NONPARAMETRIC ESTIMATION OF
THE CUMULATIVE INTENSITY FUNCTION
FOR A NONHOMOGENEOUS POISSON PROCESS

Lawrence M. Leemis

The University of Oklahoma

School of Industrial Engineering

202 West Boyd, Room 124

Norman, OK 73019

November, 1990

A nonparametric technique for estimating the cumulative intensity function of a nonhomogeneous Poisson process from one or more realizations is developed. This technique does not require any arbitrary parameters from the modeler, and the estimated cumulative intensity function can be used to generate a point process for Monte Carlo simulation by inversion. Three examples are given.

(NONSTATIONARY POISSON PROCESS; REPAIRABLE SYSTEMS; TIME-DEPENDENT ARRIVALS; SIMULATION; VARIATE GENERATION)

1. Introduction

A nonhomogeneous Poisson process (NHPP) is often suggested as an appropriate model for a system whose rate (e.g. arrival rate in a queuing system) varies over time. This paper illustrates a nonparametric technique for estimating the cumulative intensity function of a NHPP on the time interval $(0, S]$ from one or more realizations. This procedure only applies to terminating simulations. Unlike many existing techniques, this method does not require the modeler to specify any parameters or weighting functions. If the NHPP is used as an input to a Monte Carlo simulation, inversion can be used to generate event times so that variance reduction techniques can be implemented. Although the discussion here is oriented towards arrivals to queuing systems, the estimation technique applies to any sequence of events occurring over time or space, such as earthquake times, failure times of a repairable system or defect positions on a magnetic tape.

A NHPP is a generalization of an ordinary Poisson process where events occur randomly over time at the rate of λ events per unit time. The rate at which events occur in a NHPP varies over time as determined by the *intensity function,* $\lambda(t)$. The *cumulative intensity function* is defined by

$$\Lambda(t) = \int_0^t \lambda(\tau)\,d\tau \qquad\qquad t > 0$$

and is interpreted as the expected number of events by time t. The probability of exactly n events occurring in the interval $(a, b]$ is given by

$$\frac{\left[\int_a^b \lambda(t)dt\right]^n e^{-\int_a^b \lambda(t)dt}}{n!}$$

for $n = 0, 1, \ldots$ (Cinlar, 1975).

Many simulation textbook authors (e.g., Bratley, Fox and Schrage (1987), Fishman (1978), Lavenberg (1983), Law and Kelton (1991), Lewis and Orav (1989), Morgan (1984) and Ross(1990)) suggest the use of NHPPs for modeling systems with inputs whose rates vary over time. Schmeiser (1980) reviews variate generation techniques for NHPPs, including *thinning* (Lewis and Shedler, 1979b), where a NHPP can be simulated when the intensity function is not tractable and inversion is not closed form.

There have been several parametric techniques suggested for estimating the cumulative intensity function from a data set. One of these efforts assumes that

$$\Lambda(t) = (\alpha t)^{\beta} \qquad\qquad t > 0$$

which is often called a power law or Weibull process (Bain and Engelhardt (1982), Jang and Bai (1987), Rigdon and Basu (1989), Rigdon and Basu (1990)). Lee, Wilson and Crawford (1991) suggest a general model that uses an exponential-polynomial-trigonometric function in the intensity function

$$\lambda(t) = \exp\left\{\sum_{i=0}^{m} \alpha_i t^i + \gamma \sin(\omega t + \phi)\right\} \qquad\qquad t > 0$$

which they apply to modeling off-shore weather events in the Arctic Sea involving both a cyclic component and a trend. Kao and Chang (1988) model the times of calls for analysis of electrocardiograms at a hospital over several days using a piecewise-polynomial intensity function. Law and Kelton (1991, page 407) suggest a nonparametric procedure for estimating $\lambda(t)$ with a piecewise-constant function. Their procedure requires the modeler to divide the time axis into nonoverlapping time intervals where the intensity function is assumed to be fairly constant, and estimate a single rate for each interval. While this procedure is simple to implement, the modeler must make arbitrary decisions concerning the number and widths of the intervals. Lewis and Shedler (1976b) illustrate techniques for estimating $\lambda(t)$ to model the transactions in a database system. One nonparametric estimator that they define is

$$\hat{\lambda}(t;\, n,\, t_0) = \frac{1}{b(n)} \sum_{j=1}^{n} W\left(\frac{t - T_j}{b(n)}\right) \qquad t > 0$$

where t_0 is the upper limit of the time interval, n is the number of observations in $(0,\, t_0]$, $T_1, \ldots, T_n$ are the observations, W is a bounded, nonnegative, integrable weight function satisfying $\int_{-\infty}^{\infty} W(u)du = 1$, and $b(n)$ is a bandwidth function that tends to zero as n approaches infinity. Nelson (1988) considers nonparametric estimates for the cumulative cost and repair functions for repairable system data that includes right censored observations. Vallarino (1988) uses a time scale transformation of a Brownian bridge on [0, 1] to derive simultaneous confidence bands around an estimator that is similar to the one given here.

