Optimum Monte-Carlo Sampling Using Markov Chains With Applications to Bootstrapping

Giorgio Sgarbi

December 11, 2019

1 Background

1.1 Traditional Bootstrapping

Bootstrapping is a relatively well-understood and widely used re-sampling method basically consisting of independent re-sampling steps with replacement. The paper by **Peskun (1973)**, however, makes us wonder if we can implement his idea of replacing independent re-sampling steps with dependent ones in the context of bootstrapping.

A typical bootstrapping process consists of the following:

a) Given a set of samples S $= \{s_1, ..., s_n\}$, pick n samples with replacement from that set, generating a bootstrap sample $S^* = \{s_1^*, ..., s_n^*\}$.

b) Repeat this step m times.

c) Use your m bootstrap samples to estimate the distribution for a desired test statistic.

We show a code in R as example, with a histogram in the following page:

```
set.seed(547)
# number of iid random variables in our model
N = 500
# number of times the algorithm will run
M = 1000

# this distribution is unknown to the bootstrap user
x = rnorm(N, 2, 1.5)

bmeans = c()
bvars = c()

for (i in 0:M) {
  # randomly select N samples *with replacement*
  bsample = sample(x, size=N, replace=TRUE)

  bmean = mean(bsample)
  bmeans = c(bmeans, bmean)

  # example calculating another statistic: variance
  bvar = (N-1)/N*var(bsample)
  bvars = c(bvars, bvar)
  }

# plot histogram for means
hist(bmeans, breaks = 20)
```

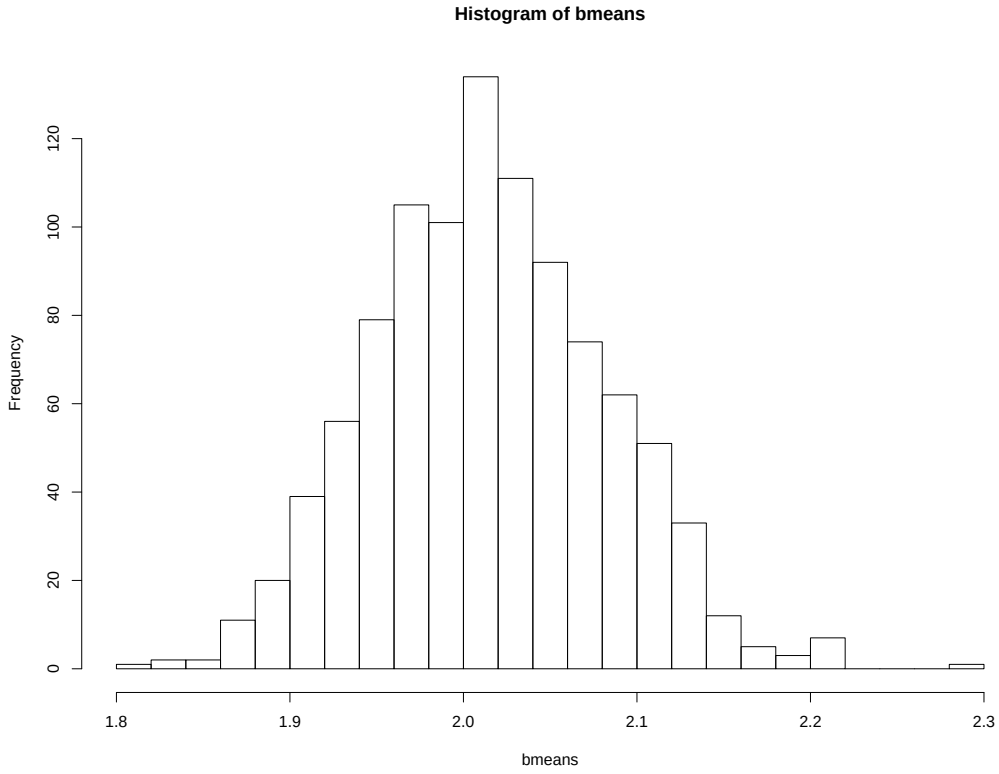

Figure 1: A histogram of bootstrap means.

