A Matlab tool for Cox Processes with truncated Gaussian mean

Bouzas, P.R.*
paula@ugr.es
Department of Statistics and Operations Research
University of Granada, 18071-Granada, Spain

Ruiz-Fuentes, N.
nfuentes@ujaen.es
Department of Statistics and Operations Research
University of Jaén, 23071-Jaén, Spain

November 4, 2011

Abstract

The truncated Gaussian distribution rises in many practical situations, the aim of this paper is to give some tools to solve common tasks within this kind of random variables. A modified maximum likelihood estimation of the parameters of the distribution from an observed data set is given, we also implement a goodness-of-fit test for a theoretical truncated Gaussian distribution. Finally, if we assume that the mean of a Cox process at each instant of time is distributed as a truncated Gaussian distribution, we give the most probable value of the process at a given time point.

Keywords: doubly stochastic Poisson process, Cox Process, truncated Gaussian distribution, goodness-of-fit test, modified maximum likelihood estimation.

*This work was partially supported by project MTM2010-20502 of Dirección General de Investigación y Gestión del Plan Nacional I+D+I, Ministerio de Ciencia e Innovación, Spain.

1 Introduction

Many counting Poisson processes should be considered as having mean process modeled as a truncated Gaussian distribution at each time point. This fact is taken into account in Bouzas et al. [1] where a Cox process or doubly stochastic Poisson process with truncated Gaussian mean is considered. Among the results, they derived the expression of the probability mass function of the mean and also stated how to calculate the value of the process with maximum probability at each instant of time. Both results and a modified Kolmogorov-Smirnov test, to asses if an observed set of data fits a truncated Gaussian distribution, are considered in the current paper. We give the tools to solve three items in a suitable way, namely, the estimation of the parameters of a truncated Gaussian distribution from a set of observed data, a goodness-of-fit test to prove a truncated Gaussian source and the estimation of the mode of the truncated Gaussian distribution.

TruncNormMMLE (Bouzas et al. [2]) is a MATLAB function to calculate the modified maximum likelihood estimators of a random variable with truncated Gaussian distribution with truncation points A and B, mean μ and standard deviation σ. This function was used in Bouzas et al.[1] where a Cox process, $N(t)$, is analyzed under the assumption that its mean, $E[N(t)] = \Lambda(t, x(t))$, has a truncated Gaussian distribution at any instant of time.

TruncNormKS (Bouzas et al. [3]) is also a MATLAB function, its aim is to carry out a Kolmogorov-Smirnov goodness-of-fit test for a truncated Gaussian distribution. The Kolmogorov-Smirnov test is modified for the special case in which data are tested to follow not a Gaussian distribution but a truncated one. The theoretical distribution function to compare with the experimental one is derived in the core of the function.

CpTruncNorm (Bouzas et al. [4]) is also inspired in the outcomes in Bouzas et al.[1], it calculates the most probable value of a Cox process with mean a truncated Gaussian distribution. It was proved (Bouzas et al. [1]) that this value, the mode of the Cox process at each time point, is within a bounded interval and can be calculated.

2 TruncNormMMLE, TruncNormKS and CpTruncNorm, syntax and parameters

This section gives the syntax of each proposed function, a description of the parameters to evaluate them in order to solve the three features we are interested in and a brief explanation of their outputs. Appendixes will provide the complete code of the functions.

2.1 Function TruncNormMMLE

The syntax of function `TruncNormMMLE` is given by:

[μ,σ,i,aux]=TruncNormMMLE(A,B,n,μ_0,σ_0,tol)

Let us describe the parameters to run function `TruncNormMMLE` file and to obtain the desired results. It is evaluated in the following set of parameters:

`A` Lower truncation point of the truncated Gaussian distribution.

`B` Upper truncation point of the truncated Gaussian distribution.

`n` Size of the data set.

μ_0 Mean of the sample data.

σ_0 Standard deviation of the sample.

`tol` Sets the maximum error to assume if approximate solutions are given.

