

# Package ‘kernopt’

May 8, 2026

**Title** Estimating Count Data Distributions with Discrete Optimal Symmetric Kernel

**Version** 1.0.0

**Maintainer** Thomas Fillon <thomas.fillon@univ-ubs.fr>

**Description** Implementation of Discrete Symmetric Optimal Kernel for estimating count data distributions, as described by T. Senga Kiessé and G. Durrieu (2024) <[doi:10.1016/j.spl.2024.110078](https://doi.org/10.1016/j.spl.2024.110078)>. The nonparametric estimator using the discrete symmetric optimal kernel was illustrated on simulated data sets and a real-word data set included in the package, in comparison with two other discrete symmetric kernels.

**License** GPL (>= 3)

**Encoding** UTF-8

**RoxygenNote** 7.3.2

**Imports** stats

**Suggests** knitr, rmarkdown, testthat (>= 3.0.0)

**Config/testthat/edition** 3

**VignetteBuilder** knitr

**Depends** R (>= 2.10)

**LazyData** true

**URL** <https://thomasfillon.github.io/kernopt/>,  
<https://github.com/thomasfillon/kernopt>

**BugReports** <https://github.com/thomasfillon/kernopt/issues>

**NeedsCompilation** no

**Author** Tristan Senga Kiese [aut] (ORCID:  
<<https://orcid.org/0000-0003-2710-5825>>),  
Gilles Durrieu [aut] (ORCID: <<https://orcid.org/0000-0003-0375-912X>>),  
Thomas Fillon [aut, cre] (ORCID:  
<<https://orcid.org/0000-0003-1207-8414>>),  
INRAE, Institut Agro, CNRS - UMR SAS [cph],  
Université de Bretagne Sud, CNRS - UMR 6205 LMBA [cph]

**Repository** CRAN

**Date/Publication** 2025-03-19 13:20:05 UTC

## Contents

cv_bandwidth . . . . .	2
discrete_binomial . . . . .	3
discrete_epanech . . . . .	4
discrete_kernel . . . . .	4
discrete_optimal . . . . .	5
discrete_triang . . . . .	6
estim_kernel . . . . .	7
fish_data . . . . .	8

<b>Index</b>	<b>9</b>
--------------	----------

---

cv_bandwidth	<i>Cross-Validation function for bandwidth parameter selection of discrete kernel</i>
--------------	---

---

## Description

Cross-Validation function for bandwidth parameter selection of discrete kernel

## Usage

```
cv_bandwidth(
  kernel = c("optimal", "triang", "epanech", "binomial"),
  v,
  h,
  k = NULL
)
```

## Arguments

kernel	the type of kernel. Currently supported kernels are limited to: "optimal", "triang", "epanech" and "binomial"
v	the vector of observations
h	the list of bandwidth parameters to test in cross validation
k	Optional: the integer (positive) parameter that defined the support of the kernel function (corresponds to parameter 'a' for triangular kernel). It is only used for optimal and triangular kernel

## Value

the optimal bandwidth value

**Examples**

```
n <- 250
mu <- 2 # Mean
y <- sort(rpois(n, mu))
# kernel support parameter
k <- 1
H <- seq((max(y) - min(y)) / 200, (max(y) - min(y)) / 2, length.out = 50)
hcv <- cv_bandwidth(kernel = "optimal", y, h = H, k = k)
```

---

discrete_binomial	<i>Discrete binomial kernel</i>
-------------------	---------------------------------

---

**Description**

Discrete binomial kernel

**Usage**

```
discrete_binomial(x, z, h)
```

**Arguments**

x	the target point at which the density is calculated
z	the vector of observations
h	the bandwidth (or smoothing parameter) which should match the condition $0 <= h < 1$

**Value**

Returns the value of the associated kernel function according to the target x and the bandwidth h.

**Examples**

```
# Basic usage of discrete_binomial() to compute a Discrete Binomial Kernel
discrete_binomial(x = 25, z = 1:50, h = 0.5)
```

discrete\_epanech      *Discrete Epanechnikov kernel*

---

**Description**

Discrete Epanechnikov kernel

**Usage**

```
discrete_epanech(x, z, h)
```

**Arguments**

x                    the target point at which the density is calculated  
z                    the vector of observations  
h                    the bandwidth (or smoothing parameter)

**Value**

Returns the value of the associated kernel function according to the target x and the bandwidth h.

**Examples**

```
# Basic usage of discrete_epanech() to compute a discrete Epanechnikov kernel  
discrete_epanech(x = 25, z = 1:50, h = 20)
```

---

discrete\_kernel      *Discrete kernel function*

---

**Description**

Discrete kernel function

**Usage**

```
discrete_kernel(  
  kernel = c("optimal", "triang", "epanech", "binomial"),  
  x,  
  z,  
  h,  
  k = NULL  
)
```

**Arguments**

kernel	the type of kernel. Currently supported kernels are limited to: "optimal", "triang", "epanech" and "binomial"
x	the target point at which the density is calculated
z	the vector of observations
h	the bandwidth (or smoothing parameter)
k	Optional: the integer (positive) parameter that defined the support of the kernel function (corresponds to parameter 'a' for triangular kernel). It is only used for optimal and triangular kernel

**Value**

Returns the value of the associated kernel function

**See Also**

[discrete\\_optimal\(\)](#), [discrete\\_triang\(\)](#), [discrete\\_epanech\(\)](#), [discrete\\_binomial\(\)](#) which this function wraps.