1.2 Markov Chains

The following definitions and properties will be the base to the the second part of this project. The definitions and theorems in this section use as guide professors Ben's and Geyer's notes. I chose to add them here, even if not in depth, because they provide notions or vocabulary necessary to understand part 2.

1.2.1 Transition Kernel

Transition Kernel definition: Let (E,$\mathcal{E}$) and (F,$\mathcal{F}$) be two measurable spaces, and let K be a mapping from E x $\mathcal{F}$ into $\bar{R}_+$. Then K is called a **transition kernel** from (E,$\mathcal{E}$) into (F,$\mathcal{F}$) if:

a) for any fixed B in $\mathcal{F}$, K(x,B), as a function of x, is E-measurable; and

b) the mapping B $\rightarrow$ K(x,B) is a measure on (F,$\mathcal{F}$) for every x in E.

1.2.2 Algebra of Kernels

Given a measure λ, a measurable function f and kernels K, L, we have:

a) $\lambda K(B) \stackrel{def}{=} \int \lambda(dx)K(x,B), B \in \mathcal{E}$. This operation will produce the **new measure** λK.

b) $KL(x,B) \stackrel{def}{=} \int K(x,dy)L(y,B), B \in \mathcal{E}$. This operation will produce the **new kernel** KL. Notice that, by definition, $KL \neq LK$.

c) $Kf \stackrel{def}{=} \int K(x,dy)f(y)$ given that the integral exists. This operation will produce the **new measurable function** Kf.

1.2.3 Definition: Markov Chain

Special attention is given to **finite state spaces** in this section as they are the type of state space in our application.

A stochastic process $X_1, X_2, ...$ taking values in a measurable space, which is called the **state space**, is a **Markov chain** if the conditional distribution of the future given the past and present depends only on the present.

Notation: We assume the conditional distribution of X_{n+1} given X_n is given by a **Markov kernel P**, which is just a kernel that has a few extra properties. We discuss more about it in 2.2.2.

1.2.4 Finite state space: operations

In section 1.2.2, we defined operations in a familiar way, for instance being careful with the side by which we performed products, the same way we do when multiplying matrices. If we treat measurable functions as column vectors, kernels as matrices, and (probability) measures as row vectors, the operations work the same as the way we learned in a linear algebra course.

1.2.5 Definition: Irreducible kernel

A kernel P is irreducible if there is a measure φ such that: for every $x \in E$ and φ-positive $A \in \mathcal{A}$, there exists a positive integer n such that $P^n(x, A) > 0$. In such a case, we also say φ is an irreducibility measure for P or P is φ-irreducible.

1.2.6 Communicating states

A set $B \in \mathcal{A}$ is φ-communicating if for every $x \in B$ and every φ-positive $A \in A$ such that $A \subset B$ there exists a positive integer n such that $P_n(x, A) > 0$. A kernel P is φ-irreducible if and only if the whole state space is φ-communicating.

1.2.7 Finite state space: irreducibility

When we have a countable state space, irreducibility and existence of paths are associated.This will be the case in this project as our spaces will be more than countable: they will be finite. A **path** from $x = x_1$ to $y = x_n$ is a finite sequence of states $x_1, ..., x_n$ such that: $P(x_i, x_{i+1}) > 0, i = 1, ..., n-1$. If there exists a state y such that there is a path from x to y for every $x \in E$, then the kernel is irreducible.

1.2.8 Theorem: Irreducible kernels and invariant measures

A measure λ is invariant (by a kernel K) if $\lambda K = \lambda$. If a Markov kernel is irreducible and has an invariant measure, then the invariant measure is unique up to multiplication by positive constants. A proof can be seen in Theorems 10.0.1 and 10.1.2 in Meyn and Tweedie (2009), which can be found in Geyer's notes.

