Statistical Computing II Assignment

RAMIT NANDI || MD2211

2024-05-30

Question

If $X_1, ..., X_{n+1}$ are independent exponentially distributed random variables with a common mean of 1, then we need to verify that $X_{(n+1)} - X_{(1)}$ has the density

$$n \sum_{k=1}^{n} (-1)^{k-1} \binom{n-1}{k-1} e^{-kx}$$

This is also the density of the sum $Y_1 + Y_2 + ... + Y_n$ of independent exponentially distributed random variables Y_i with respective means $1, \frac{1}{2}, ..., \frac{1}{n}$. In addition we need to compare the following approximations of the above density:

1. **Approximation Using Edgeworth Expansions**

$$g(x) = \phi(x) \left[1 + \frac{\rho_3 H_3(x)}{6\sqrt{n}} + \frac{\rho_4 H_4(x)}{24n} + \frac{\rho_3^2 H_3(x)}{72n} + O_P(n^{-3/2}) \right]$$

2. **Using Saddle Point Approximation**

$$g(x_0) = \frac{e^{-t_0 x_0 + nK(t_0)}}{\sqrt{2\pi K''(t_0)}} \left[1 + O(n^{-1}) \right]$$

3. **Using Refined Saddle Point Approximation**

$$g(x_0) = \frac{e^{-t_0 x_0 + nK(t_0)}}{\sqrt{2\pi K''(t_0)}} \left[1 + \frac{3\rho_4(t_0) - 5\rho_3^2(t_0)}{24n} + O(n^{-1}) \right]$$

Answer

We will first show that the density of $X_{(n+1)} - X_{(1)}$ has the form mentioned in question. To find the distribution of $X_{(n+1)} - X_{(1)}$, we start with the joint density of $X_{(1)}$ and $X_{(n+1)}$. The joint density of $X_{(1)}$ and $X_{(n+1)}$ is given by:

$$f_{X_{(1)},X_{(n+1)}}(u,v) = n(n+1)e^{-(u+v)}(e^{-u} - e^{-v})^{n-1} \quad \text{for } 0 < u < v < \infty.$$

Then We take the following transformation of variables: $Y_1 = X_{(n+1)} - X_{(1)}, Y_2 = X_{(1)}$. The Jacobian of this transformation is given by

$$J = \left| \frac{\partial(X_{(1)}, X_{(n+1)})}{\partial(Y_1, Y_2)} \right| = \begin{vmatrix} \frac{\partial X_{(1)}}{\partial Y_1} & \frac{\partial X_{(1)}}{\partial Y_2} \\ \frac{\partial X_{(n+1)}}{\partial Y_1} & \frac{\partial X_{(n+1)}}{\partial Y_2} \end{vmatrix} = \begin{vmatrix} 0 & 1 \\ 1 & 1 \end{vmatrix} = (0 \times 1 - 1 \times 1) = -1.$$

Then the joint density of (Y_1, Y_2) is given by

$$f_{Y_1,Y_2}(y_1, y_2) = n(n+1)e^{-(y_2+(y_1+y_2))}\left(e^{-y_2} - e^{-(y_1+y_2)}\right)^{n-1} \cdot 1.$$

After simplification, we have the following form -

$$f_{Y_1,Y_2}(y_1, y_2) = n(n+1)e^{-y_1}e^{-(n+1)y_2}(1 - e^{-y_1})^{n-1}.$$

Now, to obtain marginal density of Y_1, we integrate $f_{Y_1,Y_2}(y_1, y_2)$ w.r.t y_2.

$$f_{Y_1}(y_1) = \int_0^\infty n(n+1)e^{-y_1}e^{-(n+1)y_2}(1 - e^{-y_1})^{n-1}\, dy_2.$$

$$f_{Y_1}(y_1) = n(n+1)e^{-y_1}(1 - e^{-y_1})^{n-1} \int_0^\infty e^{-(n+1)y_2}\, dy_2.$$

The integral is the PDF of an exponential distribution with rate $n+1$, so it equals $1/(n+1)$:

$$\int_0^\infty e^{-(n+1)y_2}\, dy_2 = \frac{1}{(n+1)}$$

Thus we have the pdf of $Y_1 = X_{(n+1)} - X_{(1)}$ as

$$f_{Y_1}(y_1) = ne^{-y_1}(1 - e^{-y_1})^{n-1}, \text{for } y_1 > 0$$

Now, we need to show that $f_{Y_1}(x) = n\sum_{k=1}^{n}(-1)^{k-1}\binom{n-1}{k-1}e^{-kx}$. This follows easily by simplifying the summation.