**Examples**

```
discrete_kernel(kernel = "optimal", x = 25, z = 1:50, h = 0.9, k = 20)
discrete_kernel(kernel = "triang", x = 25, z = 1:50, h = 10, k = 20)
discrete_kernel(kernel = "epanech", x = 25, z = 1:50, h = 20)
discrete_kernel(kernel = "binomial", x = 25, z = 1:50, h = 0.5)
```

---

discrete_optimal	<i>Discrete optimal kernel</i>
------------------	--------------------------------

---

**Description**

Discrete optimal kernel

**Usage**

```
discrete_optimal(x, z, h, k)
```

**Arguments**

x	the target point at which the density is calculated
z	the vector of observations
h	the bandwidth (or smoothing parameter), which should match the condition $(3 / 5) * (1 - 1 / k) < h < 1$
k	the integer (positive) parameter that defined the support of the kernel function

**Value**

Returns the value of the associated kernel function according to the target  $x$  and the bandwidth  $h$ .

**Examples**

```
discrete_optimal(x = 25, z = 1:50, h = 0.9, k = 20)
```

---

discrete_triangular	<i>Discrete triangular kernel</i>
---------------------	-----------------------------------

---

**Description**

Discrete triangular kernel

**Usage**

```
discrete_triangular(x, z, h, a)
```

**Arguments**

$x$	the target point at which the density is calculated
$z$	the vector of observations
$h$	the bandwidth (or smoothing parameter)
$a$	the integer (positive) parameter that defined the support of the kernel function

**Value**

Returns the value of the associated kernel function according to the target  $x$  and the bandwidth  $h$ .

**Examples**

```
# Basic usage of discrete_triangular() to compute a Discrete triangular kernel  
discrete_triangular(x = 25, z = 1:50, h = 10, a = 20)
```

---

estim\_kernel                      *Discrete Kernel Density Estimator*

---

## Description

Discrete Kernel Density Estimator

## Usage

```
estim_kernel(  
  kernel = c("optimal", "triang", "epanech", "binomial"),  
  x,  
  h,  
  v,  
  k = NULL  
)
```

## Arguments

kernel	the type of kernel. Currently supported kernels are limited to: "optimal", "triang", "epanech" and "binomial"
x	the list of target points at which the density is calculated
h	the bandwidth (or smoothing parameter)
v	the vector of observations
k	Optional: the integer (positive) parameter that defined the support of the kernel function (corresponds to parameter 'a' for triangular kernel). It is only used for optimal and triangular kernel

## Value

The estimated discrete kernel density values

## Examples

```
n <- 250  
mu <- 2 # Mean  
x <- 0:10 # target values  
y <- sort(rpois(n, mu)) # simulated Poisson observations  
# kernel parameters  
kernel <- "optimal"  
k <- 1  
# Cross Validation  
H <- seq((max(y) - min(y)) / 200, (max(y) - min(y)) / 2, length.out = 50)  
hcv <- cv_bandwidth(kernel = kernel, y, h = H, k = k)  
# Kernel estimation  
fn_opt_k <- estim_kernel(kernel = kernel, x = x, h = hcv, v = y, k = k)
```

---

`fish_data`*Fish dataset from the SIMTAP project*

---

### Description

The SIMTAP project (Self-sufficient Integrated Multi-Trophic AquaPonic systems) aimed to develop sustainable aquaculture production system that, in particular, contributes to reduce fish feed inputs and resources consumption. We used data from an experiment in which gilthead seabream (*sparus aurata*) were stocked in 1.6 m<sup>3</sup> tanks at a density of 1.5 kg·m<sup>-3</sup>. Fish were reared for 46 days in a single recirculating aquaculture system composed of three rearing tanks. At the beginning of the experiment, a number  $n = 200$  of the fish were individually weighed (dg), and their length at the caudal fork (mm) was measured.

### Usage

`fish_data`

### Format

`fish_data:`

A data frame with 200 rows and 2 columns:

**length** Length in mm**weight** Weight in dg ...

### Source

<https://www.simtap.eu>

# Index

## \* datasets

fish\_data, 8

cv\_bandwidth, 2

discrete\_binomial, 3

discrete\_binomial(), 5

discrete\_epanech, 4

discrete\_epanech(), 5

discrete\_kernel, 4

discrete\_optimal, 5

discrete\_optimal(), 5

discrete\_triangular, 6

discrete\_triangular(), 5

estim\_kernel, 7

fish\_data, 8