Sampling Algorithms for Statistical Model Checking

Presented By
Devyani Goel
CSE Department
HBTU Kanpur
Under The Guidance Of
Prof M.V Panduranga Rao

INTRODUCTION

➢ Sampling

Sampling is a process used in statistical analysis in which a predetermined number of observations are taken from a larger population. The methodology used to sample from a larger population depends on the type of analysis being performed but may include simple random sampling or systematic sampling.

➢ Heterogeneous Subset Sampling

The topic follows the concept of heterogeneous sampling in which there is a domain set with each element in the domain set having different probabilities of being included in the sample, which is a subset of the domain set.

A Hybrid algorithm is proposed for drawing the sample which is a combination of 2 basic algorithms- Naive algorithm and Sieve algorithm.

PROBLEM STATEMENT

The problem statement involves drawing several samples from the domain set of size n. Let the domain set be D={1,.....,n} in which each element i is associated with an inclusion probability p_i. For each drawn sample, which is the subset of the domain set, the probability of each element i being included is equal to p_i. We need to device an efficient mechanism to drawn sample subsets. We assume the source used to generate variates from U(0,1) standard uniform distribution is given. The HSS problem is a case of sampling without replacement.

ALGORITHMS

First, the Naive algorithm is used for solving HSS problem. The computational cost of solving the HSS problem with the Naive algorithm is O(n) time for each sample. There is no pre-processing time. Later, we propose the Hybrid algorithm, which imposes a tighter bound on the computation time, as a trade-off for pre-processing time and extra space. Our algorithm requires O(n) pre-processing time, O(n) extra space, and $O(n\sqrt{p*})$ time on average to draw each sample, where p^* is $\min(p_\mu, 1-p_\mu)$ and p_μ denotes the mean of inclusion probabilities. The Hybrid algorithm

is more efficient than the Naive algorithm and Sieve algorithm when p_μ deviates from 1/2 significantly. Here, we discuss the two algorithms used for individually solving HSS problem- **NAIVE ALGORITHM** AND **SIEVE ALGORITHM**.

➢ Naive Algorithm

To determine whether an element i should be included in a drawn sample, the algorithm compares its inclusion probability pi with a variate generated from U(0,1). Because the size of the domain set D is n, the total cost is O(n).

Algorithm 1: Naive algorithm

1. input : a domain set D = {1,...,n} in which each element i is associated with an inclusion probability pi output: S, a drawn sample
2. $S \leftarrow \varphi$
3. foreach $i \in D$ do
4. $t \leftarrow U(0,1)$
5. if $t < p_i$ then
6. $S \leftarrow S \cup \{i\}$
7. end
8. end
9. return S

The algorithm is implemented using JAVA as the basic computational language.

1. A domain set in the form of an array is taken as input from the user along with an array containing the inclusion probabilities. Both are of size n.
2. A sample set S is taken as an array and then initialized as NULL.
3. A for loop begins and runs while every element in the domain set is accessed and computed. It states for every element in domain set do the following steps-
4. Take a variable t and assign it a random value form the standard uniform set U(0,1).
5. Now, if t is less than the inclusion probability for that particular element i.e. $t<p_i$ then add i to S. The loop ends after all elements are processed.

6. Return the sample set S.

➢ Sieve Algorithm

In Sieve Algorithm instead of making a decision about each element in the domain set, we take a subset which is selected by a particular mechanism. After selecting the subset only its elements are taking into account. The mechanism is based on a simply idea. For each element i in the domain set, we decouple the decision about it being included in a drawn sample into two decisions, A_i and B_i. To guarantee the success of the decoupling, we let $Pr(A_i)$ be equal to p_{max}, the maximum of all p_i's. One advantage of the decoupling procedure is that the outcome of a decision B_i is meaningful if and only if the correspondent decision A_i is included. Another advantage is that all A_i's form a homogeneous case of the HSS problem which can be solved efficiently by two methods, namely, Procedure SWOR(k,D) and a binomial sampling from B(|D|,pmax).

Algorithm 2: Sieve algorithm

input : a domain set D = {1,...,n} in which each element i is associated with an inclusion probability p_i
output : a drawn sample S
$p_{max} \leftarrow \max i \in D\ p_i$
k ← B(|D|,pmax)
R ← SWOR(k,D)
S ← φ
foreach i ∈R do
t ← U(0,1)
if $t < p_i/p_{max}$ then
S ← S ∪{i}
End
End
return S

JAVA is used to implement the algorithm into ready to use code.

1. A domain set d of length n along with a set of inclusion probabilities is taken as input array from the user.
2. Now we find the maximum probability out of given inclusion probabilities using for loop to navigate through the probability

array. After finding p_{max}, the binomial(k) is calculated with n number of trials and p_{max} as the rate of success.

3. This k is used in the SWOR method which is then called. Value returned by the SWOR method is stored in R.
4. Now the output sample S is assigned a NULL value.
5. A for loop begins and continues till each element in R is not accessed.
6. A variable t is assigned a random value from the standard uniform distribution set U(0,1).
7. If this t is less than p_i/p_{max}, then i is included in the S sample.
8. After completion S is returned.

Procedure SWOR(k,X)

built-in: an array E of size n, where E[i] = i for all $1 \leq i \leq n$
input : an integer k, where $k \leq \|X\| \leq n$ and a set $X = \{X1,...,X\|X\|\}$
output : R, a randomly selected subset of X, whose size is k
$R \leftarrow \varphi$
for i = 1 to k do
$t \leftarrow U(0,1)$
$t \leftarrow \min(\lfloor t(n-i+1) \rfloor, n-i)$
exchange E[i + t] with E[i]
$R \leftarrow R \cup \{X_{E[i]}\}$
End
re-swap array E into the built-in state
return R

1. The SWOR method has a built in array E with each E[i]=i. Array X is taken as input along with an integer k. R is the output array which is selected from X and whose size is k. SWOR method selects at random a subset of k from X and stores it in R.
2. Initially R is initialized to NULL value.
3. A for loop begins with i from 1 to k.
4. Initialize t with a random value from the standard uniform set U(0,1).
5. Now, t is assigned the minimum value of the two values- [t*(n-i+1)] and [n-i].
6. Element at E[t+i] gets exchanged with E[i] through a series of steps.
7. Now, X[E[i]] is added to the set R.
8. For loop ends. E is swapped back into its built in state.
9. R is returned.

Naive and Sieve both algorithms through used for HSS problem, Sieve algorithm is more efficient than Naive algorithm. Both these algorithms together are used in building the hybrid algorithm which is used for solving HSS problem in a less time consuming, less space utilization and more efficient way.

THANK YOU