem. It is based on adjusting the distribution function from $\rho(x)$ to $\rho_a(x)$ and then utilizing the following obvious formula

$$\mathbb{E}_\rho[f(x)] = \int dx\rho(x)f(x) = \int dx\rho_a(x)\frac{f(x)\rho(x)}{\rho_a(x)} = \mathbb{E}_{\rho_a}\left[\frac{f(x)\rho(x)}{\rho_a(x)}\right]$$

Consider an illustrative DS example, $\rho(x) = \frac{1}{\sqrt{2\pi}}\exp\left(-\frac{x^2}{2}\right)$ and $f(x) = \exp\left(-\frac{(x-4)^2}{2}\right)$, vs IS with the proposal distribution, $\rho_a(x) = \frac{1}{\sqrt{\pi}}\exp\left(-(x-2)^2\right)$.

This example is illustrated in the the IJulia notebook

The simple example shows that we are clearly wasting samples with DS. Note one big problem with IS. In the real (multi-dimensional) cases we typically do not have a good guess for $\rho_a(x)$. One way of solving this problem is to search for good $\rho_a(x)$ adaptively. A sophisticated adaptive importance sampling package is available at `https://pypi.python.org/pypi/pypmc/1.0`.

See

- `https://statmechalgcomp.wikispaces.com/`
- `http://www.math.nyu.edu/faculty/goodman/teaching/Monte_Carlo/direct_sampling.ps`
- `http://www.cs.berkeley.edu/~jordan/courses/260-spring10/lectures/lecture17.pdf`

for additional material/discussions of DS-MC.

5. Direct Brut-force Sampling

This algorithm relies on availability of the uniform sampling algorithm from $[0,1]$, rand(). One splits the $[0,1]$ interval into pieces according to the weights of all possible states and then use rand() to select the state. The algorithm is impractical as it requires keeping in the memory information about all possible configurations. The use of this construction is in providing the bench-mark case useful for proving independence of samples.

6. Direct Sampling from a multi-variate distribution with a partition function oracle

So far we have discussed sampling from a single-valued distribution. Suppose we have an oracle capable of computing the partition function (normalization) for the multivariate probability itself and also for any of the marginal probabilities. (Notice that we are not asking about complexity of the oracle, at least not for now.) Does it give us the power to generate independent samples?

We get affirmative answer to this question through the following **decimation** algorithm generating independent sample $x \sim P(x)$, where $x \doteq (x_i | i = 1, \cdots, N)$:

Algorithm 1 Decimation Algorithm

Input: $P(x)$ (expression). Partition function oracle.

$x^{(d)} = \varnothing; \quad I = \varnothing$
while $|I| < N$ **do**
  Pick i at random from $\{1, \cdots, N\} \setminus I$.
  $x^{(I)} = (x_j | j \in I)$
  Compute $P(x_i | x^{(d)}) \doteq \sum_{x \setminus x_i ; x^{(I)} = x^{(d)}} P(x)$ with the oracle.
  Generate random $x_i \sim P(x_i | x^{(d)})$.
  $I \cup i \leftarrow I$
  $x^{(d)} \cup x_i \leftarrow x^{(d)}$
end while

Output: $x^{(\text{dec})}$ is an independent sample from $P(x)$.

Validity of the algorithm follows from the following exact representation for the joint distribution function via a chain of ordered conditional distribution function (chain rule for distribution):

$$P(x_1, \cdots, x_n) = P(x_1) P(x_2|x_1) P(x_3|x_1, x_2) \cdots P(x_n|x_1, \cdots, x_{n-1}). \tag{I.1}$$

(The chain rule follows directly from the Bias rule. Notice also that ordering of variables in the chain rule is arbitrary.) One way of proving that an algorithm produce independent sample is to show that the algorithm outcome is equivalent to another algorithm for which the independence is already proven. The benchmark algorithm we can use to state that the Decimation algorithm (1) produces independent samples is the brute-force sampling algorithm described in the beginning of the lecture. The crucial point here is in splitting the $[0,1]$ interval hierarchically, first according to $P(x_1)$, then subdividing pieces for different x_1 according to $P(x_2, x_1)$, etc. This guidanken experiment will result in the desired proof.

In general efforts of the oracle (for computing the partition function) are exponential. However in some special case the partition function can be computed efficiently (polynomially in the number of steps). For example this is the case for (glassy) Ising model without magnetic field over planar graph. See `http://surface.syr.edu/cgi/viewcontent.cgi?article=1176&context=phy` and references there in for details.