Other articles on NHPPs and their application to queuing systems include Albin (1982), Chouinard and McDonald (1985), Foley (1986) and Thorisson (1985). Work on generating variates from a NHPP includes Devroye (1986), Fishman and Kao (1977), Kaminsky and Rumph (1977), Klein and Roberts (1984), Lee, Wilson and Crawford

(1991), Lewis and Shedler (1976a, 1979a), and Shanthikumar (1986).

2. Estimation Procedure

The intensity function, $\lambda(t)$, for a NHPP is assumed to be positive for all $t \in (0, S]$ and is continuous for almost every $t \in (0, S]$. The cumulative intensity function is to be estimated from k realizations of the NHPP on $(0, S]$, where S is a known constant. The interval $(0, S]$ may represent the time a system allows arrivals (e.g., 9 AM to 5 PM at a bank) or one period of a cycle (e.g., one day at a 24 hour drive-up window). The estimation procedure described in this section is nonparametric and does not require any arbitrary decisions (e.g., parameter values) from the modeler. Let n_i $(i = 1, 2, \ldots, k)$ be the number of observations in the i^{th} realization, $n = \sum_{i=1}^{k} n_i$, and let $t_{(1)}, t_{(2)}, \ldots, t_{(n)}$ be the order statistics of the superposition of the k realizations, $t_{(0)} = 0$ and $t_{(n+1)} = S$. Setting $\hat{\Lambda}(S) = \frac{n}{k}$ yields a process where the expected number of events by time S is the average number of events in k realizations, since $\Lambda(S)$ is the expected number of events by time S. The piecewise-linear estimator of the cumulative intensity function between the time values in the superposition is

$$\hat{\Lambda}(t) = \frac{in}{(n+1)k} + \left[\frac{n(t - t_{(i)})}{(n+1)\,k(t_{(i+1)} - t_{(i)})}\right] \qquad t_{(i)} < t \le t_{(i+1)};\ i = 0, 1, 2, \ldots, n$$

as illustrated in Figure 1. This estimator passes through the points $\left(t_{(i)}, \frac{in}{(n+1)k}\right)$, for $i = 1, 2, \ldots, n+1$. The $\frac{n}{n+1}$ factor in the value of the estimate for the cumulative intensity function at the data values accounts for the fact that there are $n+1$ "gaps" created on $(0, S]$ by the data values.

The assumption that there will not be any ties, i.e., $t_{(i)} < t_{(i+1)}$ for $i = 0, 1, \ldots, n$, may not always be satisfied in practice due to rounding. The estimate for $\Lambda(t)$ given above should be modified so that there is a discontinuity at the value where tied values occur. For example, if $t_{(m)} = t_{(m+1)}$ for some m, then $\hat{\Lambda}(t_{(m)}) = \hat{\Lambda}(t_{(m+1)}) = \dfrac{mn}{(n+1)\,k}$, and $\lim_{t \downarrow t_{(m+1)}} = \dfrac{(m+1)n}{(n+1)\,k}$. In other words, there is a jump in the estimate of the cumulative intensity function of $\dfrac{n}{(n+1)\,k}$ where the tie occurs. Multiple tied values are handled analogously. An example containing tied values is given in Section 5. Note that the variate generation algorithm described in the next section will not be affected by tied observations, although there may be multiple event times generated at the values where ties occur in the data set.

The rationale for using a linear function between the data values is that inversion can be used for generating realizations (as shown in Section 3) without having tied events. If the usual step-function estimate of $\Lambda(t)$ is used (see the Appendix for a definition), only the $t_{(i)}$ values could be generated.

