Fast evaluation of the inverse Poisson CDF

Mike Giles

University of Oxford
Mathematical Institute

Ninth IMACS Seminar on Monte Carlo Methods

July 16, 2013

Outline

- problem specification
- incomplete Gamma function
- CPUs versus GPUs
- asymptotic Normal approximation
- asymptotic Temme approximation
- Temme asymptotic evaluation
- putting it all together

Poisson CDF and inverse

The CDF for Poisson rate λ is

$$\overline{C}(n) \;\equiv\; \mathbb{P}(N \leq n) \;=\; e^{-\lambda} \sum_{m=0}^{n} \frac{\lambda^m}{m!}.$$

The inverse CDF is defined as $\overline{C}^{-1}(u) = n$ where n is the smallest integer such that

$$u \;\leq\; e^{-\lambda} \sum_{m=0}^{n} \frac{\lambda^m}{m!} \qquad \text{(bottom-up)}$$

or

$$1-u \;\geq\; e^{-\lambda} \sum_{m=n+1}^{\infty} \frac{\lambda^m}{m!} \qquad \text{(top-down)}$$

Poisson CDF and inverse

When λ is fixed and not too large ($\lambda < 10^4$?) can pre-compute $\overline{C}(n)$ and perform a table lookup.

When λ is variable but small ($\lambda < 10$?) can use bottom-up/top-down summation.

When λ is variable and large, then rejection methods can be used to generate Poisson r.v.'s, but the inverse CDF is sometimes helpful:

- stratified sampling
- Latin hypercube
- QMC

This is the problem I am concerned with — approximating $\overline{C}^{-1}(u)$ at a cost similar to the inverse Normal CDF, or inverse error function.

Incomplete Gamma function

If X is a positive random variable with CDF

$$C(x) \;\equiv\; \mathbb{P}(X < x) \;=\; \frac{1}{\Gamma(x)} \int_\lambda^\infty e^{-t}\, t^{x-1}\, \mathrm{d}t.$$

then integration by parts gives

$$\mathbb{P}(\lfloor X \rfloor \leq n) \;=\; \frac{1}{n!} \int_\lambda^\infty e^{-t}\, t^n\, \mathrm{d}t \;=\; e^{-\lambda} \sum_{m=0}^{n} \frac{\lambda^m}{m!}$$

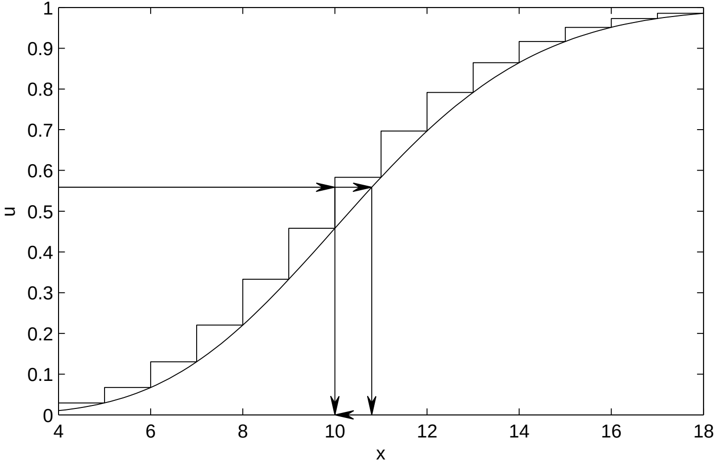
$$\Longrightarrow \quad \overline{C}^{-1}(u) \;=\; \lfloor C^{-1}(u) \rfloor$$

We will approximate $Q(u) \equiv C^{-1}(u)$ so that $|\widetilde{Q}(u) - Q(u)| \;<\; \delta \;\ll\; 1$

This will round down correctly except when $Q(u)$ is within δ of an integer – then we need to check some $\overline{C}(m)$

Incomplete Gamma function

Illustration of the rounding down of $Q(u) \equiv C^{-1}(u)$ to give $\overline{C}^{-1}(u)$



CPUs and GPUs

On a CPU, if the costs of $\widetilde{Q}(u)$ and $\overline{C}(m)$ are C_Q and C_C, the average cost is approximately

$$C_Q + 2\,\delta\, C_C.$$

However, on a GPU with a vector length of 32, the C_C penalty is incurred if any element needs it, so the average cost is

$$C_Q + \left(1 - (1-2\,\delta)^{32}\right)\, C_C \;\approx\; C_Q + 64\,\delta\, C_C \quad \text{if } \delta \ll 1.$$

This pushes us to more accurate approximations for GPUs.

