

# Package: LPsmooth (via r-universe)

October 27, 2024

**Type** Package

**Title** LP Smoothed Inference and Graphics

**Version** 0.1.3

**Author** Xiangyu Zhang <zhan6004@umn.edu>, Sara Algeri <salgeri@umn.edu>

**Maintainer** Xiangyu Zhang <zhan6004@umn.edu>

**Description** Classical tests of goodness-of-fit aim to validate the conformity of a postulated model to the data under study. In their standard formulation, however, they do not allow exploring how the hypothesized model deviates from the truth nor do they provide any insight into how the rejected model could be improved to better fit the data. To overcome these shortcomings, we establish a comprehensive framework for goodness-of-fit which naturally integrates modeling, estimation, inference and graphics. In this package, the deviance tests and comparison density plots are performed to conduct the LP smoothed inference, where the letter L denotes nonparametric methods based on quantiles and P stands for polynomials. Simulations methods are used to perform variance estimation, inference and post-selection adjustments. Algeri S. and Zhang X. (2020) <[arXiv:2005.13011](https://arxiv.org/abs/2005.13011)>.

**Imports** LPGraph,LPBkg,truncnorm,nloptr,Hmisc,orthopolynom,polynom

**License** GPL-3

**LazyData** True

**NeedsCompilation** no

**Date/Publication** 2020-11-24 11:30:05 UTC

**Repository** <https://xiangyu2022.r-universe.dev>

**RemoteUrl** <https://github.com/cran/LPsmooth>

**RemoteRef** HEAD

**RemoteSha** 14f21c681e7c92b816f26f6c8e9be015d3ca84dd

## Contents

CDplot . . . . .	2
dmixnegbinom . . . . .	4
dmixtruncnorm . . . . .	5
d_hat . . . . .	6
find_h_cont . . . . .	7
find_h_disc . . . . .	9
rmixnegbinom . . . . .	11
rmixtruncnorm . . . . .	12
<b>Index</b>	<b>14</b>

---

CDplot	<i>CD-plot and Deviance test</i>
--------	----------------------------------

---

### Description

Constructs the CD-plot and computes the deviance test for exhaustive goodness-of-fit.

### Usage

```
CDplot(data,m=4,g,par0=NULL,range=NULL,lattice=NULL,selection=TRUE,criterion="BIC",
        B=1000,samplerG=NULL,h=NULL,samplerH=NULL,R=500,ylim=c(0,2),CD.plot=TRUE)
```

### Arguments

data	A data vector. See details.
m	If selection = FALSE, it corresponds to the desired size of the polynomial basis to be used. If selection = TRUE, it is the size of the polynomial basis from which the terms to include in the model are selected.
g	Function corresponding to the parametric start. See details.
par0	A vector of starting values for the parameters of g when the latter is not fully known. See details.
range	Interval corresponding to the support of the continuous data distribution.
lattice	Support of the discrete data distribution.
selection	A logical argument indicating if model selection should be performed. See details.
criterion	If selection=TRUE, the selection criterion to be. The two possibilities are "AIC" or "BIC". See details.
B	A positive integer corresponding to the number of bootstrap replicates.
samplerG	A function corresponding to the random sampler for the parametric start g. See details.
h	Instrumental probability function. If samplerG is not NULL, the argument h will not be used.

samplerH	A function corresponding to the random sampler for the instrumental probability function $h$ . If samplerG is not NULL, the argument samplerH will not be used.
R	A positive integer corresponding to the size of the grid of equidistant points at which the comparison densities are evaluated. The default is $R = 500$ , a larger value may be needed when the smoothness of the comparison densities decrease.
ylim	If check.plot=TRUE, the range of the y-axis of the respective comparison density plot. The default is $c(0, 2)$ .
CD.plot	A logical argument indicating if the comparison density plot should be displayed or not. The default is TRUE.