1.3 Estimation and variation

In this section we will discuss a little notation and terminology used in estimation.

Assume we have an irreducible Markov chain with discrete and finite states 1, ..., S. Recall that irreducibility guarantees that if there's an invariant measure π for our kernel P, then that invariant measure is unique. Let f be a measurable function. Define A as:

$$A = E_\pi[f(X)] = \sum_{i=1}^{S} f(i)\pi$$

We simulate our Markov chain with kernel P for times 1, ..., N and estimate A as follows:

$$\tilde{A} = \sum_{t=1}^{N} f(X(t))/N$$

Let now the asymptotic variance $v(f, \pi, P)$ be

$$v(f, \pi, P) = \lim_{N \to \infty} N\text{Var}(\tilde{A})$$

Those quantities will be important when we discuss our research question.

2 Open questions and research directions

2.1 Research question and goals

Our goal is to change the sample step in the bootstrap procedure described in 1.1 using a kernel that assigns probabilities of picking states, with those probabilities now **depending on the present sample** in such a way that our sampling is optimized. By "optimized", we mean we want to reduce the asymptotic variance v of our estimator $\tilde{I}$, as defined in 1.3, when compared to using the traditional bootstrapping discussed in 1.1. To numerically measure our accuracy or how "optimal" we are, we will measure the quantity v in our simulations, with low values of it indicating a better estimation. Our main way to create those new kernels will be to keep the kernel from connecting any state to itself. Intuitively, this will prompt our chain to visit more states, increasing the precision of our estimate and reducing its variance, as pointed out in the discussion of Theorem 2.1.1 from Optimum Monte-Carlo Sampling Using Markov Chains, **Peskun (1973)**.

2.2 Gathering our tools

We will now narrow down this general definition according to our context and discuss each of the terms involved:

1. Sample Spaces (and sigma-algebra):
   In our context of bootstrapping, E and F will be finite and discrete. Both will be equal to the state space with all possible bootstrap samples[1]. A natural choice for $\mathcal{E}$ is 2^E, which will denote the power set of E. We still need a concrete description of our state space in the bootstrapping case. How we represent our state space is actually up to us to choose. Say we are initially given the following samples $S_0 = \{s_1, ..., s_n\}$. We can choose to denote, for instance, this set as $123...n$. In the example below, we start with the set of samples $\{1.67, 2.3, 2.5\}$. After our first bootstrap step, we pick the

   Example:

   $S_0 = \{1.67, 2.3, 2.5\} := 123$
   $S_1 = \{2.3, 2.3, 1.67\} := 221$

   Our state space will be then be the set of all possible permutations with repetitions of 1, 2, ..., n. This is example uses the way we may refer to as the **standard way** to represent our states.

2. Kernel K:
   Our kernel K will also have some restrictions. The condition in 1.2.1 b) will be extended to: K(x,B) is a **probability** measure: K will always take non-negative values and be such that $K(x, E) = 1, \forall x \in E$. This type of kernel is called a Markov or stochastic kernel.

2.3 Re-sample step

Given a sample, we now wish to choose a new way to select the next one, in such way that it depends on the previous sample only. Using kernels, that means that we need to assign a probability to each singleton in $\mathcal{E}$, i.e, each of our bootstrap samples (or states). This will be enough as any other element of ϵ can be expressed as a disjoint union of singletons. Assigning those probabilities, or equivalently, the values of the kernel, can be done in many different ways. However, based on our motivational paper, we will choose to select ways that avoid re-sampling the same state again. Perhaps the simplest one is to select one of the other states uniformly, as we show in the example below:

Example:

Let S be the set of all singletons in our sigma-algebra (i.e, our state space) and n be the number of initial samples we are given before we start bootstrapping. First, notice that $S \subset \mathcal{E}$. For all x in E, for all s in $S - \{x\}$, define $K(x, \{s\}) = \dfrac{1}{n^n - 1}$. Naturally, we still need to assign a probability for x to return to itself and that will be 0, i.e, $K(x, x) = 0$.