The function returns μ and σ, that is, the modified estimations of the mean and the standard deviation of the truncated Gaussian distribution. It also returns the number of iterations until the solution is adopted, `i`. Finally, it returns `aux=1` if an exact solution is obtained or `aux=2` if an approximated solution is given at iteration i or when a maximum of iterations is reached (the maximum is fixed at 40 in the body of the function).

2.2 Function TruncNormKS

The syntax of function `TruncNormKS` is given by:

`[H,p]=TruncNormKS(Sp,`μ`,`σ`,`α`)`

Let us now describe the parameters to introduce in this function. Although the theoretical distribution is truncated Gaussian, only its mean and its standard deviation are required because the function considers as the lower and the upper truncation points, the minimum and maximum observed values, respectively.

`Sp` Insert here the sample data separated by one blank space or by commas, $[x_1\ x_2\ ...\ x_n]$ or $[x_1, x_2, ..., x_n]$.

μ Mean of the truncated Gaussian distribution to be assumed in the null hypothesis.

σ Standard deviation of the truncated Gaussian distribution to be assumed in the null hypothesis.

α Significance level (default = 0.05).

The function returns `H=0` if the hypothetic distribution may be adopted at a significance level α, and returns `H=1` otherwise, as usual in a goodness-of-fit test. The function returns a second value `p`, it is the associated p-value.

2.3 Function CpTruncNorm

Function `CpTruncNorm` has the following syntax:

`[Mo,pr,cdf]=CpTruncNorm(A,B,`μ`,`σ`,`λ`)`

Let us now describe the parameters needed to evaluate this function.

`A` Lower truncation point of the truncated Gaussian distribution.

`B` Upper truncation point of the truncated Gaussian distribution.

μ Mean of the truncated Gaussian distribution.

σ Standard deviation of the truncated Gaussian distribution.

λ The distribution function of the specified truncated Gaussian distribution will be evaluated at this point.

Function `CpTruncNorm` returns `Mo`, that is the most probable value of the Cox process in an instant of time with mean a truncated Gaussian distribution with the specified parameters at this instant. `pr` is the probability of occurrence of the previous most probable value. `cdf` is the value of the distribution function of the truncated Gaussian random variable at λ, this is an additional task solved by the function.

A MATLAB functions codes

The complete code of the functions described in the previous section is given in the following appendixes[1].

A.1 Modified maximum likelihood estimates. TruncNormMMLE

```
function [μ,σ,i,aux]=TruncNormMMLE(A,B,n,μ0,σ0,tol)
  m = 40; aux = 0;
  par=[μ0,σ0,A,B];
  for i=1:m
    μ=par(1);
    σ=par(2);
    F1=G(1,par)/G(0,par)-μ0;
    F2=G(2,par)/G(0,par)-(G(1,par)/G(0,par))^2+2*σ^2/n+σ0^2-1/n;
    F=[F1 F2];
    F11=G(2,par)/(σ^2*G(0,par))-μ*G(1,par)/(σ^2*G(0,par))- ↙
        G(1,par)^2/(σ^2*G(0,par)^2)+μ*G(1,par)/(σ^2*G(0,par));
    F12=G(2,par)/(σ^3*G(0,par))-G(2,par)/(σ^3*G(0,par))- ↙
        G(1,par)*G(2,par)/(σ^3*G(0,par)^2)+ ↙
        2*G(1,par)^2/(σ^3*G(0,par))-μ^2*G(1,par)/(σ^3*G(0,par));
    F21=G(3,par)/(σ^2*G(0,par))-μ*σ^2/(σ^2*G(0,par))- ↙
        G(1,par)*G(2,par)/(σ^2*G(0,par))+ ↙
        μ*G(2,par)/(σ^2*G(0,par))-2*F11;
```

