Statistics and Probability Letters 132 (2018) 99–106

Contents lists available at ScienceDirect

Statistics and Probability Letters

journal homepage: www.elsevier.com/locate/stapro

Generation of discrete random variables in scalable frameworks

Giacomo Aletti [1]

ADAMSS Center, Università degli Studi di Milano, 20131 Milano, Italy

ARTICLE INFO

Article history:
Received 24 December 2016
Received in revised form 10 July 2017
Accepted 8 September 2017
Available online 4 October 2017

MSC:
62D05
68A55
65C10

Keywords:
Discrete random number generation
Discrete choice model
Scalable framework
Parallelizable algorithm

ABSTRACT

In this paper, we face the problem of simulating discrete random variables with general and varying distributions in a scalable framework, where fully parallelizable operations should be preferred. The new paradigm is inspired by the context of discrete choice models. Compared to classical algorithms, we add parallelized randomness, and we leave the final simulation of the random variable to a single associative operation. We characterize the set of algorithms that work in this way, and those algorithms that may have an additive or multiplicative local noise. As a consequence, we could define a natural way to solve some popular simulation problems.

1. Introduction

The aim of this paper is to define and to characterize a new method for the generation of discrete random variables in a scalable framework. This is done by merging two apparently different fields, namely the discrete random variables generation and the discrete choice framework.

The generation of discrete random variables may be made in different ways (see Devroye, 1986; Marsaglia et al., 2004; Rubinstein and Kroese, 2016; Shmerling, 2013 and the references therein). The most popular idea is to invert the cumulative function F_k defined on the sets of the indexes $k = 1, \dots, n$ of the support. When the cumulative function is not parametrized, we recall that a bisection search takes $O(\log_2(n))$ comparison to invert F (see Devroye 1986, Section III.2.4), once we have computed and stored the table $\{(k, F_k), k = 1, \dots, n\}$ properly. A traditional linear search may be done with $O(n)$ comparison.

There are more sophisticated ways to invert a discrete distribution. They typically require other precomputations and bookkeepings. We recall here three fast popular methods. Even if these three methods might be dated, the most recent books do not provide other paradigms that generate directly discrete random variables with general distributions (cfr., for example, Rubinstein and Kroese, 2016).

The *table look-up method* is the fastest way to simulate a large number of i.i.d. discrete random variables (see Devroye (1986, Section III.3) and Marsaglia et al., 2004), when all the probabilities $\{p_j = \frac{n_j}{m}, j = 1, \dots, n\}$ are rational numbers with the same denominator m. Obviously, $n_1+n_2+\cdots+n_N = m$, and hence we may set up a table $\{(k, Q_k), k = 1, \dots, m\}$ such that

E-mail address: giacomo.aletti@unimi.it.

[1] Member of "Gruppo Nazionale per il Calcolo Scientifico (GNCS)" of the Italian Institute "Istituto Nazionale di Alta Matematica (INdAM)". This work was partially developed during a visiting research period at the Volgenau School of Engineering, George Mason University, Fairfax (VA), United States. It was revised during a visiting research period at the School of Mathematical Science, Fudan University, Shanghai, China. The author thanks for the hospitality.

https://doi.org/10.1016/j.spl.2017.09.004
0167-7152/

n_j distinct values of Q's are set to j, for each $j = 1, \dots, n$. To simulate the required random variable, take an integer uniform discrete random variable Y on $\{1, \dots, m\}$ and then compute Q_Y in $O(1)$ time. The drawback of this method compared to the previous ones is the amount of required space, that is necessary to store the exact pseudo-inverse function $Q_j = F^{-1}(\frac{j}{m})$ on the equispaced nodes $\{\frac{1}{m}, \dots, 1\}$.

The *method of guide tables* was introduced by Chen (1974), and stores a second "guiding table" $\{(k, G_k), k = 1, \dots, m\}$ that helps the generation, by reducing the expected number of comparison to less than $1 + \frac{n}{m}$, see (Devroye, 1986, Section III.3.4).