Some empirical justification for the use of the proposed estimator is provided in Figures 2a, 2b, 2c, and 2d, where the population cumulative intensity function (smooth curves) and the proposed estimator (piecewise-linear curves) are plotted for four different processes. The $k = 5$ realizations are generated by thinning. The four parent intensity functions are the piecewise-linear intensity function

$$\lambda(t) = \begin{cases} 10t + 1 & 0 < t \le 1.5 \\ 16 & 1.5 < t \le 2.5 \\ -6t + 31 & 2.5 < t \le 4.5 \end{cases}$$

from Klein and Roberts (1984),

$$\lambda(t) = 1 + \cos t \qquad 0 < t \le 4\pi,$$

yielding a cyclic arrival rate,

$$\lambda(t) = e^{2t-1} \qquad 0 < t \le 3$$

from Lewis and Shedler (1976a) and

$$\lambda(t) = t^2 \qquad 0 < t \le 5,$$

a special case of a power law process. The sample sizes for the four processes are $n = 247, 45, 39, 216$, respectively. In all four plots, the estimator roughly follows the shape of the parent cumulative intensity function, and improves with n, the number of observations collected in the five realizations.

Since the number of events that occur in the NHPP of interest by time t has the Poisson distribution with mean $\Lambda(t)$, a strong consistency result is obtained, i.e.,

$$\lim_{k \to \infty} \hat{\Lambda}(t) = \Lambda(t) \quad \text{with probability one.}$$

The proof, given in the appendix, uses the fact that the proposed estimator can be expressed as a function of the usual step-function estimator for the cumulative intensity function. The appendix also contains a derivation of an asymptotically exact $100(1-\alpha)\%$ confidence interval for $\Lambda(t)$

$$\hat{\Lambda}(t) - z_{\alpha/2}\sqrt{\frac{\hat{\Lambda}(t)}{k}} < \Lambda(t) < \hat{\Lambda}(t) + z_{\alpha/2}\sqrt{\frac{\hat{\Lambda}(t)}{k}}$$

where $z_{\alpha/2}$ is the $1 - \alpha/2$ fractile of the standard normal distribution.

3. Variate Generation

The cumulative intensity function for a NHPP is often estimated in order to generate variates for Monte Carlo simulation. Using a time transformation (Cinlar, 1975, page 96), the event times from a unit Poisson process, E_1, E_2, ..., can be transformed to the event times of a NHPP via $T_i = \Lambda^{-1}(E_i)$. For the NHPP estimate considered here, the events at times $T_1, T_2, \ldots$ can be generated for Monte Carlo simulation by the algorithm below, given n, k, S and the superpositioned values.

1. $i \leftarrow 1$
2. generate $U_i \sim U(0, 1)$
3. $E_i \leftarrow -\log_e (1 - U_i)$
4. **while** $E_i < \dfrac{n}{k}$ **do**

 begin

$$m \leftarrow \left\lfloor \frac{(n+1)\, k\, E_i}{n} \right\rfloor$$

$$T_i \leftarrow t_{(m)} + [t_{(m+1)} - t_{(m)}] \left(\frac{(n+1)\, k\, E_i}{n} - m \right)$$

$$i \leftarrow i + 1$$

$$\text{generate } U_i \sim U(0, 1)$$

$$E_i \leftarrow E_{i-1} - \log_e (1 - U_i)$$

 end

Thus, it is a straightforward procedure to obtain a realization of $i - 1$ events on $(0, S]$ from the superpositioned process and $U(0, 1)$ values $U_1, U_2, \ldots, U_i$. Inversion has been used to generate this NHPP, so certain variance reduction techniques, such as antithetic variates or common random numbers, may be applied to simulation output. Replacing $1-U_i$ with U_i in steps 3 and 4 will save CPU time although the direction of the monotonicity is reversed. Tied values in the superposition do not pose any problem to this algorithm although there may be tied values in the realization. As n increases, the amount of memory required increases, but the amount of CPU time required to generate a realization depends only on the ratio n/k, the average number of events per realization. Thus collecting more realizations (resulting in narrower confidence intervals) increases the amount of memory required, but does not impact the expected CPU time for generating a realization.

4. Examples

Two examples will be given in this section. The first contains a rush hour situation, and the second contains an arrival pattern which is cyclic.