### Details

The argument `data` collects the data for which we want to test if its distribution corresponds to the one of the postulated model specified in the argument `g`. If the parametric start is fully known, it must be specified in a way that it takes  $x$  as the only argument. If the parametric start is not fully known, it must be specified in a way that it takes arguments  $x$  and `par`, with `par` corresponding to the vector of unknown parameters. The latter are estimated numerically via maximum likelihood estimation and `par0` specifies the initial values of the parameters to be used in the optimization. The value `m` determines the smoothness of the estimated comparison density, with smaller values of `m` leading to smoother estimates. If `selection=TRUE`, the largest coefficient estimates are selected according to either the AIC or BIC criterion as described in Algeri and Zhang, 2020 (see also Ledwina, 1994 and Mukhopadhyay, 2017). The resulting estimator is the one in Gajek's formulation with orthonormal basis corresponding to LP score functions (see Algeri and Zhang, 2020 and Gajek, 1986).

### Value

Deviance	Value of the deviance test statistic.
p_value	P-value of the deviance test.

### Author(s)

Sara Algeri and Xiangyu Zhang

### References

- Algeri S. and Zhang X. (2020). Exhaustive goodness-of-fit via smoothed inference and graphics. arXiv:2005.13011.
- Gajek, L. (1986). On improving density estimators which are not bona fide functions. *The Annals of sStatistics*, 14(4):1612–1618.
- Ledwina, T. (1994). Data-driven version of neyman's smooth test of fit. *Journal of the American Statistical Association*, 89(427):1000–1005.
- Mukhopadhyay, S. (2017). Large-scale mode identification and data-driven sciences. *Electronic Journal of Statistics* 11 (2017), no. 1, 215–240.

### See Also

[d\\_hat](#), [find\\_h\\_disc](#), [find\\_h\\_cont](#).

**Examples**

```

data<-rbinom(50,size=20,prob=0.5)
g<-function(x)dpois(x,10)/(ppois(20,10)-ppois(0,10))
samplerG<-function(n){xx<-rpois(n*3,10)
                      xxx<-sample(xx[xx<=20],n)
                      return(xxx)}
CDplot(data,m=4,g,par0=NULL,range=NULL,lattice=seq(0,20),
        selection=FALSE,criterion="BIC",B=10,samplerG,R=300,ylim=c(0,2))

```

dmixnegbinom

*Probability mass function of a mixture of negative binomials***Description**

Computes the probability mass function of a mixture of the negative binomials.

**Usage**

```
dmixnegbinom(x, pis, size, probs)
```

**Arguments**

x	A scalar or vector of non-negative integer values.
pis	A vector collecting the mixture weights. See details.
size	A positive value corresponding to the target for number of successful trials.
probs	A vector collecting the probabilities of success for each mixture component.

**Details**

The argument `pis` is a vector with length equal the number of components in the mixture. The vector `pis` must sum up to one, e.g. `c(0.7, 0.2, 0.1)`. All the negative binomials contributing to the mixture are assumed to have the same size.

**Value**

Value of the probability mass function of the mixture of negative binomials evaluated at `x`.

**Author(s)**

Xiangyu Zhang and Sara Algeri

**See Also**

[rmixtruncnorm](#), [dmixtruncnorm](#), [rmixnegbinom](#), [find\\_h\\_disc](#).

**Examples**

```
xx<-seq(0,30,length=31)
dmixnegbinom(xx,pis=c(0.4,0.6),size=25,probs=c(0.6,0.7))
```

---

dmixtruncnorm	<i>Probability density function of a mixture of truncated normals</i>
---------------	---

---

**Description**

Computes the probability density function of a mixture of truncated normals.

**Usage**

```
dmixtruncnorm(x, pis, means, sds, range)
```

**Arguments**

x	A scalar or vector of real values.
pis	A vector collecting the mixture weights. See details.
means	A vector collecting the means of the mixture components.
sds	A vector collecting the standard deviations of the mixture components.
range	Interval corresponding to the support of each of truncated normal contributing to the mixture. See details.