Normal approximation

It's well known that

$$C(x) \;\approx\; \Phi\left(\frac{x-\lambda}{\sqrt{\lambda}}\right)$$

which motivates the following change of variables

$$x = \lambda + \sqrt{\lambda}\, y, \quad t = \lambda + \sqrt{\lambda}\, (y-z)$$

giving

$$C(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{y} I(y,z)\; \mathrm{d}z$$

where

$$\log I = \tfrac{1}{2}\log(2\pi) - \log\Gamma(x) - t + (x-1)\log t - \tfrac{1}{2}\log\lambda$$

Normal approximation

An asymptotic expansion in powers of $\varepsilon \equiv \lambda^{-1/2}$ yields

$$I(y,z) = \exp(-\tfrac{1}{2}z^2) \left(1 + \sum_{n=1}^{\infty} \varepsilon^n p_n(y,z)\right)$$

where $p_n(y,z)$ are polynomial in y and z. Integrating by parts gives

$$C(x) \approx \Phi(y) + \phi(y) \Big(\varepsilon\,(-\tfrac{1}{3} - \tfrac{1}{6}\,y^2) + \varepsilon^2(\tfrac{1}{12}y + \tfrac{1}{72}\,y^3 - \tfrac{1}{72}\,y^5) \\ + \varepsilon^3(-\tfrac{1}{540} - \tfrac{23}{540}\,y^2 + \tfrac{7}{2160}\,y^4 + \tfrac{5}{648}\,y^6 - \tfrac{1}{1296}\,y^8)\Big)$$

and inverting this gives the asymptotic expansion

$$Q(u) = \lambda + \sqrt{\lambda}\,w + (\tfrac{1}{3} + \tfrac{1}{6}\,w^2) + \lambda^{-1/2}(-\tfrac{1}{36}\,w - \tfrac{1}{72}\,w^3) \\ + \lambda^{-1}(-\tfrac{8}{405} + \tfrac{7}{810}\,w^2 + \tfrac{1}{270}w^4) + O(\lambda^{-3/2})$$

where $w = \Phi^{-1}(u)$.

Normal approximation

All asymptotic expansions were performed using MATLAB's Symbolic Toolbox.

This gives three approximations:

$$\begin{aligned}
\widetilde{Q}_{N1}(u) &= \lambda + \sqrt{\lambda}\, w \;+\; (\tfrac{1}{3} + \tfrac{1}{6}\, w^2) \\
\widetilde{Q}_{N2}(u) &= \widetilde{Q}_{N1}(u) \;+\; \lambda^{-1/2}(-\tfrac{1}{36}\, w - \tfrac{1}{72}\, w^3) \\
\widetilde{Q}_{N3}(u) &= \widetilde{Q}_{N2}(u) \;+\; \lambda^{-1}(-\tfrac{8}{405} + \tfrac{7}{810}\, w^2 + \tfrac{1}{270}\, w^4)
\end{aligned}$$

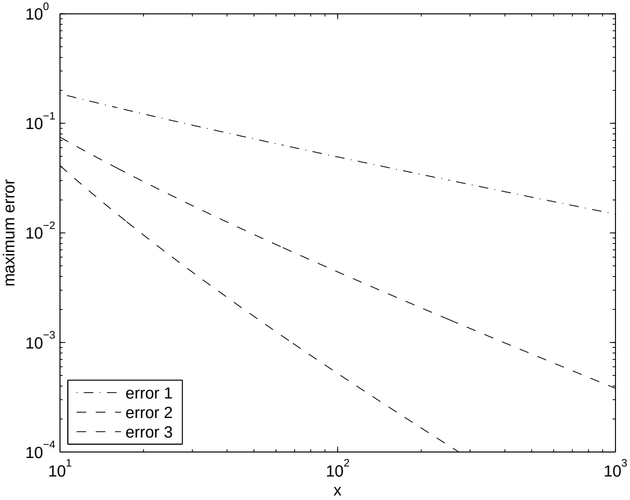
and suggests an error bound for $\widetilde{Q}_{N2}$:

$$\delta = \lambda^{-1}(\tfrac{1}{40} + \tfrac{1}{80}\, w^2 + \tfrac{1}{160}\, w^4)$$

with $\mathbb{E}[\delta] = \frac{9}{160}\, \lambda^{-1}$.

Normal approximation

Maximum error over range $|w| \leq 3$:



Temme approximation

The Normal approximation is not good when w is large, and also the asymptotic convergence is poor in powers of $\lambda^{-1/2}$.

