LaTeX Template for STAT 547C Final Project

Giorgio Sgarbi

December 9, 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, P. H., 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:

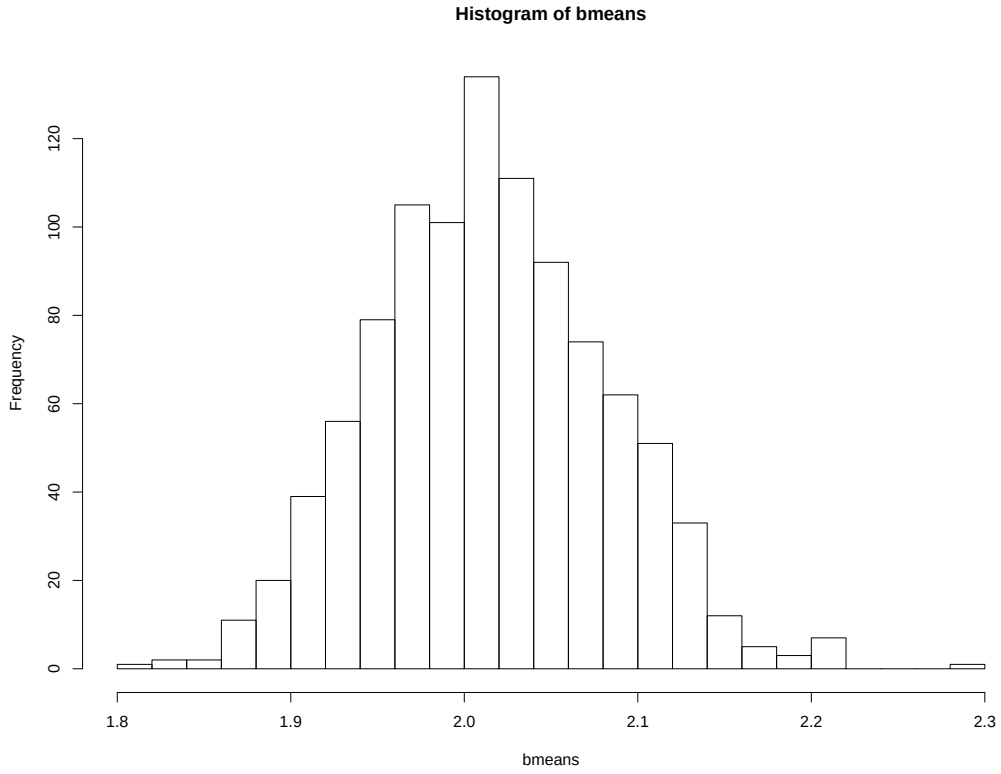
a) Given a set of samples S = $\{s_1, ..., s_n\}$, pick n samples with replacement from that set, generating a bootstrap sample $S^* = \{s_1^*, ..., s_n^*\}$.

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)

  bvar = (N-1)/N*var(bsample)
  bvars = c(bvars, bvar)
  }

hist(bmeans, breaks = 20)
hist(bvars, breaks = 20)
```



1.2 Markov Chains

The following the definitions and properties will be the base to the the second part of this project. This section definitions and theorems use professor Ben's and Gyer'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 will discuss more about it in the second part of the project.

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 non-negative 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 $P_n(x, A) > 0$. Clearly the 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

If a Markov kernel is irreducible and has an invariant measure, then the invariant measure is unique up to multiplication by positive constants.

1.3 Estimation

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

Assume we have an irreducible Markov

1.4 Inspirational paper and limitations

The paper suggested by my advisor, professor Alexandre Bouchard-Cote, will be the inspiration for the project described in the second part. Notice that its use will be limited at this point. In particular, we highlight the following:

a. Variation will be measured with simulations. See more in part 2.

b. The main idea to create will be translated for now into removing the weight from the kernel diagonals, i.e, finding kernels such that the probabilities of returning to the same state are low or 0.

2 Open questions and research directions

2.1 Research question

Our goal is to change the sample step described in 1.1 into a step that assigns probabilities of picking states that now **depend on the present sample** in a very particular way. By doing that, we *hope* to reduce the variance of our estimator $\tilde{I}$ when compared to using the traditional step discussed in 1.1. 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 of Optimum Monte-Carlo Sampling Using Markov Chains.

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.

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.

2.2 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 B (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 $S - \{x\}$, $K(x, \{s\}) = \dfrac{1}{n^n - 1}$. Naturally, we still need to assign a probability for x to return to itself and that will be 0, i.e, $K(x, x) = 0$.

2.3 Next steps

[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.

A Exercises

We use the state notation described in part 2 when not stated otherwise.

1. Exercise:

   (a) How would you define the identity matrix in the context of kernels so that it behaves as the identity in multiplication?

   (b) Show that your definition is consistent with the properties of kernels given in part 1.

2. Exercise:

   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 $K(x, \{y\})$ as: uniformly pick an accessible state y from state x.

   a) List (or draw) the states that are connected to 112.

   b) Argue that this chain is not irreducible.

   c) If you were to simulate such a chain, how can you make irreducible by not changing the kernel "too much" (I mean, try not to create a whole new kernel)?

   Exercise:

   a) Come up with a way to describe our state space different from the way described in part 2.

   b) How would you represent the elements 123 and 221 from the example in section 2.1?

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

   Answer: One possibility is $i_1...i_n$, wherei_k counts the number of times element k is picked. Notice that, in this case, $\sum i_k = n$.

3. Completely describe in Kernel notation for n = 3.

   Prove theorem in the case of finite state spaces