Methods for Low and High Frequencies in the Frequency-severity Model.

November 6, 2018

1 Case of low frequencies.

1.1 Statement of the problem.

Suppose that we want to model $X_1 + \ldots + X_n$, where n is a random number drawn from a frequency distribution F (typically, a Poisson, a Binomial, or a Negative Binomial distribution), X_i is a random number drawn from a severity distribution S. Note that if $n = 0$, then we assume that the sum is also 0. Denote this model by $F * S$. Suppose that we are calculating a discretization of a sample of this model by some Monte Carlo method.

Assume that F has a very low mean, e.g., 0.0001. In this case, in average, we will need to generate 9999 of zeros, in order to get a single non-zero element. First, it is obviously a non-optimal way of using the resources, and, second, an estimate of the mean will be inadequate if S has a large mean (like 1000000). The estimate of the mean will be especially inadequate if we generate only 5000 of elements probably getting 5000 of zeros and concluding that the mean is also zero.

1.2 Description of the algorithm.

Denote by p_0 the probability of getting 0 for frequency distribution F. In case of $p_0 > 0.9$ it makes sense to apply the algorithm described below.

The idea is to apply to F simple stratified sampling with 2 strata: the first for 0 with probability p_0 and the second for being greater than 0 with probability $1 - p_0$ modelled by the conditional distribution $\bar{F}$ of F being greater than 0. Next, we consider only the case of second strata, i.e. $\bar{F}$ (which can also be calculated by stratified samlping). We calculate the model for this case using some Monte Carlo method and obtain the sample of size m for this model. Now we enrich in our mind this sample by adding $\frac{mp_0}{1-p_0}$ zeros. Adding it in reality would have been a waste of resources. Assume that we need to obtain a discretization of size s as an output, then we return the discretization of the zero-enriched sample of size s. Most part of this discretization will be zeros, since the non-zero elements would affect only the right tail of the distribution.

This procedure would allow to safe memory by not generating extra zeros and to obtain better precision in estimating the right tail.

1.3 Expectation of benefits.

I expect that this algorithm would work only slightly slower than a non-modified Monte Carlo method, but the precision would be much better.

2 Case of high frequencies.

2.1 Statement of the problem.

Similarly with the previous section, we consider a frequency-severity model $F*S$. Again the goal is to calculate discretization of a sample of this model by some Monte Carlo method.

Assume that F has a very high mean, e.g., 10000. In this case, we will need to generate a lot of random elements from X for a single Monte Carlo path. It will take a lot of time. Maybe we do not need to generate that many elements in reality.

2.2 Description of the algorithm.

If F is Poisson, then a well-known frequency reduction formula holds:

$$X * \mathrm{Pois}(\lambda) = \left(X * \mathrm{Pois}\left(\frac{\lambda}{m}\right)\right)^{(m)}, \tag{1}$$

where (m) means an m-th convolutional power. Thus, formula (1) means that we may decrease the frequency mean m-times, but for this we need to pay by calculating $\log_2 m$ convolutions ($\log_2 m$, but not m, since we can calculate them iteratively). There exist well-known analogs of formula (1) for other Panjer-like distributions (i.e. the Binomial and the Negative Binomial cases).

The idea is to decrease λ a sufficient number of times, to model $X * \mathrm{Pois}\left(\frac{\lambda}{m}\right)$, by a Monte Carlo method, to obtain its discretization, and then to sum up the results using bootstrapping. In order to get good results, we need to bootstrap a sufficiently large number of times that should be determined experimentally.

2.3 Expectation of benefits.

I expect that this algorithm would have a precision approximately equal to the precision of a non-modified Monte Carlo, but it will be much faster.

2.4 Description of alternative approaches.

Suppose that we have generated a random number n from frequency distribution F. Assuming that n is large, we can generate not n random elements from X, but $\frac{n}{C}$ elements, where C is some constant. The other elements could be obtained by bootstrapping.

Another alternative could be to generate C random elements from X first, before the loop, and then to get them from this static array by bootstrapping.

2.5 Expectation of benefits.

One of the main benefits of this algorithm is that it does not depend on the type of frequency distribution. I expect that the algorithm would work quite fast, that there will be problems with precision, but probably it will be possible to overcome this difficulties by the right choice of C. However it will not be trivial.