ESTIMATING DISTRIBUTION FUNCTIONS WITH THE SAS® SYSTEM

author: Werner Bundschuh, University of Heidelberg, Rechenzentrum

coauthor: Joachim Hefner, now SAS Institute

Abstract

In this poster we are showing how to estimate and construct confidence intervals and confidence bands for the cumulative distribution function. We are offering two solutions: the first one is a classical approach using asymptotic results, the second one is using the Bootstrap method. Furthermore we have written a "procedure macro" for density estimation. We have integrated a FORTRAN subroutine (subroutine DDESKN of the IMSL program library) which computes kernel estimates for the density function.

Introduction

A simple but common need arises in data analysis when we have a single set of numbers, and we want to understand their basic characteristics as a collection of data. Graphical methods described in the book of Chambers and Tukey are now implemented in some software packages (e.g. in JMP running on Macintosh - JMP is a software package for statistical graphics developed by SAS Institute). Beyond this we are interested to estimate the density function and the cumulative density function (CDF). In data analysis we could be interested in confidence intervals and confidence bands for the CDF, too. When having collected data to analyze we normally don’t know anything about the underlying CDF F. So we cannot build a parametric model to estimate model parameters. A natural nonparametric estimate of the probability \( F(x) \) is defined by:

\[
\hat{F}_n(x) = \frac{\text{number of } X_i \leq x}{n}
\]

\( \hat{F}_n \) is called the "empirical cumulative density function" (ECDF) of the sample \( X_1, \ldots, X_n \). For demonstration purposes we have chosen a data set which consists of the elapsed time above a certain high level for a series of 66 wave records at San Francisco bay. The examples show the ECDF for this wave data. This plot reveal that about 50 percent of the waves are above the level criteria for about 4 time units or less.

Looking at the estimate only we don’t have any information about the quality of the estimate. To circumvent this problem we are constructing confidence intervals locally and confidence bands globally. Local confidence intervals can be easily computed for all situations using Normal approximations. However the classical asymptotic method for constructing confidence bands has the assumption of a continuous CDF. In other cases this method is not valid. But there is another method solving this problem: The Bootstrap method.

In the case of a smooth CDF - the first derivative exists and is continuous - we want to study the distributional shape of our population. So we try to estimate the density function. In the last years the kernel estimation technique has become an important tool in curve fitting. Therefore we tried to integrate such a method in the SAS System. The IMSL program library contains the FORTRAN subroutine DDESKN
for kernel estimation of the density function. Using the version 5 interface to FORTRAN functions we developed a "procedure macro". The user of this macro can bring special parameters necessary for the kernel estimation to the subroutine DDDESKN. These parameters - bandwidth, kernel and data window - will heavily influence the estimation results. We will show some examples about that subject and give hints how to get good results. A similar implementation of this "procedure macro" should be possible in version 6, too.

Confidence Intervals for the CDF
The statistical aspects of \( \hat{F}_n \) are well known. Here are some important asymptotic results which can be used to construct confidence intervals for the CDF: \( \hat{F}_n(.) \) is a consistent estimate of \( F(.) \):

\[ \hat{F}_n(x) \to_p F(x) \text{ for each } x \]

\( \hat{F}_n(x) \) is asymptotically normal for each \( x \):

\[ P \left( \frac{\sqrt{n}(\hat{F}_n(x) - F(x))}{\sqrt{F(x)(1-F(x))}} \leq \lambda \right) \to_{n \to \infty} \Phi(\lambda) \]

Where \( \Phi \) is the Standard Normal CDF.

Using this results we are able to construct an asymptotic confidence interval for \( F(x) \):

\[ \lim_{n \to \infty} P \left( \hat{F}_n(x) - \sqrt{n}\frac{\hat{F}_n(x)(1-F(x))}{\sqrt{F(x)(1-F(x))}} \leq F(x) \leq \hat{F}_n(x) + \sqrt{n}\frac{\hat{F}_n(x)(1-F(x))}{\sqrt{F(x)(1-F(x))}} \right) = 1 - \Phi(\lambda_n) = 1 - \alpha \]

In this formula \( \lambda_n \) is the \( \alpha \)-quantile of the Standard Normal distribution. To get a 95\% confidence interval you have to choose \( \lambda = 1.96 \). The value 1.96 is the 97.5\% quantile of the Standard Normal distribution.