The procedure for computing the nonparametric estimate of $\Lambda(t)$ is illustrated using $k=3$ realizations of a process on (0, 4.5]. The events in this example are arrivals to a lunchwagon between 10:00 AM and 2:30 PM (Klein and Roberts 1984), and the three realizations ($n_1=46$, $n_2=69$, $n_3=49$) and their superposition ($n=164$) were generated by thinning. The realizations were generated from a population with parent cumulative intensity function

$$\Lambda(t) = \begin{cases} 5t^2 + t & 0 < t \le 1.5 \\ 16t - 11.25 & 1.5 < t \le 2.5 \\ -3t^2 + 31t - 30 & 2.5 < t \le 4.5 \end{cases}$$

The parent cumulative intensity function, the estimated cumulative intensity function and 95% confidence bands are shown in Figure 3. The smooth curve is the parent cumulative intensity function, the piecewise-linear function is $\hat{\Lambda}(t)$ for the $n = 164$ observations in the superpositioned process and the dashed lines are 95% confidence bands. Since the intensity function increases linearly initially, is constant between 11:30 AM and 12:30 PM, then decreases linearly, the nonparametric approach provides a more accurate model than using a parametric model, such as a power law process.

A Monte Carlo experiment was conducted to assess the accuracy of the confidence intervals in the lunchwagon example with three realizations at times 1.5, 2.5, and 3.5. For 100,000 replications of the experiment at nominal coverage 0.95, the actual coverages at the three points in time were 0.94754, 0.94779, and 0.94675. This experiment indicates that the approximate confidence intervals for the cumulative intensity function estimate are fairly accurate for a large sample size n. This is not a surprising result since the Poisson distribution converges to a normal distribution as its mean increases.

The second example illustrates how the estimator tracks the cyclic intensity function considered earlier

$$\lambda(t) = 1 + \cos(t) \qquad 0 < t \le 4\pi,$$

which corresponds to a cumulative intensity function

$$\Lambda(t) = t + \sin(t) \qquad 0 < t \le 4\pi.$$

In this case, $k = 10$ realizations of the process were generated by thinning yielding

$n = 120$ observations. Figure 4 shows the parent cumulative intensity function, the estimated cumulative intensity function and 95% confidence bands for the cumulative intensity function. The parent cumulative intensity function falls outside the 95% confidence bands at approximately $t = 0.4$ and $t = 1.6$. It was determined that this was due to sampling variability since a Monte Carlo study using 100,000 replications yielded coverages of 0.94542, 0.94714, and 0.94839 at times $t = 0.4$, $t = 1.6$ and $t = 2\pi$, respectively, for 95% confidence intervals.

5. Extensions

This section presents two extensions to the nonparametric cumulative intensity function estimator given in Section 2. The first extension accommodates time intervals on $(0, S]$ where events cannot occur. The second extension involves the use of a piecewise- quadratic, rather than a piecewise-linear estimate of the cumulative intensity function.

In the discussion so far, it has been assumed that $\lambda(t) > 0$ for all t in $(0, S]$. The cumulative intensity function estimation technique in Section 2 does not account for periods of time (e.g., lunchbreaks) where events can not occur. If the beginning and ending times of these breaks are known, they are easily incorporated into the cumulative intensity function estimator.

Consider the interval $(a, b]$, (with a and b known) where the intensity function is assumed to be zero. Let $t_{(i)}$ be the time of the most recent event prior to a and $t_{(i+1)}$ be the time of the first event after b. The dashed line in Figure 5 shows $\hat{\Lambda}(t)$ on $(t_{(i)}, t_{(i+1)}]$

without a lunchbreak, and the solid line shows the modifications proposed below. The cumulative intensity function estimate should be constant on $(a, b]$ and the piecewise-linear segments on $(t_{(i)}, a]$ and $(b, t_{(i+1)}]$ follow the usual pattern. On the interval $(t_{(i)}, t_{(i+1)}]$, one estimator is

$$\hat{\Lambda}_1(t) = \begin{cases} \dfrac{in}{(n+1)k} + \dfrac{n(a+b-2t_{(i)})(t-t_{(i)})}{2(n+1)k(a-t_{(i)})(t_{(i+1)}-t_{(i)})} & t_{(i)} < t \le a \\ \dfrac{n(a+b-2t_{(i)}+2i(t_{(i+1)}-t_{(i)}))}{2(n+1)k(t_{(i+1)}-t_{(i)})} & a < t \le b \\ \dfrac{(i+1)n}{(n+1)k} + \dfrac{n(2(t_{(i+1)}-t_{(i)})-a-b+2t_{(i)})(t-t_{(i+1)})}{2(n+1)k(t_{(i+1)}-b)(t_{(i+1)}-t_{(i)})} & b < t \le t_{(i+1)} \end{cases}$$

