CDF of Sample Quantile

Prepared by: Dr. Bentley Coffey, The Cadmus Group

2.17.11

We are interested in computing the probability that a sample quantile, q, exceeds a threshold, τ:[1]

$$Pr\left(q = (1-\gamma)X_{[j]} + \gamma X_{[j+1]} \geq \tau\right)$$

Where $X_{[j]}$ is the j^{th} order statistic of an iid sample drawn of size n from a parent distribution, $X_{[j+1]} \geq X_{[j]}$, and the CDF of X is given by Φ and pdf by ϕ (so normal here but this should work for any continuous parent distribution). By default, R computes the sample quantile using the following values for $\gamma \in [0,1)$ and $j \in \{0,\ldots,n-1\}$:

$$\gamma = (n-1)p - j$$
$$j = \lfloor (n-1)p \rfloor$$

Where $\lfloor\,\rfloor$ is the floor operator (i.e. round down to the nearest integer) and $p = \Phi(q) > 0$. We can accomplish this in R via brute force Monte Carlo:

Prob.Sample.Quantile.Exceeds.Threshold <-
mean(replicate(Num.Simulations,quantile(rnorm(N,mean,sd),Percent)>=Threshold))

But, unfortunately, that's far too slow for the application. Rather than performing this N-dimensional integration with simulation, we can do some math to simplify the problem and speed up its solution. The joint pdf of $X_{[j]}$ and $X_{[j+1]}$ is given by:

$$pdf\left(X_{[j]}, X_{[j+1]}\right) = \left[\frac{n!}{(j-1)!\,\left(n-(j+1)\right)!}\right]\Phi\left(X_{[j]}\right)^{j-1}\phi\left(X_{[j]}\right)\left[1-\Phi\left(X_{[j+1]}\right)\right]^{n-(j+1)}\phi\left(X_{[j+1]}\right)$$

The marginal pdf of $X_{[j+1]}$ is given by:

$$pdf\left(X_{[j+1]}\right) = \left[\frac{n!}{\left((j+1)-1\right)!\,\left(n-(j+1)\right)!}\right]\Phi\left(X_{[j+1]}\right)^{j}\left[1-\Phi\left(X_{[j+1]}\right)\right]^{n-(j+1)}\phi\left(X_{[j+1]}\right)$$

[1] The quantity gets computed MANY times inside of a larger program and it is imperative that the computation of this is fast. The larger program performs an optimization search with a numerically integration of an interpolation of many points generated from this routine at each step in the optimization. The ultimate application is to find n so as to maximize a net benefit objective that is linear in costs and benefits are linear in the confidence of a quality acceptance sampling scheme (i.e. benefits accrue if the quantile of parent distribution with probability p exceeds the threshold AND the quantile of the size n sample also exceeds the threshold. This is repeated for several different parent distributions…

Using Bayes' identity defining the condition pdf of $X_{[j]}|X_{[j+1]}$:

$$pdf\left(X_{[j]}|X_{[j+1]}\right) = \frac{pdf\left(X_{[j]}, X_{[j+1]}\right)}{pdf\left(X_{[j+1]}\right)} = j\left[\frac{\Phi\left(X_{[j]}\right)}{\Phi\left(X_{[j+1]}\right)}\right]^{j-1} \frac{\phi\left(X_{[j+1]}\right)}{\Phi\left(X_{[j+1]}\right)} 1\left\{X_{[j]} \leq X_{[j+1]}\right\}$$

We can solve for the smallest $X_{[j]}$ as a function of $X_{[j+1]}$, call it $X_{[j]}^{*}(X_{[j+1]})$ such that q exceeds the threshold (using definition above):

$$X_{[j+1]} \geq X_{[j]}^{*}(X_{[j+1]}) = \frac{\tau - \gamma X_{[j+1]}}{(1-\gamma)}$$

Integrating the conditional probability over the region where q exceeds τ:

$$Pr\left(X_{[j]} \geq X_{[j]}^{*}(X_{[j+1]})\middle|X_{[j+1]}\right) = \left[1 - \left[\frac{\Phi\left(X_{[j]}^{*}(X_{[j+1]})\right)}{\Phi\left(X_{[j+1]}\right)}\right]^{j}\right] 1\left\{X_{[j+1]} \geq \tau\right\}$$

Our quantity of interest can be found by multiplying the conditional probability by the marginal pdf and integrating:

$$Pr(q \geq \tau) = \int_{X_{[j+1]}} Pr\left(X_{[j]} \geq X_{[j]}^{*}(X_{[j+1]})\middle|X_{[j+1]}\right) pdf\left(X_{[j+1]}\right)$$