$$
\begin{aligned}
n\sum_{k=1}^{n}(-1)^{k-1}\binom{n-1}{k-1}e^{-kx} &= ne^{-x}\sum_{k=1}^{n}(-1)^{k-1}\binom{n-1}{k-1}e^{-(k-1)x}\\
&= ne^{-x}\sum_{u=o}^{n-1}(-1)^{u}\binom{n-1}{u}e^{-ux} = ne^{-x}(1-e^{-x})^{n-1}
\end{aligned}
$$

Thus we have verified that that pdf of $X_{(n+1)} - X_{(1)}$ has the form mentioned in question.

We know the approximations for the density of the sum of IID random variables. We use the fact that the density of $X_{(n+1)} - X_{(1)}$ is the same as the density of the sum of independent random variables $Y_1 + Y_2 + \cdots + Y_n$, where Y_i are exponentially distributed with mean $\frac{1}{i}$. Although these variables are not identically distributed, we will test the validity of the approximations under this slight deviation.

To find these approximations, we first determine the cumulant generating function of $S_n = Y_1 + Y_2 + \cdots + Y_n$. If $K_n(t)$ is the cumulant generating function, it is defined as $K_n(t) = \log(M_n(t))$. Where $M_n(t)$ is the moment generating function of S_n:

$$
M_n(t) = \mathbb{E}[e^{tS_n}] = \prod_{j=1}^{n}\frac{j}{j-t} \quad \forall t < \min\{1, 2, ..n\}
$$

Since Y_j are independent and exponentially distributed with mean $\frac{1}{j}$, their MGF is $\mathbb{E}[e^{tY_j}] = \frac{j}{j-t}$. Thus, the cumulant generating function $K_n(t)$ is given by

$$
K_n(t) = -\sum_{j=1}^{n}\ln(1 - t/j)
$$

Taking $nK(t) = K_n(t)$, we calculate the m-th derivative as:

$$
nK^{(m)}(t) = \frac{d^m(nK_n(t))}{dt^m} = (m-1)!\sum_{j=1}^{n}\frac{1}{(j-t)^m}
$$

Using these we can compute the quantities ρ_i for the approximations.

Related Calculations

1. To calculate $\rho_l = \frac{\kappa_l}{\sigma^l}$ for $l = 3, 4$, we use:

   $$
   \rho_l = \frac{nK^{(l)}(0)}{[nK''(0)]^{l/2}}
   $$

   where H_n denotes the Hermite polynomial of order n. For the approximation at x_0, we use the standardized version $x = \frac{x_0 - nK^{'}(0)}{\sqrt{nK''(0)}}$, and then adjust by dividing by suitable constant.

2. To get the saddle point approximation at x_0, solve the saddle point equation $nK^{'}(t) = x_0$ numerically. After finding t_0, we compute the normalized cumulants:

   $$
   \rho_l = \frac{nK^{(l)}(t_0)}{[nK''(t_0)]^{l/2}}
   $$

Table 1: True density vs Approximations

x	True Value	Edgeworth Expansion	Saddle Point	Refined Saddle Point
0.5	0.0013710	0.0393630	0.0013813	0.0013803
1.0	0.0592795	0.1004206	0.0595481	0.0595243
1.5	0.2299814	0.1881552	0.2299022	0.2299403
2.0	0.3656290	0.2743311	0.3631593	0.3634832
2.5	0.3797397	0.3230587	0.3747353	0.3752994
3.0	0.3144162	0.3169296	0.3088902	0.3094062
3.5	0.2291498	0.2662806	0.2247890	0.2250747
4.0	0.1550847	0.1959247	0.1528157	0.1528872
4.5	0.1004640	0.1281999	0.0994611	0.0994085
5.0	0.0634019	0.0756288	0.0633168	0.0632205
5.5	0.0393889	0.0411022	0.0395861	0.0394893
6.0	0.0242400	0.0211733	0.0247717	0.0246926
6.5	0.0148322	0.0104753	0.0151727	0.0151133
7.0	0.0090443	0.0048673	0.0094179	0.0093762
7.5	0.0055034	0.0020307	0.0057143	0.0056860
8.0	0.0033445	0.0007305	0.0035018	0.0034830
8.5	0.0020310	0.0002210	0.0021271	0.0021150
9.0	0.0012327	0.0000556	0.0013114	0.0013036
9.5	0.0007480	0.0000116	0.0007990	0.0007941
10.0	0.0004538	0.0000020	0.0004742	0.0004711