which is determined by setting the cumulative intensity on $a < t \le b$ to $\hat{\Lambda}(\frac{(a+b)}{2})$, and using a linear estimate on the other intervals. A second way of accommodating lunchbreaks is to match the slopes of the cumulative intensity function estimates on the intervals $(t_{(i)}, a]$ and $(b, t_{(i+1)}]$. This results in a slightly more tractable cumulative intensity estimate

$$\hat{\Lambda}_2(t) = \begin{cases} \dfrac{in}{(n+1)k} + \dfrac{n(t-t_{(i)})}{(n+1)k(t_{(i+1)}-t_{(i)}-b+a)} & t_{(i)} < t \le a \\ \dfrac{in}{(n+1)k} + \dfrac{n(a-t_{(i)})}{(n+1)k(t_{(i+1)}-t_{(i)}-b+a)} & a < t \le b \\ \dfrac{(i+1)n}{(n+1)k} + \dfrac{n(t-t_{(i+1)})}{(n+1)k(t_{(i+1)}-t_{(i)}-b+a)} & b < t \le t_{(i+1)} \end{cases}$$

The two estimators are almost identical when the lunchbreak is short relative to $t_{(i+1)} - t_{(i)}$ (i.e., $\dfrac{b-a}{t_{(i+1)}-t_{(i)}}$ is small), and are identical when $a - t_{(i)} = t_{(i+1)} - b$.

One drawback with the assumption of a piecewise-constant intensity function is the possibility of unrealistic jumps in $\hat{\lambda}(t)$ at the data values. This may cause problems if n is small or if there is considerable nonlinearity in the intensity function. Figure 6 shows the estimator for the cumulative intensity function estimate for $k = 1$ realization of unscheduled maintenance action times on the U.S.S. Halfbeak No. 3 main propulsion diesel engine (Ascher and Feingold, 1984, page 75). Scheduled engine overhauls are not treated separately for this data set of $n = 78$ event times, and the ending time of the observation interval is assumed to be $S = 25,600$ hours. There appears to be significant degree of nonlinearity after 20,000 hours, and the adjustment to the estimator outlined below may be warranted. There are tied values at times 11993, 24006 and 25000, and the cumulative intensity function is discontinuous at these values.

The estimator can be easily modified when there are no ties to be a piecewise-linear intensity function by joining the midpoints of the intensity function values between each of the data points as shown for $n = 4$ by the dashed line in Figure 7. Since the value of the intensity function for the nonparametric estimator between $t_{(i)}$ and $t_{(i+1)}$ is

$$\hat{\lambda}(t) = \frac{n}{(n+1)\,k\,[t_{(i+1)} - t_{(i)}]} \qquad t_{(i)} < t \le t_{(i+1)};\ i = 0, 1, \ldots, n$$

the midpoints can be joined with a line to yield the piecewise- linear estimator

$$\hat{\lambda}_3(t) = \frac{n}{(n+1)\,k\,[t_{(i+1)} - t_{(i)}]} \left[1 + \frac{(2t_{(i+1)} - t_{(i+2)} - t_{(i)})\,(2t - t_{(i)} - t_{(i+1)})}{(t_{(i+2)} - t_{(i+1)})\,(t_{(i+2)} - t_{(i)})} \right]$$

for $\frac{t_{(i)} + t_{(i+1)}}{2} < t \le \frac{t_{(i+1)} + t_{(i+2)}}{2}$ and $i = 0, 1, \ldots, n-1$. This accounts for all time periods except the intervals $(0, \frac{t_{(1)}}{2}]$ and $(\frac{t_{(n)} + S}{2}, S]$, where the corresponding $\hat{\lambda}(t)$

value can be used. Variate generation may be performed by using the technique in Lee, Wilson and Crawford (1991), deleting Step 7 in their event time generation algorithm, where thinning is performed.

6. Summary

A method has been presented for the nonparametric estimation of the cumulative intensity function for a NHPP from one or more realizations. The method does not require any arbitrary parameters to be specified, and is easily generated via inversion. Time intervals where events cannot occur are easily accommodated, and the method can be extended to a piecewise- quadratic estimate.