**Details**

The argument `pis` is a vector with its length equal the number of components in the mixture. The vector `pis` must sum up to one, e.g. `c(0.7, 0.2, 0.1)`. The argument `range` is an interval corresponding to the support of each truncated normal contributing to the mixture. In other words, all the truncated normals contributing to the mixture are assumed to have the same range.

**Value**

Value of the probability density function of the mixture of normals evaluated at `x`.

**Author(s)**

Sara Algeri and Xiangyu Zhang

**See Also**

[rmixtruncnorm](#), [dmixnegbinom](#), [rmixnegbinom](#), [find\\_h\\_cont](#)

**Examples**

```
xx<-seq(0,30,length=10)
dmixtruncnorm(xx,pis=c(0.3,0.6,0.1),means=c(3,6,25),sds=c(3,4,10),range=c(0,30))
```

---

d\_hat *Comparison density estimate*

---

### Description

Estimates the comparison density for continuous and discrete data.

### Usage

```
d_hat(data,m=4,g,range=NULL,lattice=NULL,selection=TRUE,criterion="BIC")
```

### Arguments

data	A data vector. See details.
m	If selection = FALSE, it corresponds to the desired size of the polynomial basis to be used. If selection = TRUE, it is the size of the polynomial basis from which the terms to include in the model are selected.
g	Function corresponding to the parametric start. See details.
range	Interval corresponding to the support of the continuous data distribution.
lattice	Support of the discrete data distribution.
selection	A logical argument indicating if model selection should be performed. See details.
criterion	If selection=TRUE, the selection criterion to be used. The two possibilities are "AIC" or "BIC". See details.

### Details

The argument data collects the data for which we want to test if its distribution corresponds to the one of the postulated model specified in the argument g. The parametric start is assumed to be fully specified and takes x as the only argument. The value m determines the smoothness of the estimated comparison density, with smaller values of m leading to smoother estimates. If selection=TRUE, the largest coefficient estimates are selected according to either the AIC or BIC criterion as described in Algeri and Zhang, 2020 (see also Ledwina, 1994 and Mukhopadhyay, 2017). The resulting estimator is the one in Gajek's formulation with orthonormal basis corresponding to LP score functions (see Algeri and Zhang, 2020 and Gajek, 1986).

### Value

LPj	Estimates of the coefficients.
du	Function corresponding to the estimated comparison density in the u domain corresponding to the probability integral transformation.
dx	Function corresponding to the estimated comparison density in the x domain.
f	Function corresponding to the estimated probability function of the data.

**Author(s)**

Sara Algeri and Xiangyu Zhang

**References**

- Algeri S. and Zhang X. (2020). Exhaustive goodness-of-fit via smoothed inference and graphics. arXiv:2005.13011.
- Gajek, L. (1986). On improving density estimators which are not bona fide functions. The Annals of sStatistics, 14(4):1612–1618.
- Ledwina, T. (1994). Data-driven version of neyman’s smooth test of fit. Journal of the American Statistical Association, 89(427):1000–1005.
- Mukhopadhyay, S. (2017). Large-scale mode identification and data-driven sciences. Electronic Journal of Statistics 11 (2017), no. 1, 215–240.

**See Also**

[CDplot](#)

**Examples**

```
library("LPBkg")
#Example discrete
data<-rbinom(1000, size=20, prob=0.5)
g<-function(x)dpois(x,10)/(ppois(20,10)-ppois(0,10))
ddhat<-d_hat(data,m=4,g, range=NULL,lattice=seq(0,20), selection=TRUE,criterion="BIC")
xx<-seq(0,20)
ddhat$dx(xx)
ddhat$LPj

#Example continuous
data<-rnorm(1000,0,1)
g<-function(x)dt(x,10)
ddhat<-d_hat(data,m=4,g, range=c(-100,100), selection=TRUE,criterion="AIC")
uu<-seq(0,1,length=10)
ddhat$du(uu)
ddhat$LPj
```

---

find\_h\_cont

*Finding optimal instrumental density.*

---

**Description**

Finds the optimal instrumental density  $h$  to be used in the bidirectional acceptance sampling.