Confidence Bands for the CDF: Classical Method
Furthermore there are some global results for the ECDF. An important one is concerning the Kolmogorov statistic \( D_n \) for testing the hypothesis \( F = F_0 \):

\[ D_n = \sup_{-\infty < x < \infty} \sqrt{n} |\hat{F}_n(x) - F_0| \]

Hint: With \( F_0(.) = \Phi \left( \frac{z}{s} \right) \) we get the Kolmogorov statistic for testing normality which is used in SAS procedure UNIVARIATE.

The process \( D_n \) is well studied and the limiting process (Brownian Bridge process) is well known. Therefore we can construct confidence bands for the CDF:

\[ \lim_{n \to \infty} P \left( \hat{F}_n(x) - \frac{\lambda_{n,n}}{\sqrt{n}} \leq F(x) \leq \hat{F}_n(x) + \frac{\lambda_{n,n}}{\sqrt{n}}, \text{ for all } x \in (-\infty, \infty) \right) = 1 - \alpha \]

\( \lambda_{n,n} \) is the \( \alpha \)-quantile of the limiting process depending on \( n \) because of the continuity correction - s. [2].
Confidence Bands for the CDF: Bootstrap Method

The classical asymptotic construction of fixed width confidence bands for $F$ shown above is feasible only when $F$ is assumed continuous. One alternative method is bootstrap and the bootstrap method is valid whether or not $F$ is continuous.

To construct a bootstrap confidence band for $F$ we have to estimate the CDF of the Kolmogorov statistic $D_n$. We are doing this by computing $m$ bootstrap samples from the original sample. For each bootstrap sample we get a bootstrap Kolmogorov statistic:

$$D_n^i = \sup_{-\infty < x < \infty} \sqrt{n} |F^*_n(x) - \hat{F}_n(x)|, \; i = 1, \ldots, m$$

where $F^*_n$ is the ECDF of the $i$th bootstrap sample. The critical value for the bootstrap confidence band can be obtained by calculating the $(1 - \alpha)$-quantile $\lambda_{n,n}^b$ of the sample $D_1^b, \ldots, D_m^b$. The $(1 - \alpha)$ confidence band now looks like follows:

$$\lim_{n \to \infty} P \left( \frac{\lambda_{n,n}^b}{\sqrt{n}} \leq \frac{F(x) - \hat{F}_n(x)}{\sqrt{n}}, \text{ for all } x \in (-\infty, \infty) \right) = 1 - \alpha$$

Kernel Estimation of the Density Function

When we are interested in the shape of the distribution we usually look at the density function $f$ of $F$. The simplest way to get a non-parametric density estimate is to smooth histograms. However this is a somewhat heuristic solution.

A relative new and thoroughly studied method for density approximation is the kernel estimation. The kernel estimate $\hat{f}_n(t)$ for $f$ smooths the "density" of the ECDF $\hat{F}_n$:

$$\hat{f}_n(t) = \int_{-\infty}^{\infty} \frac{1}{hn} K \left( \frac{t - x}{hn} \right) d\hat{F}_n(x) = \frac{1}{nhn} \sum_{i=1}^{n} K \left( \frac{t - x^i}{hn} \right)$$

Here $hn$ is called the bandwidth and $K$ is the kernel function normally being a density function with finite or infinite (e.g. Normal density) support. $K$ determines the influence of the data points on the estimation. $hn$ is strongly connected with the smoothness of the estimate.

The asymptotic properties of kernel estimation has been published in many papers and are beyond this paper because of the special notation necessary.

In the finite sample case the estimate is heavily influenced by the bandwidth. The choose of the kernel function is not so important. But a special high bandwidth depending on $n$ and the range of your data will give you an extreme smooth estimate vealing interesting information. On the other hand a special low value of $hn$ will show a noisy estimate. We will demonstrate these properties in the examples.
Notes about Realisation with the SAS System

To simplify the use of the programs for estimating distribution functions we have written SAS macros. The SAS macros CDFCON and CDFBCON will work under SAS version 6. There are necessary slight modifications for CDFCON to run under SAS version 5. In the moment the SAS macro KDENS is running only under SAS version 5. We think to get it work under version 6 when SAS Institute will integrate the FORTRAN interface in version 6, too.