As in classical statistics, an estimate from a single realization ($k = 1$) or a small total number of observations (i.e., n small) should be considered cautiously due to sampling variability. Estimates containing uncharacteristically clustered event times, for example, will produce simulations with the same feature. It is worthwhile having *several*, rather than *one* realization (to see the variability from one realization to the next and since the confidence interval is asymptotically valid with respect to the number of realizations), and the sample size should be large enough so that the halfwidth of the confidence interval for $\Lambda(t)$ is sufficiently small.

Acknowledgement

The author thanks Jim Wilson for providing the proof given in the Appendix and editing the manuscript, Carlos Vallarino, Jerry Lawless and two referees for their help with the contents of the paper, and Bryan Norman and Todd Tillinghast for their assistance in preparing the figures for this paper.

Appendix

This appendix contains (i) a proof of strong consistency for $\hat{\Lambda}(t)$, i.e., the estimate $\hat{\Lambda}(t) \to \Lambda(t)$ with probability one as the number of realizations collected, k, approaches infinity for all $t \in (0, S]$ and (ii) a derivation of an asymptotically valid $100(1-\alpha)\%$ confidence interval for $\Lambda(t)$ for all $t \in (0, S]$.

Consider first the corresponding properties of the usual step-function estimator of $\Lambda(t)$. For the jth independent replication of the target NHPP ($j = 1, \ldots, k$), let $N_j(t)$ denote the number of events observed in the time interval $(0, t]$ and let

$$N_k^*(t) \equiv \sum_{j=1}^{k} N_j(t) \quad \text{for all } t \in (0, S] \tag{1}$$

denote the aggregated counting (or superposition) process so that $n = N_k^*(S)$. The usual step-function estimator of $\Lambda(t)$ is

$$\tilde{\Lambda}(t) \equiv \frac{1}{k} \sum_{j=1}^{k} N_j(t) = \frac{N_k^*(t)}{k} \quad \text{for all } t \in (0, S]. \tag{2}$$

Now the $\{N_j(t)\colon j = 1, \ldots, k\}$ are IID Poisson variates with mean $\Lambda(t)$; and it follows immediately that

$$\left\{ \begin{array}{l} E[\tilde{\Lambda}(t)] = \Lambda(t) \\ V[\tilde{\Lambda}(t)] = \Lambda(t)/k \end{array} \right\} \quad \text{for all } t \in (0, S]. \tag{3}$$

Given an arbitrary $t \in (0, S]$, we can apply the Strong Law of Large Numbers to conclude that

$$\lim_{k \to \infty} \tilde{\Lambda}(t) = \Lambda(t) \quad \text{with probability one;} \tag{4}$$

moreover by the Central Limit Theorem, equation (4), and Slutsky's Theorem (Serfling 1980), we have

$$\frac{\tilde{\Lambda}(t) - \Lambda(t)}{\sqrt{\tilde{\Lambda}(t)/k}} = \sqrt{\frac{\Lambda(t)}{\tilde{\Lambda}(t)}} \cdot \frac{\tilde{\Lambda}(t) - \Lambda(t)}{\sqrt{\Lambda(t)/k}} \xrightarrow{D} 1 \cdot N(0,\, 1) \sim N(0,\, 1) \; as \; k \to \infty. \tag{5}$$

From (5) we can construct the following asymptotically exact $100(1-\alpha)\%$ confidence interval for $\Lambda(t)$:

$$\tilde{\Lambda}(t) \pm z_{\alpha/2} \sqrt{\frac{\tilde{\Lambda}(t)}{k}} \, . \tag{6}$$

Objectives (i) and (ii) are shown by relating the proposed estimator $\hat{\Lambda}(t)$ to the step-function estimator $\tilde{\Lambda}(t)$. For any fixed $t \in (0,\, S]$, we have

$$\hat{\Lambda}(t) = U_k \tilde{\Lambda}(t) + R_k(t), \tag{7}$$

where the random variables U_k and $R_k(t)$ are given by

$$U_k \equiv \frac{n}{n+1} = \frac{N_k^*(S)}{N_k^*(S)+1} \tag{8}$$

and

$$R_k(t) \equiv \left\{ \frac{N_k^*(S)}{[N_k^*(S)+1]k} \right\} \left[\frac{t - t_{(N_k^*(t))}}{t_{(N_k^*(t)+1)} - t_{(N_k^*(t))}} \right] \tag{9}$$