[1] The transition idea will be a bit lost here if you prefer to see that way as we are having our 'transitions' in the same space, which I will simply call E.

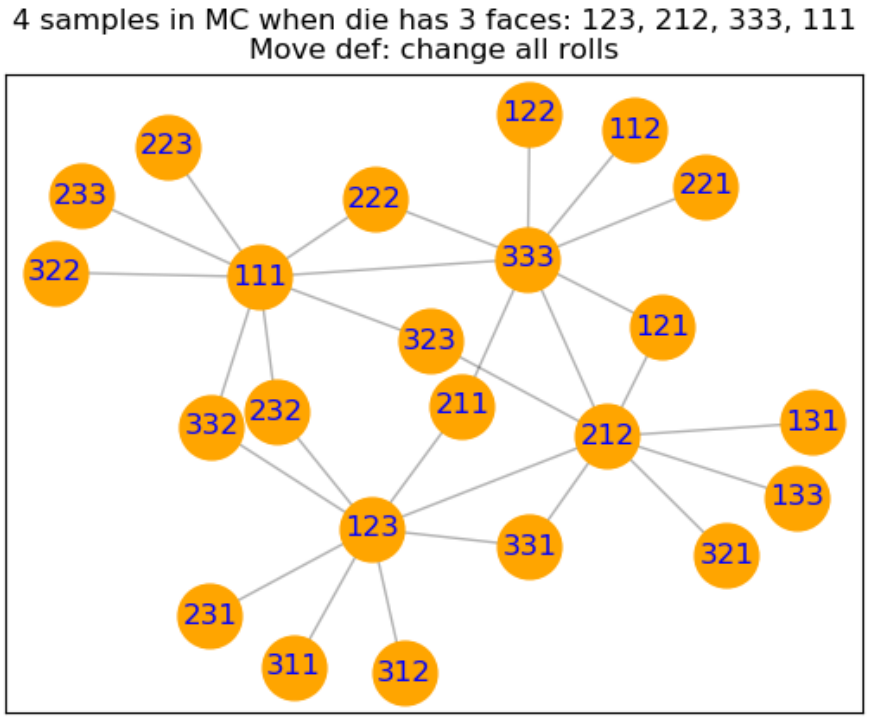

Figure 2: All neighboors for states 123, 212, 333, and 111 are shown with Kernel defined as: change all rolls (or entries).

2.4 Simulating a more elaborate Markov chain

As we mentioned, we can create more elaborate examples than one above while still avoiding self-sampling. To give more context, we may alternatively think, for n = 3, that we are rolling 3 RPG die with 3 faces each: 1, 2, 3, and recording the results. Our kernel K will be defined as follows:

Let a state $j = j_1 j_2 j_3$ be accessible from a state $i = i_1 i_2 i_3$ if $j_k \neq i_k$ for k = 1, 2, 3 . Define the kernel $K(i, \{j\})$ as: uniformly pick an accessible state y from state x. The result is a chain that still avoids picking the same state consecutively, having a kernel that picks an accessible state with probability 2^{-3}.

2.5 Running the provided code

In the graph above, only the states 111, 123, 212, and 333 and all states that communicate with them are shown to avoid a crowded plot.

In this project, we provide a **code** that simulates a Markov chain with the kernel we just defined. You can run the code by running the setup and then running from the source folder: **"python3 dependent_bootstrap n"**, where n is the size of E (number of singletons in the sigma-algebra or number of "entries" in our standard notation if you prefer). Your samples will be saved in the **"deliverables"** folder, which will be created when you run it for the first time. The program will prevent you from running the code with $n < 3$.

2.6 Next steps

Our next steps will be checking that we obtain the desired invariant measures π in concrete examples, such as the ones we just saw in the previous sections, and then finally we can compare the asymptotic variance

associated to different kernels. If our theory is correct, we expect to see lower asymptotic variance when we use kernels that do not allow self re-sampling compared to those that do or to the traditional bootstrapping method.