For the construction of confidence intervals and confidence bands derived from the asymptotic properties of the ECDF we worked out the SAS macro CDFCON. In this macro the ECDF can be estimated by the SAS procedure RANK. The critical value can be calculated for the interval case just by using SAS function probit which is the inverse normal distribution function. If the macro user wishes to construct simultaneous confidence bands he just has two alternatives for the confidence level: 0.05 or 0.01. The corresponding critical values are implemented in the data step. The SAS macro CDFCON has the following parameters:

VAR= specify the variable to be analyzed;
      this is an necessary option
DATA= the data set to be used; default is _LAST_.
OUT=XRANK specify the name of the output data set; default is XRANK
TITLE= the TITLE text; default is none
ALPHA=0.05 the confidence level ;default is 0.05
WAY=GLOBAL do you want confidence intervals - specify LOCAL;
       in the other case use the default GLOBAL

The examples of the classical case are both of 95% confidence. The confidence intervals are much narrower than the confidence bands. But this is to be expected because the intervals can only locally be interpreted for a fixed x while the confidence bands are valid for all x simultaneously.

In the bootstrap case we developed a SAS macro, named CDFBCON. There the bootstrap samples can be obtained by resampling with replacement (s. [6]) in refer to the ECDF. The ECDF's of the original sample and the bootstrap samples can be calculated by the SAS procedure RANK, too. The empirical α quantile of the Kolmogorov statistics $D_1^m$, $D_m^m$ can be derived in a data step and then be used to construct the confidence bands. The SAS macro CDFBCON has the following parameters:

VAR= specify the variable to be analyzed;
      this is an necessary option
DATA= the data set to be used; default is _LAST_.
OUT=XRANK specify the name of the output data set; default is XRANK
M=200 the number of bootstrap samples; default is 200
ALPHA=0.05 the confidence level ;default is 0.05

The confidence band in the bootstrap case is slightly wider than in the classical case. This should
become clear because the bootstrap method is valid for a greater class of distribution functions.

For some time we had the idea to implement the IMSL FORTRAN subroutine DDESKN for kernel estimation of the density function in the SAS system. Therefore we have written a SAS function DICHTE. This function is our "bridge" to the subroutine DDESKN. The data to be analyzed are read in by the SAS function DICHTE from a temporary file and are passed to the subroutine DDESKN. The parameters necessary for the density estimation (e.g. the kernel type or the bandwidth) are the arguments of the SAS function DICHTE. The SAS function DICHTE transfers the necessary parameters and the data vector which was read before. At the end the results of the density function are written back to the temporary file (see the diagrams Flow Chart and Sketch of the SAS Function DICHTE below).

To simplify the use of the "density program" we have written the SAS macro KDENS which performs the data exchange with the function in real binary format (SAS Format RB8.). The parameters of the macro are the arguments of the function DICHTE (see the diagrams Sketch of the SAS Macro KDENS and the macro code below). Note that the SAS function DICHTE has to be called only once. The macro KDENS has the following parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VAR</td>
<td>specify the variable to be analyzed; the number of observation must be less than 100000 this parameter must be specified</td>
</tr>
<tr>
<td>DATA=</td>
<td>the data set to be used; default is _LAST.</td>
</tr>
<tr>
<td>OUT=XXXX4</td>
<td>specify the name of the output data set; default is XXXX4</td>
</tr>
<tr>
<td>WIN=1</td>
<td>the bandwidth for the kernel function</td>
</tr>
<tr>
<td>XMAX=-1</td>
<td>Cutoff value for the kernel; there is no cutoff by default: XMAX=-1</td>
</tr>
<tr>
<td>NXPT=20</td>
<td>number of points at which a density estimated is desired the range of data is divided in NXPT equal spaced intervals</td>
</tr>
<tr>
<td>KERN=2</td>
<td>you can choose the kernel function; there are implemented five kernel functions;</td>
</tr>
<tr>
<td>PLOT=0</td>
<td>specify not 0 if you wish a plot</td>
</tr>
</tbody>
</table>

The four examples shown above demonstrate the heavy influence of the bandwidth on the smoothness of the estimate. For bandwidth 0.5 we see a very noisy function. On the other side a bandwidth of 2 extremly smoothes the curve. So a bandwidth of 1 could be the best possible. The demonstration shows that we should try several estimates with different bandwidths.