7. Ising Model

Let us digress and consider the Ising model – example of a larger class of important/interesting multi-variate statistics often referred to (in theoretical engineering) as Graphical Models (GM). We will study GM in even more details later. Consider a system of spins or pixels (binary variables) on a graph, $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is a set of nodes/vertices and $\mathcal{E}$ is the set of edges. The graph may be 1d chain, tree, 2d lattice ... or any other graph. (The cases of regular lattices are prevalent in physics, while graphs of interest for engineering are, generally, richer.) Consider a binary variable, residing at every node of the graph, $\forall i \in \mathcal{V}: \ \sigma_i = \pm 1$, we call them "spins". If there are N spins in the system, 2^N is the number of possible configuration of spins — notice exponential scaling with N, meaning, in particular that just counting the number of configurations is "difficult". If we would be able to do it in algebraic/polynomial number of steps, we would call it "easy", or rather "theoretically easy", while the practically easy case - which is the goal – would correspond to the case when "complexity" of, say counting, would be $O(N)$ -

linear in N at $N \to \infty$. (Btw $o(N)$ is the notation used to state that the behavior is actually slower than $O(N)$, say $\sim \sqrt{N}$ at $N \to \infty$, i.e. asymptotically $o(N) \ll O(N)$.) In magnetism (field of physics where magnetic materials are studied) probability of a spin configuration (vector) is

$$p(\sigma) = \frac{\exp\left(-\beta E(\sigma)\right)}{Z}, \quad E(\sigma) = -\frac{1}{2}\sum_{\{i,j\}\in\mathcal{E}} \sigma_i J_{ij} \sigma_j + \sum_{i\in\mathcal{V}} h_i \sigma_i, \tag{I.2}$$

$$Z = \sum_{\sigma} \exp\left(-\beta E(\sigma)\right). \tag{I.3}$$

$E(\sigma)$ is energy of the given spin configuration, σ. The first term in $E(\sigma)$ is pair-wise (wrt nodal spins), spin exchange/interaction term. The last term in $E(\sigma)$ stands from (potentially node dependent) contribution of the magnetic field, $h = (h_i | i \in \mathcal{V})$ on individual spins. Z is the partition function - weighted sum of the spin configurations. Introduced to enforce the normalization condition, $\sum_\sigma p(\sigma) = 1$. For a general graph, arbitrary values of J and h, Z is the difficult object to compute, i.e. complexity is $O(2^N)$. (Notice that for some special cases, such as when graph is a tree, or when graph is planar and $h = 0$, the computations become easy.) Moreover computing other important characteristics, such as the most probable configuration of spins

$$\sigma_{ML} = \arg\max_{\sigma} p(\sigma), \tag{I.4}$$

also called Maximum Likelihood and Ground State in information sciences and physics respectively, or probability of observing a particular node in state σ_i (can be $+$ or $-$)

$$p_i(\sigma_i) = \sum_{\sigma\setminus\sigma_i} p(\sigma), \tag{I.5}$$

are also difficult problems. (Wrt notations – $\arg\max$ - pronounced argmax - stands for particular σ at which the maximum in Eq. (I.4) is reached. $\sigma \setminus \sigma_i$ in the argument of the sum in Eq. (I.5) means that we sum over all σ consistent with the fixed value of σ_i at the node i.)

B. Lecture #7. Markov-Chain Monte-Carlo.

Markov Chain Monte Carlo (MCMC) methods belong to the class of algorithms for sampling from a probability distribution based on constructing a Markov chain that converges to the target steady distribution.

Examples and flavors of MCMC are many (and some are quite similar) – heat bath, Glauber dynamics, Gibbs sampling, Metropolis Hastings, Cluster algorithm, Warm algorithm, etc – in all these cases we only need to know transition probability between states while the actual stationary distribution may be not known or, more accurately, known up to the normalization factor, also called the partition function. Below, we will discuss in details two key examples: Gibbs sampling & Metropolis-Hastings.

1. Gibbs Sampling

Assume that the direct sampling is not feasible (because there are too many variables and computations are of "exponential" complexity — more on this latter). The main point of Gibbs sampling is that given a multivariate distribution it is simpler to sample from a conditional distribution than to marginalize by integrating over a joint distribution. Then we create a chain: start from a current sample of the vector x, pick a component at random, compute probability for this component/variable conditioned to the rest, and sample from this conditional distribution. (The conditional distribution is for a simple component and thus it is easy.) We continue the process till convergence, which can be identified (empirically) by checking if estimation of the histogram or observable(s) stopped changing.

Question (specific): Does the Gibbs sampling obey the Detailed Balance? Show it.

Question (generic): if one wants to prove convergence of an MCMC, how one can do it? (What is the expectation/average to study?)

