1 Insurance Risk Models

1.1 Classic model

Starting with initial reserve $x > 0$, an insurance risk business earns money at constant rate $c > 0$ per unit time (from interest (say)). Meanwhile, according to a point process with times $\{t_n : n \geq 1\}$, $0 < t_1 < t_2 < \cdots$, and counting process $\{N(t)\}$, *claims* against the business occur in the amounts $B_n > 0$ causing the reserve to jump downward by such amounts. If at any such jump, the reserve falls ≤ 0, then the business is said to be *ruined*. To model this in discrete time (since after all, ruin can only occur at a claim time), we first define the *unrestricted* reserve process by

$$X_n(x) = x + ct_n - \sum_{j=1}^{n} B_j,\ n \geq 1, \quad X_0(x) = x. \tag{1}$$

$X_n(x)$ denotes the reserve level right after the n^{th} claim has come in, and it is allowed to fall ≤ 0 and continue.

$X_n(x)$ denotes the reserve level right after the n^{th} claim has come in, and it is allowed to fall ≤ 0 and continue.

In continuous time units, the time of ruin is defined by $\tau(x) \stackrel{\text{def}}{=} \min\{t_n : X_n(x) \leq 0\}$, where by convention $\tau(x) \stackrel{\text{def}}{=} \infty$ if ruin never occurs, that is, if $X_n(x) > 0,\ n \geq 0$. Of major interest is to compute $P(\tau(x) < \infty)$, the probability that ruin will ever occur, or $P(\tau(x) \leq T)$, the probability that ruin will occur by time T.

One can also define ruin in discrete-time units $\tau^{(d)}(x) \stackrel{\text{def}}{=} \min\{n : X_n(x) \leq 0\}$, representing the index of the claim (if any) that caused ruin, and then if ruin does occur, the magnitude of ruin, $M = |X_{\tau^{(d)}(x)}(x)|$ might be of interest; it is the amount of money that the company is ruined by.

We define the *risk process* itself (in discrete time) as

$$R_n(x) = \begin{cases} X_n(x), & \text{if } \tau(x) > n; \\ 0, & \text{if } \tau(x) \leq n. \end{cases}$$

It is simply the X process until ruin at which time it remains at value 0 forever after.

Letting $A_n = t_n - t_{n-1},\ n \geq 1$ denote interaarrival times, (1) can be re-written as a random walk, with increments $cA_j - B_j$ via

$$X_n(x) = x + \sum_{j=1}^{n} (cA_j - B_j),\ n \geq 1, \quad X_0(x) = x. \tag{2}$$

Duality between risk models and queueing models

When the point process forms a renewal process, and independently the claim sizes are iid, then this model shares an elegant duality with a FIFO GI/GI/1 queue: using $T_n = cA_{n+1}$ as customer interrarrival times, and the B_n as service times, it holds that $P(\tau(x) \leq T) = P(D_T(0) \geq x)$, where $D_n(0)$ denotes the n^{th} customer's delay (using the delay recursion $D_{n+1} = (D_n + B_n - T_n)^+$) if initially $D_0 = 0$. More generally, $P(R_n(x) \leq y) = P(D_n(y) \geq x)$, where $D_n(y)$ denotes the n^{th} customer's delay using the delay recursion if initially $D_0 = y$. For example, consider $n = 1$. Then $R_1(x) \leq y$ is equivalent to $x + cT_1 - B_1 \leq y$ which can be re-written as

$y + B_1 - cT_1 \geq x$. On the other hand, the event $\{D_1(y) \geq x\}$ has the same distribution as the event $\{(y + B_1 - cT_1)^+ \geq x\}$ which is equivalent to the event $\{y + B_1 - cT_1 \geq x\}$; we conclude that indeed $P(R_1(x) \leq y) = P(D_1(y) \geq x)$.

1.2 Simple simulation of the classic model risk process

In the following we show how to simulate for computing $P(\tau(x) \leq T)$, when the arrivals are Poisson at rate λ, and the claim sizes are iid with general distribution G. It is assumed that one already has an efficient algorithm for generating $B \sim G$. The code can easily be modified to accommodate a general interarrival time distribution, or a non-stationary Poisson process of arrivals.

The following algorithm generates one copy of the indicator $I\{\tau(x) \leq T\}$.

t : simulated time,
$R = R(x)$: the reserve level at that time given it started initially at x,
$\tau = \tau(x)$: the (continuous) time of ruin,
I : the indicator for the event {ruin by time T} $= \{\tau(x) \leq T\}$, and
M : the magnitude of ruin given that it occurred.

1. $t = 0$, $R = x$, $\tau = \infty$, $I = 0$, $M = 0$.
2. Generate U
3. $t = t + (-(1/\lambda)\ln(U))$. If $t > T$ stop. $R = R + c(-(1/\lambda)\ln(U))$
4. Generate $B \sim G$. Set $R = R - B$. If $R \leq 0$, then set $I = 1$, set $\tau = t$, set $M = |R|$ and stop.
5. Go to 2.