**Usage**

```
find_h_cont(data,g,dhat,range=NULL,M_0=NULL,par0=NULL,lbs,ubs,check.plot=TRUE,
            ylim.f=c(0,2),ylim.d=c(0,2),global=FALSE)
```

**Arguments**

<code>data</code>	A data vector.
<code>g</code>	Function corresponding to the parametric start or postulated model. See details.
<code>dhat</code>	Function corresponding to the estimated comparison density in the $x$ domain. See details.
<code>range</code>	Interval corresponding to the support of the continuous data distribution.
<code>M_0</code>	Starting point for optimization. See details.
<code>par0</code>	A vector of starting values of the parameters to be estimated. See details.
<code>lbs</code>	A vector of the lower bounds of the parameters to be estimated.
<code>ubs</code>	A vector of the upper bounds of the parameters to be estimated.
<code>check.plot</code>	A logical argument indicating if the plot comparing the densities involved should be displayed or not. The default is TRUE.
<code>ylim.f</code>	If <code>check.plot=TRUE</code> , the range of the y-axis of the plot for the probability density functions.
<code>ylim.d</code>	If <code>check.plot=TRUE</code> , the range of the y-axis of the plot for the comparison densities.
<code>global</code>	A logical argument indicating if a global optimization is needed to find the instrumental probability function $h$ . See details.

**Details**

The parametric start specified in `g` is assumed to be fully specified and takes  $x$  as the only argument. The argument `dhat` is the estimated comparison density in the  $x$  domain. We usually get the argument `dhat` by means of the function `d_hat` within our package. The value `M_0` and the vector `par0` are used for the optimization process for finding the optimal instrumental density  $h$ . Usually, we choose `M_0` to be the central point of the range. For example, if the range is from 0 to 30, we choose 15 as starting point. The choice of `M_0` is not expected to affect substantially the accuracy of the solution. The vector `par0` collects initial values for the parameters which characterize the instrumental density. For instance, if  $h$  is a mixture of  $p$  truncated normals, the first  $p-1$  elements of `pis` correspond to the starting values for the first  $p-1$  mixture weights. The following  $p$  elements are the initial values for the means of the  $p$  truncated normals contributing to the mixture. Finally, the last  $p$  elements of `par0` correspond to the starting values for the standard deviations of the  $p$  truncated normals contributing to the mixture. The argument `global` controls whether to use a global optimization or not. A local method allows to reduce the optimization time but the solution is particularly sensible to the choice of `par0`. Conversely, setting `global=TRUE` leads to more accurate result.

**Value**

<code>Mstar</code>	The reciprocal of the acceptance rate.
<code>pis</code>	The optimal set of mixture weights.
<code>means</code>	The optimal mean vector.
<code>sds</code>	The optimal set of standard deviations.
<code>h</code>	Function corresponding to the optimal instrumental density.



**Author(s)**

Sara Algeri and Xiangyu Zhang

**References**

Algeri S. and Zhang X. (2020). Exhaustive goodness-of-fit via smoothed inference and graphics. arXiv:2005.13011.