The last method is called *alias table* (see Devroye, 1986, Section III.4). It was firstly introduced in Walker (1974, 1977), and then it was improved together with a simple probabilistic proof in Kronmal and Peterson (1979). This method does not need the computation of the cumulative function even if it requires the probabilities of the events to be normalized. Besides this, it uses a special table build in $O(n)$ time (called alias table). During the simulation process, it uses only 2 comparisons.

In all these methods, the simulation of a random variable is made by constructing a table based on the *probabilities* of the possible events. In many applied situations, the probabilities are computed up to a multiplicative constant, or in logarithm scale up to the translational constant. Accordingly, a preprocess must be done to reconstruct the normalized probabilities before using any of the above methods. Moreover, if one has to simulate random variables with different distributions, it must be allocated a different table for each of them. Finally, if the probability of one of the events changes (or if the support itself changes), one must restart all the process.

The novelty of this paper is the introduction of a new method for the generation of several discrete independent random variables with possible different distributions, and whose distributions does not belong to a parametric family. A key point is the fact that this method does not precompute a table based on the probability of the events. Instead, it is based on the following two assumptions:

- first, for each random variable, and for each point of the support, we can perform an action (called utility) that involves the sole local accessible information. This operation is hence fully parallelizable, and we do not need to take care of scaling constants when the probabilities are given in logarithm space and/or up to a constant;
- secondly, for each random variable, a single associative operation on the utilities on the points of its support must finally simulate the discrete random variable.

This new method may also be updated in a fast natural way when the probabilities are changing with time (one local update and an associative operation), and hence it may also be used in real time problems. The main result of this paper is the characterization of all the possible ways of simulating discrete random variables with such assumptions.

Obviously, there is a counterpart. On the one hand, in fact, we do not provide a precomputed table and we perform only one associative operation on some locally calculated quantities. On the other hand, each of these quantities depends on an independent source of uncertainty. Summing up, we increment the total amount of randomness (by adding a local source), and hence we could reduce the non-local operations to an associative one.

The idea behind this new method may found a counterpart in the framework of discrete choice models, where the point of view is to understand the behavioral process that leads to an agent's choice among a set of possible actions (see Train, 2009) for a recent book on this subject). The researcher knows the set of the possible actions, and by observing some factors, he may infer something about the agent's preferences. At the same time, he cannot observe other random factors, linked to each possible action, that cause the final decision. If the researcher could have observed these hidden factors, he could have predicted the action chosen by the agent by selecting the one with maximum utility function.

The process of choice selection has the two characteristics we gave above for random generation:

- for each agent (random variable), for each action (point of the support), the utility function – a given deterministic function of the observed and the hidden factors – is calculated and depends only on local variables;
- for each agent, the final choice is made by selecting the action with maximum utility value (associative operation).

In other words, we are changing the usual point of view belonging to discrete choice framework to produce and characterize new scalable simulators for discrete random variables, based on primary functions given, e.g., in a general SQL database. As a by product, we will be able to characterize all the choice models with independent and identically distributed hidden factors and such that the probability of choosing an action is (proportional to) a given function of the observable factors.

The content of the paper is structured as follows. In Example 1 of Section 2, we introduce a very general problem of randomization in classification procedures in a SQL environment, and we solve it with our new paradigm. The subsequent Example 2 shows the mathematical position of the same problem in discrete choice's framework. The reason why the two examples share the same problem is discussed at the end of the examples. The section ends with a discussion on the fast updating process that is required when the distributions vary with time.

In Section 3 we give the main results of the paper, based on the notion of max-compatible family of distributions, which is the mathematical structure at the base of our new method. This family is fully characterized in terms of the cumulative functions in Theorem 3.1, that can be seen as the main mathematical result of this paper. The section continues with the description of the new algorithm of random variable generation in terms of max-compatible families, and it ends with the characterization of some natural models that may be found in usual applied situations.

Section 4 concludes the paper with some future research regarding this topic.