2. Metropolis-Hastings Sampling

Metropolis-Hastings sampling is an MCMC method which explores efficiently detailed balance, i.e. reversibility of the underlying Markov Chain. The algorithm also uses sampling from the conditional probabilities and smart use of the rejection

Algorithm 2 Gibbs Sampling

Input: Given $p(x_i|x_{\sim i} = x \setminus x_i), \quad \forall i \in \{1, \cdots, N\}$. Start with a sample $x^{(t)}$.

loop Till convergence
  Draw an i.i.d. i from $\{1, \cdots, N\}$.
  Generate a random $x_i \sim p\left(x_i|x_{\sim i}^{(t)}\right)$.
  $x_i^{(t)} = x_i$.
  $\forall j \in \{1, \cdots, N\} \setminus i: \quad x_j^{(t)} \leftarrow x_j^{(t)}$.
  Output $x^{(t+1)}$ as the next sample.
end loop

strategy. Assume that the probability of any state x from which one wants to sample (call it the target distribution) is explicitly known up to the normalization constant, Z, i.e. $p(x) = \pi(x)/Z$, where $Z = \sum_x \pi(x)$. Let us also introduce the so-called proposal distribution, $p(x'|x)$, and assume that drawing a sample proposal x' from the current sample x is (computationally) easy.

Algorithm 3 Metropolis-Hastings Sampling

Input: Given $\pi(x)$ and $p(x'|x)$. Start with a sample x_t.

loop Till convergence
  Draw a random $x' \sim p(x'|x_t)$.
  Compute $\alpha = \frac{p(x_t|x')\pi(x')}{p(x'|x_t)\pi(x_t)}$.
  Draw random $\beta \in U([0, 1])$, uniform i.i.d. from $[0, 1]$.
  if $\beta < \min\{1, \alpha\}$ **then**
    $x_t \leftarrow x'$ [accept]
  else
    x' is ignored [reject]
  end if
  x_t is recordered as a new sample
end loop

Note that the Gibbs sampling previously introduced can be considered as the Metropolis-Hastings without rejection (thus it is a particular case).

Exercise: Arguing in terms of transitions between states show that the algorithm maintains the DB.

The proposals (conditional probabilities) may vary. Details are critical (change mixing time), especially for large system. There is a (heuristic) rule of thumb: **lower bound on number of iterations of MH**. If the largest distance between the states is L, the MH will mix in time

$$T \approx (L/\varepsilon)^2 \tag{I.6}$$

where ε is the typical step size of the random walk.

Question: How can one reason the quadratic behavior in Eq. (I.6)? What would acceleration from quadratic to linear mean? (Diffusive vs ballistic regime.)

Mixing may be extremely slow if the proposal distribution is not selected carefully. Let us illustrate how slow MCMC can be on a simple example. (See Section 29 of the McKay book for extra details.) Consider the following target distribution over N states

$$\pi(x) = \begin{cases} 1/N & x \in \{0, \cdots, N-1\} \\ 0 & \text{otherwise} \end{cases} \tag{I.7}$$

and proposal distribution over $N + 2$ states (extended by -1 and N)

$$p(x'|x) = \begin{cases} 1/2 & x' = x \pm 1 \\ 0 & \text{otherwise} \end{cases} \tag{I.8}$$

Notice that the rejection can only occur when the proposed state is $x' = -1$ or $x' = N$.

A more sophisticated example of the Glauber algorithm (version of MH) on the example of the Ising Model is to be discussed next.

3. Glauber Sampling of Ising Model

Let us return to the special version of MH developed specifically for the Ising model - the Glauber dynamics/algorithms:

Algorithm 4 Glauber Sampling

Input: Ising model on a graph. Start with a sample σ

loop Till convergence
  Pick a node i at random.
  $-\sigma_i \leftarrow \sigma_i$
  Compute $\alpha = \exp\left(\sigma_i \left(\sum_{j\in\mathcal{V}:\{i,j\}\in\mathcal{E}} J_{ij}\sigma_j - 2h_i\right)\right)$.
  Draw random $\beta \in U([0,1])$, uniform i.i.d. from $[0,1]$.
  if $\alpha < \beta < 1$ **then**
    $-\sigma_i \leftarrow \sigma_i$ [reject]
  end if
  Output: σ as a sample
end loop

Question: What is the proposal distribution turning the MH sampling into the Glauber sampling (for the Ising model)?

Exercise [Advanced]: Consider running parallel dynamics, based on the Glauber algorithm, i.e. at every moment of time update all variables in parallel according to the Glauber Sampling rule applied to the previous state. What is the resulting stationary distribution? Is it different from the Ising model? Does the algorithm satisfy the DB conditions?