**See Also**

[d\\_hat](#), [find\\_h\\_disc](#), [rmixtruncnorm](#), [dmixtruncnorm](#)

**Examples**

```
library("truncnorm")
library("LPBkg")
L=0
U=30
range=c(L,U)
set.seed(12395)
meant=-15
sdt=15
n=300
data<-rtruncnorm(n,a=L,b=U,mean=meant,sd=sdt)
poly2_num<-function(x){4.576-0.317*x+0.00567*x^2}
poly2_den<-integrate(poly2_num,lower=L,upper=U)$value
g<-function(x){poly2_num(x)/poly2_den}
ddhat<-d_hat(data,m=2,g, range=c(L,U), selection=FALSE)$dx
lb=c(0,-20,0,0,0)
ub=c(1,10,rep(30,3))
par0=c(0.3,-17,1,10,15)
range=c(L,U)
find_h_cont(data,g,ddhat,range,M_0=10,par0,lb,ub,ylim.f=c(0,0.25),ylim.d=c(-1,2))
```

---

find\_h\_disc

*Finding optimal instrumental mass function.*

---

**Description**

Finds the optimal probability mass function  $h$  to be used in the bidirectional acceptance sampling.

**Usage**

```
find_h_disc(data,g,dhat,lattice=NULL,M_0=NULL,size,par0=NULL,check.plot=TRUE,
            ylim.f=c(0,2),ylim.d=c(0,2),global=FALSE)
```

**Arguments**

<code>data</code>	A data vector.
<code>g</code>	Function corresponding to the parametric start or postulated model. See details.
<code>dhat</code>	Function corresponding to the estimated comparison density in the $x$ domain. See details.
<code>lattice</code>	Support of the discrete data distribution.
<code>size</code>	A positive value corresponding to the target for number of successful trials.
<code>M_0</code>	Starting point for optimization. See details.
<code>par0</code>	A vector of starting values of the parameters to be estimated. See details.
<code>check.plot</code>	A logical argument indicating if the plot comparing the densities involved should be displayed or not. The default is TRUE.
<code>ylim.f</code>	If <code>check.plot=TRUE</code> , the range of the y-axis of the plot for the probability density functions.
<code>ylim.d</code>	If <code>check.plot=TRUE</code> , the range of the y-axis of the plot for the comparison densities.
<code>global</code>	A logical argument indicating if a global optimization is needed to find the instrumental probability function $h$ . See details.

**Details**

The parametric start specified in `g` is assumed to be fully specified and takes  $x$  as only argument. The argument `dhat` is the estimated comparison density in the  $x$  domain. We usually get the argument `dhat` by means of the function `d_hat` within our package. The value `M_0` and the vector `par0` are used for the optimization process for finding the optimal instrumental density  $h$ . Usually, we could choose the `M_0` to be the central point of the `lattice`. For example, if the range is from 0 to 30, we could choose 15 as the starting point. The choice of `M_0` is not expected to affect substantially the accuracy of the solution. The vector `par0` collects initial values for the parameters which characterize the instrumental probability mass function. For instance, if  $h$  is a mixture of  $p$  negative binomials, the first  $p-1$  elements of `pis` correspond to the starting values for the first  $p-1$  mixture weights. The following  $p$  elements are the initial values for the probabilities of success of the  $p$  negative binomials contributing to the mixture. The argument `global` controls whether to use a global optimization or not. A local method allows to reduce the optimization time but the solution is particularly sensible to the choice of `par0`. Conversely, setting `global=TRUE` leads to more accurate result.

**Value**

<code>Mstar</code>	The reciprocal of the acceptance rate.
<code>pis</code>	The optimal set of mixture weights.
<code>probs</code>	The optimal set of probabilities of success.
<code>h</code>	Function corresponding to the optimal instrumental probability mass function.

**Author(s)**

Xiangyu Zhang and Sara Algeri

**References**

Algeri S. and Zhang X. (2020). Smoothed inference and graphics via LP modeling, arXiv:2005.13011.