Which we can write in terms of expectations using the definition of the expectation operator:

$$Pr(q \geq \tau) = E_{X_{[j+1]}}\left(Pr\left(X_{[j]} \geq X_{[j]}^{*}(X_{[j+1]})\middle|X_{[j+1]}\right)\right)$$

Working with expectation conditional on X[j+1] $\geq \tau$ is actually more computationally efficient because the integrand is only evaluated where we know it to be non-zero

$$Pr(q \geq \tau) = E_{X_{[j+1]}|X_{[j+1]} \geq \tau}\left(Pr\left(X_{[j]} \geq X_{[j]}^{*}(X_{[j+1]})|X_{[j+1]} \geq \tau\right)\right) \Pr\left(X_{[j+1]} \geq \tau\right)$$

Note that, when $\gamma = 0$, this reduces to a "closed-form" probability = $\Pr(X_{[j+1]} \geq \tau)$

Each of these pieces should be easily computable in R:

1. Probability X[j+1] ≥ τ:

$$\Pr\left(X_{[j+1]} \geq \tau\right) = 1 - CDF_{X_{[j+1]}}(\tau) = 1 - CDF_{beta}(\Phi(\tau); (j+1)+1, n-(j+1)+1)$$
$$\Pr\left(X_{[j+1]} \geq \tau\right) = CDF_{binom}(j, n, \Phi(\tau))$$

Simply:

Prob.Order.Stat.jPlus1.Exceeds.Threshold <- pbinom(j,n,pnorm(Threshold)

2. Integral of conditional probability:

$$E_{X_{[j+1]}|X_{[j+1]}\geq\tau}\left(Pr\left(X_{[j]} \geq \tau | X_{[j+1]} \geq \tau\right)\right) = \int_{\tau}^{\infty} \left[1 - \left[\frac{\Phi\left(X^*_{[j]}(X_{[j+1]})\right)}{\Phi\left(X_{[j+1]}\right)}\right]^j\right] pdf\left(X_{[j+1]}\right) d\, X_{[j+1]}$$

Because $X_{[j]} = \Phi^{-1}(U_{[j]})$, where $U_{[j]}$ is the j^{th} order statistic of the uniform distribution, we can perform a convenient change of variables here:

$$E_{X_{[j+1]}|X_{[j+1]}\geq\tau}(\cdot) = \int_{\Phi(\tau)}^{\Phi(\infty)} \left[1 - \left[\frac{\Phi\left(X^*_{[j]}\left(\Phi^{-1}\left(U_{[j+1]}\right)\right)\right)}{\Phi\left(\left(\Phi^{-1}\left(U_{[j+1]}\right)\right)\right)}\right]^j\right] pdf\left(U_{[j+1]}\right) d\, U_{[j+1]}$$

Because U[j] is distributed as beta(j,N-j+1), we can perform another convenient change of variables:

$$E_{X_{[j+1]}|X_{[j+1]}\geq\tau}(\cdot) = \int_{\Phi(\tau)}^{\Phi(\infty)} \left[1 - \left[\frac{\Phi\left(X^*_{[j]}\left(\Phi^{-1}\left(CDF^{-1}_{beta}(U; j+1, N-(j+1)-1)\right)\right)\right)}{\Phi\left(\left(\Phi^{-1}\left(CDF^{-1}_{beta}(U; j+1, N-(j+1)-1)\right)\right)\right)}\right]^j\right] d\, U$$

Just a few lines of rocket-fast code:

Uniform.Random.Draws <- runif(Num.Simulations,pnorm(Threshold),1)
Order.Stat.jPlus1.Draws <- qnorm(qbeta(Uniform.Random.Draws,j+1,n-(j+1)+1))
Order.Stat.j.Min <- (Threshold – Gamma.Weight Order.Stat.jPlus1.Draws)/(1-Gamma.Weight)*
Prob.Sample.Quantile.Exceeds.Threshold.Given.Order.Stat.jPlus1.Exceeds.Threshold <-
mean(1 – (pnorm(Order.Stat.j.Min) /pnorm(Order.Stat.jPlus1.Draws))^j)
*Prob.Sample.Quantile.Exceeds.Threshold <- Prob.Order.Stat.jPlus1.Exceeds.Threshold **
Prob.Sample.Quantile.Exceeds.Threshold.Given.Order.Stat.jPlus1.Exceeds.Threshold