For useful additional reading on sampling and computations for the Ising model see `https://www.physik.uni-leipzig.de/~janke/Paper/lnp739_079_2008.pdf`. MCMC recitation will focus on discussion of the Glauber algorithm.

4. Exactness and Convergence

MCMC algorithm is called exact if one can show that the generated distribution "converges" to the desired stationary distribution. However, "convergence" may mean different things.

The strongest form of convergence – **exact convergence** (warning - this is our 'custom' term) – states that in a finite number of steps we generate an independent sample from the target distribution. To prove this statement one needs to show that empirical correlations of consecutive independent samples are zero in the limit when Nnumber of samples $\to \infty$:

$$\lim_{N\to+\infty} \frac{1}{N}\sum_{n=0}^{N} \mathbb{E}\left[\sum_{n=1}^{N} f(x_n)g(x_{n-1})\right] \to \mathbb{E}\left[f(x)\right]\mathbb{E}\left[g(x)\right], \tag{I.9}$$

where $f(x)$ and $g(x)$ are arbitrary (however such that respective expectations on the rhs of Eq. (**??**) are well-defined).

A weaker convergence – call it **asymptotic convergence** – states that in the limit of $N \to \infty$ we reconstruct the target distribution (and all respective existing moments):

$$\lim_{N\to+\infty} \frac{1}{N}\sum_{n=0}^{N} \mathbb{E}\left[\sum_{n=1}^{N} f(x_n)\right] \to \mathbb{E}\left[f(x)\right], \tag{I.10}$$

where $f(x)$ is an arbitrary function such that the expectation on the rhs is well defined.

Finally, the weakest form of convergence – call it **parametric convergence** – corresponds to the case when one arrives at the target estimates only in a special limit with respect to a special parameter (for example, in the limit when the number of spins in the Ising model becomes infinite - this specific limit is called thermodynamic limit in statistical physics):

$$\lim_{s\to s_*}\lim_{N\to+\infty} \frac{1}{N}\sum_{n=0}^{N} \mathbb{E}\left[\sum_{n=1}^{N} f_s(x_n)\right] \to \mathbb{E}\left[f_{s_*}(x)\right]. \tag{I.11}$$

For additional math (but also intuitive as written for physicists) reading on the MCMC (and in general MC) convergence see `http://statweb.stanford.edu/~cgates/PERSI/papers/mixing.pdf` and also "Nature of Computations" by Moore and Mertens.

(This part of the lecture is a bonus material - we discuss it only if time permits.)

The material follows Chapter 32 of D.J.C. MacKay book. Some useful set of modern references and discussions are also available at `http://dimacs.rutgers.edu/~dbwilson/exact/`.

As mentioned already the main problem with MCMC methods is that one need to wait (and sometimes for too long) to make sure that the generated samples (from the target distribution) are i.i.d. If one starts to form a histogram (empirical distribution) too early it will deviate from the target distribution. One important question in this regards is: For how long shall one run the Markov Chain before it has 'converged'? To answer this question (prove) it is very difficult, in many cases not possible. However, there is a technique which allows to check the **exact convergence**, for some cases, and do it on the fly - as we run MCMC.

This smart technique is the Propp-Wilson exact sampling method, also called **coupling from the past**. The technique is based on a combination of three ideas:

- The main idea is related to the notion of **trajectory coalescence**. Let us observe that if starting from different initial conditions MCMC chains share a single random number generator, then their trajectories in the phase space can coalesce; and having coalesced, will not separate again. This is clearly an indication that the initial conditions are forgotten.

  Will running All the initial conditions forward in time till coalescence generate exact sample? Apparently not. One can show (sufficient to do it for a simple example) that the point of coalescence does not represent an exact sample.

- However, one can still achieve the goal by **sampling from a time T_0 in the past**, up to the present. If the coalescence has occurred the present sample is an unbiased sample; and if not we restart the simulation from the time T_0 further into the past, reusing the same random numbers. The simulation is repeated till a coalescence occur at a time before the present. One can show that the resulting sample at the present is exact.

- One problem with the scheme is that we need to test it for all the initial conditions - which are too many to track. Is there a way to **reduce the number of necessary trials**. Remarkably, it appears possible for sub-class of probabilistic models the so-called **'attractive'** models. Loosely speaking and using 'physics' jargon - these are **'ferromagnetic'** models - which are the models where for a stand alone pair of variables the preferred configuration is the one with the same values of the two variables. In the case of attractive model monotonicity (sub-modularity) of the underlying model suggests that the paths do not cross. This allows to only study limiting trajectories and deduce interesting properties of all the other trajectories from the limiting cases.