Results

We calculated the approximations for $n = 10$ and compared them with the true density given by $g_n(x) = n\sum_{k=1}^{n}(-1)^{k-1}\binom{n-1}{k-1}e^{-kx}$. We also considered a fixed set of x values, ranging from 0.5 to 10 in steps of 0.5. We have created a table showing the true density values and the approximations for each case. Also the Percentage Error

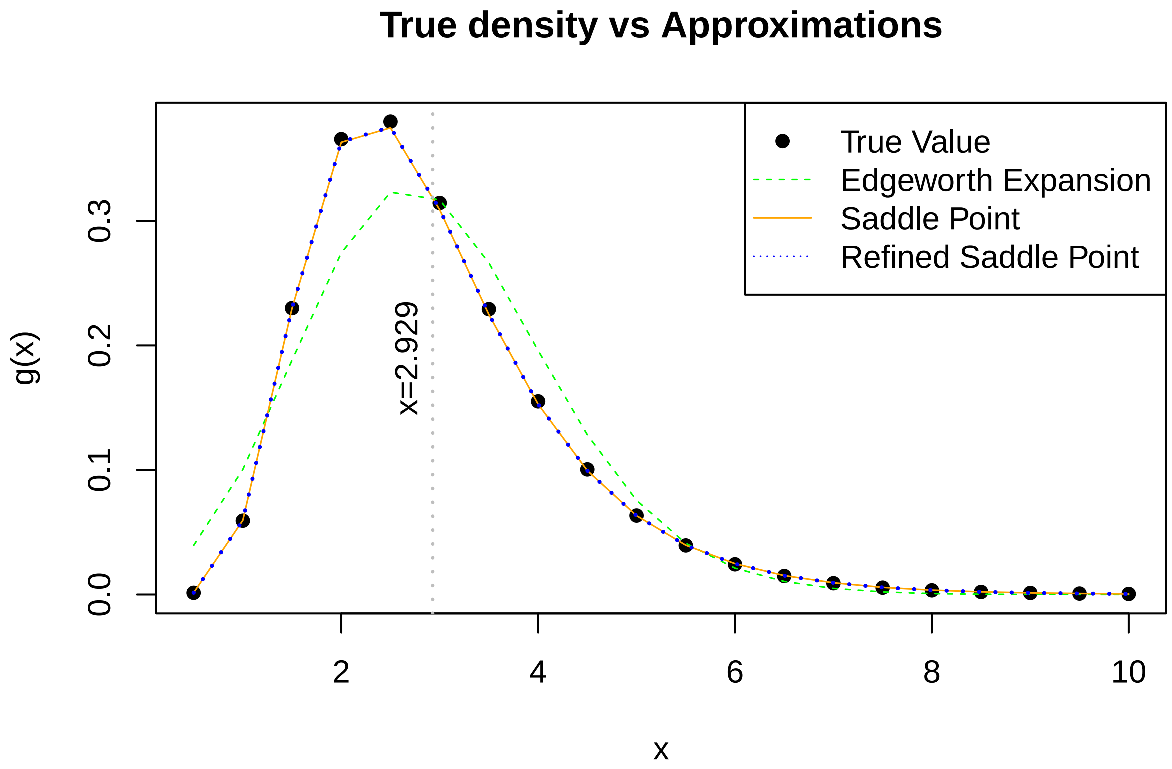
$$100 \times |\text{True value} - \text{Approx.}|/\text{True value}$$

are reported in a separate table.

Comment: In Table 1 and Table2, it is evident that Saddle Point approximations generally provide more accurate results compared to the Edgeworth expansion, except near the mean value $\sum_{i=1}^{10} 1/j \approx 2.93$. Particularly for small values of x and in the tail regions, the Saddle Point approximations perform significantly better. This is due to the fact that the Y_j's have highly variable variances, making a simple normal approximation based on the central limit theorem less reliable.