In view of (2) and (4), we must have

$$\lim_{k \to \infty} N_k^*(S) = \infty \quad \text{with probability one;} \tag{10}$$

and combining (8) with (10), we have

$$\lim_{k \to \infty} U_k = 1 \quad \text{with probability one.} \tag{11}$$

Moreover we observe that the random term enclosed in large square brackets on the right-hand side of (9) is always bounded between 0 and 1; and thus for an arbitrary fixed

$t \in (0, S]$, we have

$$0 \le R_k(t) \le \frac{N_k^*(S)}{[N_k^*(S)+1]\,k} \tag{12}$$

and using (10), this implies that

$$\lim_{k\to\infty} R_k(t) = 0 \quad \text{with probability one.} \tag{13}$$

Combining (4), (7), (11), and (13), we finally obtain the desired strong consistency property: given an arbitrary $t \in (0, S]$, we have

$$\lim_{k\to\infty} \hat{\Lambda}(t) = \Lambda(t) \quad \text{with probability one.} \tag{14}$$

Moreover, the relation (7) coupled with (5), (11), (13), (14), and Slutsky's Theorem implies that $\hat{\Lambda}(t)$ is asymptotically normal:

$$\frac{\hat{\Lambda}(t)-\Lambda(t)}{\sqrt{\hat{\Lambda}(t)/k}} = U_k \sqrt{\frac{\Lambda(t)}{\hat{\Lambda}(t)}} \cdot \frac{\tilde{\Lambda}(t)-\Lambda(t)}{\sqrt{\Lambda(t)/k}} + \sqrt{\frac{\Lambda(t)}{\hat{\Lambda}(t)}} \cdot \frac{R_k(t)}{\sqrt{\Lambda(t)/k}}$$

$$\xrightarrow{D} 1 \cdot N(0,1) + 1 \cdot 0 \sim N(0,1) \ as\ k \to \infty. \tag{15}$$

From (15) we can construct the following asymptotically exact $100(1-\alpha)\%$ confidence interval for $\Lambda(t)$:

$$\hat{\Lambda}(t) \pm z_{\alpha/2} \sqrt{\frac{\hat{\Lambda}(t)}{k}}\,. \tag{16}$$

References

ALBIN, S.L., "On Poisson Approximations for Superposition Arrival Processes in Queues," *Management Sci.*, 28, 2 (1982), 126-137.

ASCHER, H. AND FEINGOLD, H., Repairable Systems Reliability, Marcel Dekker, New York, 1984.

BAIN, L.J. AND M. ENGELHARDT, "Sequential Probability Ratio Tests for the Shape Parameter of a Nonhomogeneous Poisson Process," *IEEE Trans. on Reliability,* R-31, 1 (1982), 79-83.

BRATLEY, P., B.L. FOX, AND L.E. SCHRAGE, *A Guide to Simulation, Second Edition,* Springer-Verlag, 1987.

CHOUINARD, A. AND D. MCDONALD, "A Characterization of Non-Homogeneous Poisson Processes," *Stochastics*, 15 (1985), 113-119.

CINLAR, E., *Introduction to Stochastic Processes,* Prentice-Hall, 1975.

DEVROYE, L. *Non-Uniform Random Variate Generation,* Springer-Verlag, 1986.

FISHMAN, G.S. AND E.P.C. KAO, "A Procedure for Generating Time-Dependent Arrivals for Queueing Simulations," *Naval Research Logistics Quarterly,* 24, 4 (1977), 661-666.

FISHMAN, G.S., *Principles of Discrete Event Simulation,* Wiley-Interscience, 1978.

FOLEY, R.D., "Stationary Poisson Departure Processes from Non-Stationary Queues," *J. of App. Prob.*, 23 (1986), 256-260.

JANG, J.S. AND D.S. BAI, "Efficient Sequential Estimation in a Nonhomogeneous Poisson Process," *IEEE Trans. on Reliability,* R-36, 2 (1987), 255-258.

KAMINSKY, F.C. AND D.L. RUMPH, "Simulating nonstationary Poisson processes: a comparison of alternatives including the correct approach," *Simulation,* 29, 1 (1977), 17-20.

KAO, E.P.C. AND S. CHANG, "Modeling Time-Dependent Arrivals to Service Systems: A Case in Using a Piecewise-Polynomial Rate Function in a Nonhomogeneous

Poisson Process," *Management Sci.*, 34, 11 (1988), 1367-1379.

KLEIN, R.W. AND S.D. ROBERTS, "A time-varying Poisson arrival process generator," *Simulation*, 43, 4 (1984), 193-195.