Hint: We have chosen the triangle kernel \( K(x) = 1 - |x|, |x| \leq 1 \) for the examples.

The dashed lines in the plots demonstrate the density of the Standard Normal distribution with the same mean value and variance as the sample. The density estimates show a departure from normality. A possible distribution class with similar shape of the densities is the family of Gamma distributions.
The SAS Macro CDFCON

%macro CDFCON(VAR=,DATA=_LAST,OUT=XRANK,ALPHA=0.05,TITLE=, WAY=GLOBAL);
%local stop;
options nonotes;
proc rank data=&data f out=&out;
ranks cumf; var &var;
proc sort data=&out; by &var;

%if %upcase(&way)=LOCAL %then %do;
data &out;set &out nobs=n;
retain c;
If _n_=1 then c=probit(1-&alpha/2)/sqrt(n);
low=max(0,cumf-c*sqrt(cumf*(1-cumf)));
high=min(1,cumf+c*sqrt(cumf*(1-cumf)));
run;
%end;
%else %if %upcase(&way)=GLOBAL %then %do;
data &out;set tout nobs=n;
retain c 0;
If _n_=1 and &alpha=.05 then c=1.358/(sqrt(n)+.12+.11/sqrt(n));
else if _n_=1 and &alpha=.01 then c=1.628/(sqrt(n)+.12+.11/sqrt(n));
else if _n_=1 then do;
call symsput('Stop','STOP');
stop;end;
low=max(0,cumf-c);
high=min(1,cumf+C);
run;
%end;
%else %goto v;
%if &stop=STOP %then %goto AL;
proc gplot;
plot (low cumf high)*&var/overlay fr;
symbol1 v=none i=steplj c=white l=1;
symbol2 v=none i=steplj c=white l=2;
symbol3 v=none i=steplj c=white l=1;
Title &title;
run;
%goto ende;
%w: %put ERROR: WAY=&WAY is NOT defined; %GOTO ende;
%AL: %put ERROR: ALPHA=&ALPHA is not possible for WAY=GLOBAL;
%ende;
%mend CDFCON;
The SAS Macro CDFBCON

%MACRO CDFBCON(VAR,DATA=_LAST_,N=200,ALPHA=0.05,OUT=XRANK);

proc rank data=&data f ties=high out=sample;
   var &var;
   ranks fn;

proc sort; by &var;

data bsample; drop i;
   do bsample=1 to &m;
      do i=1 to n;
         iobs = int(ranuni(0) * n) + 1;
         set sample(keep=&var) point=iobs nobs=n; output;
      end;
   end;
   call symput('stpn', left(n));
stop;

proc sort; by bsample &var;

proc rank f ties=high out=bsample;
   var &var;
   ranks fnhat;
   by bsample;

proc sort nodup; by bsample &var;

data helpsamp;
   set sample;
   do bsample=1 to &m; output; end;

proc sort; by bsample &var;

data bsample;
   merge bsample helpsamp; by bsample &var;
   if first.bsample then help=0;
   if fnhat ne . then help = fnhat;
   retain help;
   fnhat=help;
   drop help;

data bsample;
  set bsample;
  dfn=abs(fnhat-fn);
  keep dfn bsample;

proc summary;
  var dfn;
  by bsample;
  output out=bsample(drop=_type_ _freq_) max=fnstar;

proc sort; by fnstar;

data _null_; 
  i = max( floor( &m * (1 - &alpha)), 1 );
  set bsample point = i;
  call symput ('bound',fnstar);
  stop;

data &out;
  set sample(keep=&var fn);
  lower = max(0, fn - &bound );
  upper = min(1, fn + &bound );

proc gplot data=&out;
  plot (fn lower upper) * &var / vaxis=axis1 overlay;
  axis1 order=0 to 1 by 0.2 label=(f=simplex c=blue 'P');
  symbol1 l=1 i=steplj value=none c=black;
  symbol2 l=2 i=steplj value=none c=black;
  symbol3 l=1 i=steplj value=none c=black;
%mend;
The SAS Macro KDENS