3 References

Meyn, S. P. and Tweedie, R. L. (2009). Markov Chains and Stochastic Stability, second edition. Cambridge: Cambridge University Press.

Peskun, P. (1973). Optimum Monte-Carlo Sampling Using Markov Chains. Biometrika, 60(3), 607-612. doi:10.2307/2335011

Ben's notes: https://ben-br.github.io/stat-547c-fall-2019/assets/notes/lecture-notes.pdf

Geyer's notes: http://www.stat.umn.edu/geyer/8112/notes/markov.pdf

A Exercises

Instructions: Some exercises allow more than one possible answer - in those cases, an example of such an answer is provided as solution. We use the standard state notation described in section 2 unless stated otherwise. We use the measurable spaces E and F (or just E if E=F) and their sigma-algebras as we defined in part 1.

Exercise 1

(a) How would you define the identity kernel so that it behaves as expected with respect to the kernel operations we defined?

(b) Go back to section 1.2. Show that your definition is consistent with the properties of kernels given in 1.2.1.

Exercise 2

(a) Come up with a way to describe our state space that is different from our standard way (see part 2.2).

(b) How would you represent the elements 123 and 221?

(c) Using your notation, how many elements would your state space have?

Exercise 3

Let $E = \{1, 2, ..., n\}$. Let a state $j = j_1...j_n$ be accessible from a state $i = i_1...i_n$ if $\{j_1, ..., j_n\} \subset E \setminus \{i_1, ..., i_n\}$. Define the kernel $K(i, \{j\})$ as: uniformly pick an accessible state j from state i.

(a) List (or draw) the states that communicate with 1441.

(b) Argue that this chain is not irreducible.

(c) If you were to simulate such a chain, how can you make it irreducible by making a small adaptation that fixes the problem you saw in part b?

PLEASE TURN PAGE FOR ANSWERS

Answer of exercise 1

(a) We need I to behave as following: $KI = IK = K, If = f, \lambda I = \lambda$, for all kernel K, measurable function f, and measure λ. Therefore, according to the operation definitions we discussed, we define I as:

$$I(x, B) = \begin{cases} 0 & \text{if } x \notin B \\ 1 & \text{if } x \in B \end{cases}$$

for all x in E and $B \in \mathcal{F}$.

(b) Our definition is consistent with the properties of kernels given in part 1 since:

(i) for a fixed x, we have that $I = \delta_x(B)$, which is a measure.

(ii) for a fixed $B \in \mathcal{F}$, $I = \mathbb{1}_B(x)$, which is a measurable function.

Answer of exercise 2

(a) One possibility is $i_1...i_n$, where i_k counts the number of times element k is picked. Notice that, in this case, $\sum i_k = n$.

(b) 123 means we have a sample from each, so the representation would be 111. The representation of 221 would be 021.

(c) We will not have as many as 3^3 possibilities as with our standard notation because of our constraint that $\sum i_k = n$. Rather, the possible states are in a bijection with the number of permutations of "2 bars and 3 balls": $O|O|O$. For instance, 111 and 021 are represented by $O|O|O$ and $|OO|O$, respectively. Think of this as 2 bars separating a total of 3 balls into 3 groups. Therefore, we have a total of $\frac{5!}{3!2!} = \binom{5}{3}$ elements. For the general case, where an element is described as n entries adding up to n, we have $\binom{2n-1}{n}$ elements in our state space.

Answer of exercise 3

(a) The values that do not appear in 1441 are 2 and 3, so the the states that communicate with 1441 are: 2222, 2223, 2232, 2233, 2322, 2323, 2333, 3333.

(b) The chain is not irreducible because no state communicates with 1234.

(c) The chain works fine, except for this problem with state 1234, so we need to find a way for other states to reach it and for them to be reached by it. One possibility is to always give a small possibility p for any state to access it and a uniform chance for it to access any of the other $4^4 - 1$ states. Any p is fine as long as the kernel stays a probability measure.