**See Also**

[d\\_hat](#), [find\\_h\\_cont](#), [rmixtruncnorm](#), [dmixtruncnorm](#)

**Examples**

```
lattice=seq(0,20,length=21)
n=200
data<-rbinom(n,size=20,prob=0.5)
g<-function(x)dpois(x,10)/(ppois(20,10)-ppois(0,10))
ddhat<-d_hat(data,m=1,g=g,lattice=lattice,selection=TRUE)$dx
find_h_disc(data=data,g=g,dhat=ddhat,lattice,M_0=10,size=15,par0=c(0.3,0.5,0.6),
            check.plot=TRUE,ylim.f=c(0,0.4),ylim.d=c(-3,2.5),global=FALSE)
```

---

 rmixnegbinom

*Random numbers generator for negative binomial mixtures*


---

**Description**

Generates random samples from a mixture of negative binomials.

**Usage**

```
rmixnegbinom(n, pis, size, probs)
```

**Arguments**

n	Size of the random sample.
pis	A vector collecting the mixture weights. See details.
size	A positive value corresponding to the target for number of successful trials. See details.
probs	A vector collecting the probabilities of success for every mixture component.

**Details**

The argument `pis` is a vector with length equal the number of components in the mixture. The vector `pis` must sum up to one, e.g. `c(0.7, 0.2, 0.1)`. All the negative binomials contributing to the mixture are assumed to have the same size.

**Value**

A vector collecting the random sample of size `n` from the mixture of negative binomials specified.

**Author(s)**

Xiangyu Zhang and Sara Algeri

**See Also**

[rmixtruncnorm](#), [dmixtruncnorm](#), [dmixnegbinom](#), [find\\_h\\_disc](#).

**Examples**

```
rmixnegbinom(n=100,pis=c(0.3,0.6,0.1),size=2,probs=c(0.3,0.4,0.2))
```

---

 rmixtruncnorm

*Random numbers generator for truncated normal mixtures*


---

**Description**

Generates random samples from a mixture of truncated normals.

**Usage**

```
rmixtruncnorm(n, pis, means, sds, range)
```

**Arguments**

n	Size of the random sample.
pis	A vector collecting the mixture weights. See details.
means	A vector collecting the means of the mixture components.
sds	A vector collecting the standard deviations of the mixture components.
range	Interval corresponding to the support of each of truncated normal contributing to the mixture. See details.

**Details**

The argument `pis` is a vector with its length equal the number of components in the mixture. The vector `pis` must sum up to one, e.g. `c(0.7, 0.2, 0.1)`. The argument `range` is an interval corresponding to the support of each truncated normal contributing to the mixture.

**Value**

A vector collecting the random sample of size `n` from the mixture of truncated specified.

**Author(s)**

Sara Algeri and Xiangyu Zhang

**See Also**

[dmixtruncnorm](#), [dmixnegbinom](#), [rmixnegbinom](#), [find\\_h\\_cont](#).

**Examples**

```
rmixtruncnorm(n=10, pis=c(0.5, 0.5), means=c(3, 6), sds=c(3, 4), range=c(0, 30))
```

# Index

- \* **Comparison density estimate**  
d\_hat, 6
- \* **Comparison density plot**  
CDplot, 2
- \* **Deviance test**  
CDplot, 2
- \* **Gajek estimator**  
d\_hat, 6
- \* **Mixture of negative binomials sampler**  
rmixnegbinom, 11
- \* **Mixture of negative binomials**  
dmixnegbinom, 4
- \* **Mixture of truncated normals sampler**  
rmixtruncnorm, 12
- \* **Mixture of truncated normals**  
dmixtruncnorm, 5
- \* **Optimal instrumental density for  
bidirectional acceptance sampling**  
find\_h\_cont, 7
- \* **Optimal instrumental mass function for  
bidirectional acceptance sampling**  
find\_h\_disc, 9

CDplot, 2, 7

d\_hat, 3, 6, 9, 11

dmixnegbinom, 4, 5, 12, 13

dmixtruncnorm, 4, 5, 9, 11–13

find\_h\_cont, 3, 5, 7, 11, 13

find\_h\_disc, 3, 4, 9, 9, 12

rmixnegbinom, 4, 5, 11, 13

rmixtruncnorm, 4, 5, 9, 11, 12, 12