[1]Some lines have been splitted up to adjust the text to the paper size, they must be joined to run the functions. The arrows ↙ at the end of some lines means this fact.

```
F22=G(4,par)/(σ^3*G(0,par))-2*μ*G(3,par)/(σ^3*G(0,par))+
    σ*μ^2*G(2,par)/(σ^3*G(0,par))-G(2,par)^2/
    (σ^3*G(0,par)^2)+2*μ*G(1,par)*G(2,par)/
    (σ^3*G(0,par)^2)-μ^2*G(2,par)/(σ^3*G(0,par))-2*F12;
DF=[F11,F12;F21,F22];
[μ1,σ1]=solve('DF(1,1)*(μ1-μ)+DF(1,2)*(σ1-σ)=F(1,1)',
    'DF(2,1)*(μ1-μ)+DF(2,2)*(σ1-σ)=F(1,2)','μ1,σ1');
par(1)=eval(μ1);par(2)=eval(σ1);μ1=par(1);σ1=par(2);
F1=G(1,par)/G(0,par)-μ0;
F2=G(2,par)/G(0,par)-(G(1,par)/G(0,par))^2+2*σ1^2/n+σ0^2-1/n;
F=[F1 F2];
if F==[0 0],
    aux = 1;
    break;
elseif sqrt((μ1-μ)^2)+sqrt((σ1-σ)^2)<=tol,
    aux = 2;
    break;
end
    μ=μ1;σ=σ1;
    par=[μ,σ,A,B];
end
```

This file needs one external function `G` given below:

```
function g=G(k,par)
    syms x;
    μ=par(1);
    σ=par(2);
    G_=inline('x^k.*exp(-(x-μ)^2/(2*σ^2))','k','μ','σ','x');
    g=eval(int(G_(k,μ,σ,x),par(3),par(4)));
```

A.2 Kolmogorov Smirnov test. TruncNormKS

```
function [H,p]=TruncNormKS(Sample,μ,σ,α)
    if nargin < 4,
        alpha = 0.05;  %(default)
    end
    if nargin < 3,
```

```
        error('Requires at least three input arguments.');
    end
    syms L s
    n = length(Sample);
    Y = sort(Sample);
    A = Y(1);B = Y(n);
    X=[];X = Y(1);N=[1];CDF=[];
    for i=2:n
        if Y(i)~=Y(i-1)
            X=[X,Y(i)];
            N=[N,1];
        else
            N=[N(1:length(N)-1),N(length(N))+1];
            X=[X(1:length(X)-1),Y(i)];
        end
    end
    F=inline(int(exp(-s^2/2),s,(A-μ)/σ,(B-μ)/σ)^-1*
        int(exp(-s^2/2),s,(A-μ)/σ,(L-μ)/σ));
    for i=1:length(X)
        CDF=[CDF;X(i),F(X(i))];
    end
    [H,p] = KSTEST(Y,CDF,α,0);
```

A.3 Maximum probability value. CpTruncNorm

```
function [Mo,pr,cdf]=CpTruncNorm(A,B,μ,σ,λ)
    syms s L u
    k1=(σ*int(exp(-s^2/2),(A-μ)/σ,(B-μ)/σ))^-1;
    f=inline((σ*int(exp(-s^2/2),(A-μ)/σ,(B-μ)/σ))^-1*
        exp((-1/2)*((L-μ)/σ)^2));
    x=[A:(B-A)/100:B];
    plot(x,f(x));title('Density function');

    f1=inline(int(exp(-s^2/2),(A-μ)/σ,(u-μ)/σ));
    cdf=f1(λ)/f1(B);
    k2=(1/sqrt(2*π))*f1(B);
    P=[];
    for n=floor(A):floor(B)
```

```
    p=inline(int(s^n*exp(-s)*exp(-(1/2)*(s-μ)^2/σ^2),A,B));
    P=[P;(1/factorial(n))*k2^-1*(1/sqrt(2*π*σ^2)*p(n)];
end
[pr,i]=max(P);
Mo=floor(A)+i;
```

References

[1] P.R. Bouzas, A.M. Aguilera and Valderrama, M.J., Forecasting a class of doubly stochastic Poisson process, *Statistical Papers*, 43, 507–523, 2002.

[2] P.R. Bouzas and N. Ruiz-Fuentes, TruncNormMMLE Matlab file, v.1.0.2011, *Free access DIGIBUG*.

[3] P.R. Bouzas and N. Ruiz-Fuentes, TruncNormKS Matlab file, v.1.0.2011, *Free access DIGIBUG*.

[4] P.R. Bouzas and N. Ruiz-Fuentes, CpTruncNorm Matlab file, v.1.0.2011, *Free access DIGIBUG*.