To estimate $P(\tau(x) \leq T)$, one would need to run this algorithm n (large) times to obtain iid copies $I_1, \ldots, I_n$ and use the estimate

$$P(\tau(x) \leq T) \approx \frac{1}{n}\sum_{j=1}^{n} I_j.$$

To estimate $E(M)$, one would run the simulation as many times as it takes until there are a total of n (large) ruins, so as to get n iid copies of M; $M_1, \ldots, M_n$ and then use the estimate

$$E(M) \approx \frac{1}{n}\sum_{j=1}^{n} M_j.$$

Estimation of ruin probabilitites (or equivalently $P(D > x)$ for a queue) becomes more difficult for a large x since then the probability can be quite small, and one would have to do a very long simulation to detect a ruin. Fortunately, there are a variety of more efficient methods that have been developed for simulating for estimating the probability of ruin; key words are “rare event simulation”, “importance sampling”, “change of measure”, and “fast simulation” methods. It is an active area of research. The methods used depend on the distribution of B; for example if it is light-tailed or heavy-tailed. The basic idea is to change the distributions of interarrival and claim size so that ruin occurs more often, simulate that instead, and then transform the answer back to the original one.

1.3 Markovian model

Here we consider another risk model which makes more sense when (say) considering car insurance. Policy holders join according to a Poisson process at rate v. Independently each such holder remains joined for an amount of time that is exponentially distributed with rate μ and then quits, and then is no longer a policy holder. Independently, each holder sends in claims according to a Poisson process at rate λ. All claim sizes are iid with distribution G. Also, each holder pays a premium to the company of $\$c$ per unit time.

Some observations: At any given time t, suppose there are N policy holders. Starting from time t, let X_1 denote the time until the next new holder joins, X_2 the time until one of the N holders quits, and X_3 the time until the next claim comes in. Then, by the Poisson/exponential assumptions and the memoryless property of the exponential distribution, these three random variables are independent and exponentially distributed; $X_1 \sim exp(v)$, $X_2 \sim exp(N\mu)$, $X_3 \sim exp(N\lambda)$. X_1 is the time until the next event from the Poisson rate v process, that is clear. X_2: There are N holders, each independently with their own "join time" $Y_1, \ldots, Y_N$ exponential at rate μ. Thus $X_2 = \min\{Y_1, \ldots, Y_N\}$, which is exponential at rate $N\mu$ because $P(X_2 > x) = P(Y_1 > x, \ldots, Y_N > x) = P(Y > x)^N = e^{-N\mu x}$. Similarly, $X_3 \sim exp(N\lambda)$ since it is the minimum of N iid $exp(\lambda)$ rvs. [1] We now proceed to use this fact.

For purposes of a discrete-event simulation, there are 3 "event types" in this model; e_1: a new policy holder joins, e_2: an existing policy holder quits, and e_3: a claim comes in. The time until the next event is the minimum $m = \min\{X_1, X_2, X_3\}$, hence is itself exponentially distributed with rate $r = v+N\mu+N\lambda$ = the sum of the three rates. Thus if the current time is t and there are N policy holders, then we can schedule the "next event" at time $t+(-(1/r)\ln(U)$. But we must determine what kind of event it is. Noting that each of X_1, X_2 or X_3 will be the minimum with probabilities $p_1 = v/r$, $p_2 = N\mu/r$, $p_3 = N\lambda/r$, we conclude that we simply need to independently generate a rv C with distribution $P(C = 1) = p_1$, $P(C = 2) = p_2$, $P(C = 3) = p_3$; then if $C = i$ we conclude that the event is of type e_i. Generate U. If $U \leq p_1$, then set $C = 1$. If $p_1 < U \leq p_1 + p_2$, then set $C = 2$. If $U > p_+ p_2$, then set $C = 3$.

Here then is the code for simulating this model up to time T.

1. $t = 0$, $R = x$, $N = n_0 > 0$, $\tau = \infty$, $I = 0$, $M = 0$.
2. Generate U. Set $r = v + N\lambda + N\mu$, $p_1 = v/r$, $p_2 = N\mu/r$, $p_3 = N\lambda/r$.
3. $t = t + (-(1/r)\ln(U))$. If $t > T$ stop.
4. $R = R + cN(-(1/r)\ln(U))$.
5. Generate C ($P(C = 1) = p_1$, $P(C = 2) = p_2$, $P(C = 3) = p_3$.)
6. If $C = 1$, then $N = N + 1$. If $C = 2$, then $N = N - 1$.
7. If $C = 3$, then: generate $B \sim G$. Set $R = R - B$. If $R \leq 0$, then set $I = 1$, set $\tau = t$, set $M = |R|$ and stop.
8. Goto 2.

[1] In general if $X_i \sim exp(\lambda_i)$ are independent, $i = 1, 2, \ldots, m$, then $m \stackrel{\text{def}}{=} \min\{X_1, X_2, \ldots, X_m\}$ is exponential at rate $\lambda = \lambda_1 + \cdots + \lambda_m$, the sum of the individual rates.