Temme (1979) derived a uniformly convergent asymptotic expansion for $C(x)$ of the form

$$C(x) = \Phi\left(\lambda^{\frac{1}{2}} f(r)\right) + \lambda^{-\frac{1}{2}}\, \phi\left(\lambda^{\frac{1}{2}} f(r)\right) \sum_{n=0}^{\infty} \lambda^{-n}\, a_n(r)$$

where $r = x/\lambda$ and

$$f(r) \equiv \sqrt{2\ (1 - r + r \log r)},$$

with the sign of the square root matching the sign of $r-1$.

Temme approximation

To leading order, the quantile function is

$$Q(u) \approx \lambda\, r + c_0(r)$$

where

$$r = f^{-1}(w/\sqrt{\lambda}), \qquad w = \Phi^{-1}(u)$$

and

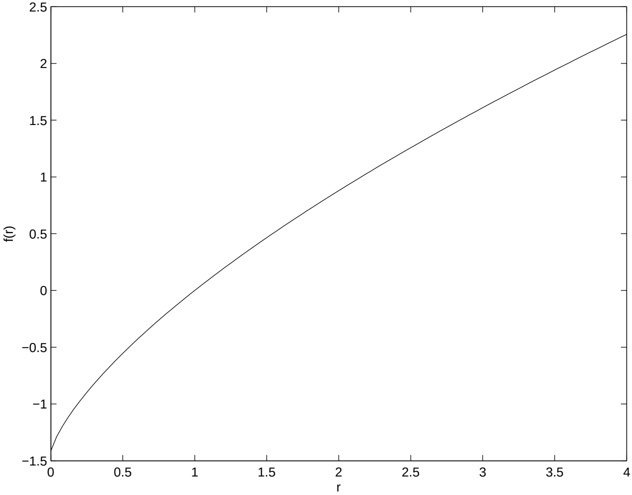
$$c_0(r) = \frac{\log\left(f(r)\sqrt{r}/(r-1) \right)}{\log r}$$

The key is that both $f^{-1}(s)$ and $c_0(r)$ can be approximated very accurately by polynomials, and an additional *ad hoc* correction gives

$$\widetilde{Q}_{T3}(u) = \lambda\ r + p_2(r) + p_3(r)/\lambda$$

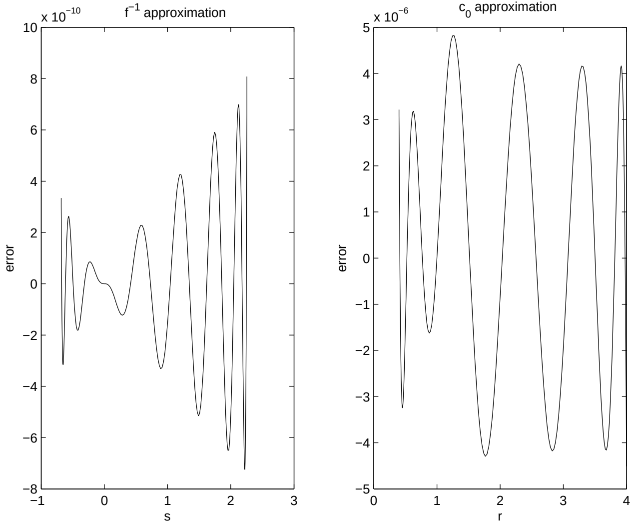
Temme approximation

The function $f(r)$



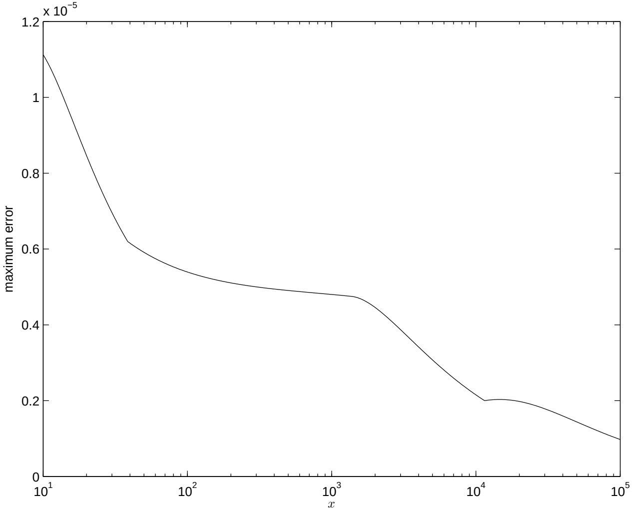
Temme approximation

Errors in $f^{-1}(s)$ and $c_0(r)$ approximations:



Temme approximation

Maximum error in $\widetilde{Q}_{T3}$ approximation:



$C(m)$ evaluation

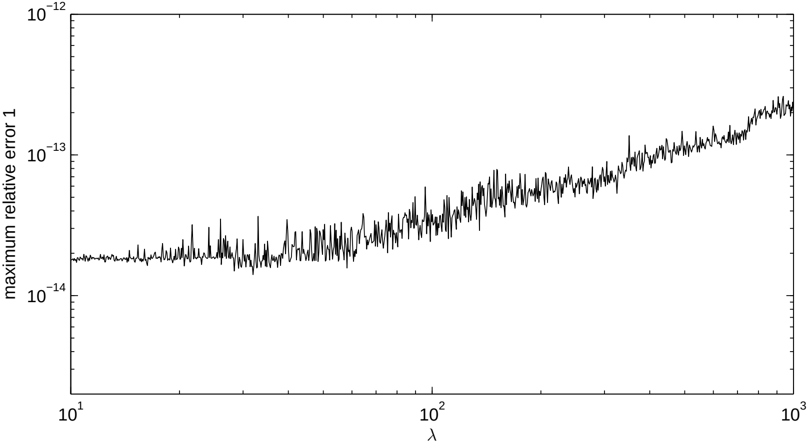
When $\widetilde{Q}(u)$ is too close to an integer, we need to evaluate $C(m)$ for integer m to choose between m and $m+1$.

When $\frac{1}{2}\,\lambda \leq m \leq 2\,\lambda$, this can be done very accurately using another approximation due to Temme (1987).

Outside this range, a modified version of bottom-up / top-down summation can be used, because successive terms decrease by factor 2 or more.

Temme approximation

Maximum relative error in Temme approximation for $C(m)$



The CPU algorithm

given inputs: λ, u

if $\lambda > 2.5$
    $w := \Phi^{-1}(u)$

    if $|w| < 3$
        $x := \widetilde{Q}_{N2}(w)$
        $\delta := \lambda^{-1}(\frac{1}{40} + \frac{1}{80}\,w^2 + \frac{1}{160}\,w^4)$
    else
        $r := f^{-1}(w/\sqrt{\lambda})$
        $x := \lambda\, r + c_0(r)$
        $x := x - (4.1/805)/(x{+}0.025\lambda)$
        $\delta := 0.01/\lambda$
    end

    $n := \lfloor x + \delta \rfloor$

The CPU algorithm

```
if x > 10
    if x−n > δ
        return n
    else if C(n) < u
        return n
    else
        return n−1
    end
end
end

if u ≤ 0.5
    use bottom-up summation to determine n
else
    use top-down summation to determine n
end
```

The GPU algorithm

given inputs: λ, u

if $\lambda > 2.5$

$w := \Phi^{-1}(u)$

$s := w/\sqrt{\lambda}$

$\delta := 5\times 10^{-7}\sqrt{\lambda}\;|w|$

if $s_{min} < s < s_{max}$

$r := p_1(s)$

$x := \lambda\, r + p_2(r) + p_3(r)/\lambda$

$\delta := \delta + 1.2\times 10^{-5}$

else

$r := f^{-1}(w/\sqrt{\lambda})$

$x := \lambda\, r + c_0(r)$

$x := x - (4.1/805)/(x + 0.025\lambda)$

$\delta := \delta + 0.01/\lambda$

end

$n := \lfloor x + \delta \rfloor$

The GPU algorithm

```
if x > 10
    if x−n > δ
        return n
    else if C(n) < u
        return n
    else
        return n−1
    end
end
end
```

use bottom-up summation to determine n

if not accurate enough
    use top-down summation to determine n
end

Conclusions

- By approximating the inverse incomplete Gamma function, have developed an approach for inverting the Poisson CDF for $\lambda > 2.5$
- Computational cost is roughly cost of inverse error function plus three polynomials of degree 8–12
- Slower than using rejection method for generating Poisson r.v.'s (at least on CPUs – may be competitive on GPUs) but can be used for Latin Hypercube sampling and QMC
- Open source implementation should be finished soon
- Student is starting work on the extension to the Binomial CDF using the incomplete Beta function

References

"The asymptotic expansion of the incomplete gamma functions", NM Temme, SIAM Journal of Mathematical Analysis, 10(4):757-766, 1979

"On the computation of the incomplete gamma functions for large values of the parameters", NM Temme, in *Algorithms for Approximation*, Clarendon Press, New York, 1987

"Fast evaluation of the inverse Poisson cumulative distribution function", MBG, in preparation, 2013.

Software will be available from my homepage when it's ready:
`http://people.maths.ox.ac.uk/gilesm/`