%MACRO KDENS(VAR,
    DATEI=_LAST_,WIN=1,XMAX=-1,NXPT=20,KERN=3,OUT=xxx4,PLOT=0);
%* VERSION 2.0 OF KDENS.
COPYRIGHT - W. BUNDSCHUH, J. HEFNER;
OPTIONS NOSOURCE NPRINT NONOTES;
*;
* CHECK THE PARAMETERS
*;
%LET A=1;
DATA XXX0; SET &DATEI END=EOF;
    IF &VAR ^= . THEN DO; N+1;OUTPUT;END;
    IF EOF THEN CALL SYMPUT('XXXXN',N);
    KEEP &VAR;
RUN;
PROC SORT DATA=xxx0 OUT=XXX1(KEEP= &VAR);
   BY &VAR;
PROC CONTENTS DATA=XXX1 NOSOURCE NOPRINT OUT=XXX2(KEEP=TYPE NOBS);
DATA XXX2; SET XXX2;
    IF TYPE = 2 THEN DO;
        PUT "ERROR: DIE VARIABLE &VAR IST KEINE NUMERISCHE VARIABLE";
        END;
        CALL SYMPUT('YTYPE',TYPE);
    DATA XXX3; SET XXX2(KEEP=NOBS);
    V_WIN = SYMGET('WIN') + 0;
    V_XMAX = SYMGET('XMAX') + 0;
    V_KERN = SYMGET('KERN') + 0;
    V_NXPT = SYMGET('NXPT') + 0;
    ERR = _ERROR_
    CALL SYMPUT('CID',ERR);
    IF _ERROR_ = 1 THEN DO;
        PUT "ERROR: DIE UEBERGABEPARAMETER WIN,XMAX,KERN UND NXPT "
            "MUSSSEN NUMERISCH SEIN";
        END;
*;
* CHECK THE PARAMETERS
*;
DATA XXX3; SET XXX3;
    IF NOBS > 10000 THEN DO;
        PUT "ERROR: ES SIND MAXIMAL 10000 BEOBUCHTUNGEN ZUGELASSEN";
        END;
        CALL SYMPUT('Y_N',NOBS);
    IF V_NXPT > 1000 THEN DO;}
Bundschuh, W., Hefner, J.

ESTIMATING DISTRIBUTION FUNCTIONS WITH THE SAS SYSTEM

PUT "ERROR: ES SIND MAXIMAL 1000 STUETZSTELLEN ZUGELASSEN";
END;
IF V_WIN < 0 THEN DO;
  PUT "ERROR: DIE FENSTERBREITE MUSS >= 0 SEIN";
END;
IF NOT(V_KERN=1 | V_KERN=2 | V_KERN=3 | V_KERN=4 | V_KERN=5)
  THEN DO;
  PUT "ERROR: DIE ZULAESSIGEN WERTE FUER KERN SIND 1,2,3,4,5";
END;
RUN;

%IF &YTYPE = 2 %THEN %GOTO ABBRUCH;
%IF &CID = 1 %THEN %GOTO ABBRUCH;
%IF &Y_N > 10000 %THEN %GOTO ABBRUCH;
%IF &NXPT > 1000 %THEN %GOTO ABBRUCH;
%IF &WIN < 0 %THEN %GOTO ABBRUCH;
%IF NOT( &KERN = 1 | &KERN = 2 | &KERN = 3 | &KERN = 4 | &KERN = 5 ) %THEN %GOTO ABBRUCH;

*;
* WRITING THE DATA IN THE TEMPORARY FILE;
*;
DATA _NULL_; SET XXX1;
  FILE FT16F001;
  PUT &VAR RB8. ;
*;
* CALL THE FUNCTION;
*;
DATA _NULL_; SET XXX3;
  CALL DICHTE(V_WIN,V_XMAX,V_NXPT,NOBS,V_KERN);
*;
* READ THE RESULTS ;
*;
DATA &OUT; INFILE FT16F001;
  INPUT &VAR RB8. / FIT RB8. ;
*
* SHOULD WE PLOT?
*;
%if &plot ne 0 %then %do;
PROC FREQ DATA=XXX0 ; TABLES &VAR/ OUT=XXXF NOPRINT;
DATA XXXF;SET XXXF; XXXF=+.1+(COUNT/XXXN)*.1;
%DCLANNO;
%SYSTEM(2,2,4);
ESTIMATING DISTRIBUTION FUNCTIONS WITH THE SAS SYSTEM
Bundschuh, W., Hefner, J.