Table 2: Percentage errors

x	Edgeworth Expansion	Saddle Point	Refined Saddle Point
0.5	2771.0242794	0.7461178	0.6726952
1.0	69.4017890	0.4531181	0.4128471
1.5	18.1867586	0.0344041	0.0178392
2.0	24.9701075	0.6754621	0.5869013
2.5	14.9262900	1.3178502	1.1693119
3.0	0.7993725	1.7575466	1.5934420
3.5	16.2037458	1.9030257	1.7783523
4.0	26.3340206	1.4630343	1.4169475
4.5	27.6077871	0.9983088	1.0506532
5.0	19.2847158	0.1342473	0.2860592
5.5	4.3496240	0.5007725	0.2547964
6.0	12.6512226	2.1936431	1.8671596
6.5	29.3744501	2.2955489	1.8950014
7.0	46.1834062	4.1314980	3.6699064
7.5	63.1003177	3.8335612	3.3184044
8.0	78.1568623	4.7024760	4.1419504
8.5	89.1169764	4.7344416	4.1360754
9.0	95.4913243	6.3830636	5.7490079
9.5	98.4535607	6.8179307	6.1548545
10.0	99.5610444	4.4855570	3.8067332



Codes:

```
## Method 0 ........................
gn <- function(x, n){
  k = 1:n
  n*sum(((-1)^(k-1))*choose(n-1, k-1)*exp(-k * x))
}


## Method 1 ........................
edgeworth_approx <- function(x0, n){
  Kd = .cumulant_derivative(0, n, m=1)
  Kdd = .cumulant_derivative(0, n, m=2)

  y0 = (x0 - Kd) / sqrt(Kdd)
  rho_3 = .cumulant_derivative(0, n, m=3) / (Kdd)^(3/2)
  rho_4 <- .cumulant_derivative(0, n, m=4) / (Kdd)^(4/2)

  factor = (rho_3 * .Hermite_polynomial(y0, n = 3) / (6 * sqrt(n))) +
            (rho_4 * .Hermite_polynomial(y0, n = 4) / (24 * n)) +
            ((rho_3^2) * .Hermite_polynomial(y0, n = 6) / (72 * n))

  return(dnorm(y0) * (1 + factor) / sqrt(Kdd))
}


## Method 2 ........................
saddle_point_approx <- function(x0, n) {
  t0 = uniroot(f = .saddle_point_eq(x0,n),
               interval = c(-400,1),
                extendInt = "yes", tol = 1e-02)$root

  factor1 = exp(-t0 * x0 + .cumulant_derivative(t0, n, m=0))
  factor2 = sqrt(2 * pi * .cumulant_derivative(t0, n, m=2))

  return(factor1 / factor2)
}


## Method 3 ........................
saddle_point_refined_approx <- function(x0, n) {
  t0 = uniroot(f = .saddle_point_eq(x0,n),
               interval = c(-400,1),
                extendInt = "yes", tol = 1e-02)$root

  Kdd_p <- .cumulant_derivative(t0, n, m=2)
  rho_3 <- .cumulant_derivative(t0, n, m=3) / (Kdd_p)^(3/2)
  rho_4 <- .cumulant_derivative(t0, n, m=4) / (Kdd_p)^(4/2)

  factor1 <- exp(-t0 * x0 + .cumulant_derivative(t0, n, m=0))
  factor2 <- sqrt(2 * pi * .cumulant_derivative(t0, n, m=2))
  factor3 <- (3 * rho_4 - 5 * (rho_3^2)) / (24 * n)

  return((1 + factor3) * factor1 / factor2)
}


## Useful functions ========================================
.Hermite_polynomial <- function(x, n){
  H = c(1,x,NA)
  if(n>1){
      for(j in 2:n){
        H[(j%%3)+1] = x*H[((j-1)%%3)+1] - (j-1)*H[((j-2)%%3)+1]
        # we don't want to use recursive function call
        # instead we keep track of the past two values
        # to avoid redundant calculation
      }
  }
  return(H[(n%%3)+1])
}

.cumulant_derivative <- function(t, n, m=0){
  ifelse(m,
         factorial(m-1)* sum(1/(((1:n)-t)^m)),
         -sum(log(1 - t/(1:n))))
}

.saddle_point_eq <- function(x0, n) {
  return(function(t) return(sum(1/((1:n)-t))-x0))
}
```