LAVENBERG, S.S., *Computer Performance Modeling Handbook,* Academic Press, 1983.

LAW, A.M. AND W.D. KELTON, *Simulation Modeling and Analysis,* McGraw-Hill, 1991.

LEE, S., WILSON, J. R., and CRAWFORD, M. M., "Modeling and Simulation of a Nonhomogeneous Poisson Process with Cyclic Features," *Communications in Statistics--Simulation and Computation,* to appear (1991).

LEWIS, P.A.W. AND G.S. SHEDLER, "Simulation of nonhomogeneous Poisson processes with log linear rate function," *Biometrica,* 63, 3 (1976a), 501-505.

LEWIS, P.A.W. AND G.S. SHEDLER, "Statistical Analysis of Non-stationary Series of Events in a Data Base System," *IBM Journal of Research and Development,* 20, 5 (1976b), 465-482.

LEWIS, P.A.W. AND G.S. SHEDLER, "Simulation of Nonhomogeneous Poisson Processes with Degree-Two Exponential Polynomial Rate Function," *Oper. Res.,* 27, 5 (1979a), 1026-1040.

LEWIS, P.A.W. AND G.S. SHEDLER, "Simulation of Nonhomogeneous Poisson Processes by Thinning," *Naval Res. Logist. Quart.,* 26, 3 (1979b), 403-413.

LEWIS, P.A.W. AND E.J. ORAV, *Simulation Methodology for Statisticians, Operations Analysts and Engineers,* Volume I, Wadsworth & Brooks/Cole, 1989.

MORGAN, B.J.T., *Elements of Simulation,* Chapman and Hall, 1984.

NELSON, W., "Graphical Analysis of System Repair Data," *J. of Quality Technology,* 20, 1 (1988), 24-35.

RIGDON, S. E. AND BASU, A. P., "The Power Law Process: A Model for the Reliability of Repairable Systems," *J. of Qual. Tech.,* 21, 4 (1989), 251-260.

RIGDON, S. E. AND BASU, A. P., "The Effect of Assuming a Homogeneous Poisson Process When the True Process is a Power Law Process," *J. of Qual. Tech.,* 22, 2

(1990), 111-117.

ROSS, S.M., *A Course in Simulation*, MacMillan Publishing Company, New York, 1990.

SCHMEISER, B.W., "Random Variate Generation: A Survey," In: Simulation with Discrete Models: A State of the Art View (T. Oren, C. Shub and P. Roth, eds.), 1980 Winter Simulation Conference, IEEE, Piscataway, NJ, 1980, 79-104.

SERFLING, R. J., *Approximation Theorems of Mathematical Statistics*, John Wiley & Sons, New York, 1980.

SHANTHIKUMAR, J., "Uniformization and Hybrid Simulation/Analytic Models of Renewal Processes," *Oper. Res.*, 34, 4 (1986), 573-580.

THORISSON, H., "On Regenerative and Ergodic Properties of the k-server Queue with Non-stationary Poisson Arrivals," *J. of App. Prob.*, 22 (1985), 893-902.

VALLARINO, C.R., "Confidence Bands for a Mean Value Function Estimated from a Sample of Right-Censored Poisson Processes," Technical Report 02.1471, IBM General Products Division, San Jose, California, 1988.

APPENDIX: FORTRAN code for generating a NHPP from the superpositioned event times.

```
dimension x(1002)
      integer n, k
*
*  this program prints event times generated via inversion
*
*  input:
*        n:  the number of events in the superposition (max: 1000)
*        k:  the number of point processes observed
*        T:  the end of the collection period (x(n+2))
*        x:  the n superpositioned event times
*
*  output:
*        the event times in the realization
*        the number of values in the realization
*
      x(1) = 0.0
      read *, n
      if (n.gt.1000) stop
      read *, k
      read *, x(n + 2)
      do 10 i = 1, n
         read *, x(i + 1)
10    continue

      xn = float (n)
      xk = float (k)
      xt = (xn + 1.0) * xk / xn
      igen = 0
      e = 0.0
20    e = e - alog (1.0 - rand (0.0))
         if (e.gt.(xn / xk)) go to 30
         igen = igen + 1
         m = int (e * xt)
         const = (e*xt - float(m))
         t = x(m + 1) + (x(m + 2) - x(m + 1)) * const
         print *, t
         go to 20
30    print *, igen
      stop
      end
```