%LINE(&VAR,-.1,&VAR,XXXXF,BLACK,1,.2);
PROC MEANS DATA=XXX1 NOPRINT; VAR &VAR;
OUTPUT OUT=XXXXP MEAN=XXXXM STD=XXXXS;
DATA XXXXP;SET XXXXP;
SET &OUT END=END;
   XXXXP=EXP(-.5*((&VAR-XXXXM)/XXXXS)**2)/(SQRT(6.28)*XXXXS);
OUTPUT;
DO WHILE(END=0); SET &OUT END=END;
   XXXXP=EXP(-.5*((&VAR-XXXXM)/XXXXS)**2)/(SQRT(6.28)*XXXXS);
OUTPUT;END;
PROC MEANS NOPRINT DATA=XXXXP; OUTPUT OUT=XXXXM MAX=MAXF MAXM; VAR FIT
   XXXXP;
DATA XXXXM;SET XXXXM;
max=max(maxf,maxm);
x=log10(max);
if x<0 then do;
x=int(-x)+1;
MAX =((INT(MAX*10**x)+1)/10**x;
end;
else MAX=((INT(MAX*10)+1)/10;
   XXXXB=MAX/5; CALL SYMPUT('XXXXB',XXXXB); CALL SYMPUT('MAX',MAX);
RUN;
*
* PLOT OF THE DENSITY WITH SPLINE INTERPOLATION;
*
PROC GPLOT DATA=XXXXP;
   PLOT FIT * &VAR= 1 XXXXP* &VAR=2/VREF=O LVREF=2 ANNO=XXXXF
   OVERLAY HAXIS = AXIS1 VAXIS = AXIS2;
SYMBOL1 V = none I = SPLINE C=BLACK;
SYMBOL2 V=NONE I=SPLINE C=BLACK L=2;
AXIS1 label = none minor = none;
AXIS2 LABEL = (F=SIMPLEX 'FIT') MINOR = NONE VALUE=(F=SIMPLEX T=1 H=0)
ORDER=-.1 0 TO &MAX BY &XXXXB;
   title f=swiss1 "kernel estimate of &var";
RUN;
*DELETE TEMPORARY WORK FILES;
Proc datasets nolist nofs ddname=work; delete xxxxp xxxxm xxxxf;
%END;
Proc datasets nofs nolist ddname=work; delete xxx0 xxx1 xxx2 xxx3 ;
%ABRUCH:
ESTIMATING DISTRIBUTION FUNCTIONS WITH THE SAS SYSTEM
Bundschuh, W., Hefner, J.

OPTIONS SOURCE NOTES;
RUN;
%MEND KDENS;

author address: Werner Bundschuh
Universitaet Heidelberg
Rechenzentrum
Im Neuenheimer Feld 293
6900 Heidelberg

Electronic Mail (Bitnet): X16 at DHDURZ1

585
ESTIMATING DISTRIBUTION FUNCTIONS WITH THE SAS SYSTEM
Bundschuh, W., Hefner, J.

Classical Confidence Intervals

Classical Confidence Bands

Bootstrap Confidence Bands
ESTIMATING DISTRIBUTION FUNCTIONS WITH THE SAS SYSTEM
Bundschuh, W., Hefner, J.

kernel estimate with bandwidth 0.5  kernel estimate with bandwidth 1.0

kernel estimate with bandwidth 1.5  kernel estimate with bandwidth 2.0
Flow Chart

%MACRO name(parameter);
  syntax check
  logical check
  data transfer
  CALL fortran;
  Read results
  %MEND name;

Sketch of the MACRO KDENS

%MACRO KDENS(parameter);
  syntax check
  logical check
  transfer of data to temporary file
  DATA _NULL_; call the FORTRAN function
  CALL DICHTE(arguments);
  read the data from temporary file
  further code if necessary
  %MEND KDENS;

Sketch of the SAS Function DICHTE

read the data

call DDESBNK

Write the results

There is no printer out of the SAS function
References


