

# Package ‘ldbod’

May 8, 2026

**Type** Package

**Title** Local Density-Based Outlier Detection

**Version** 0.1.2

**Author** Kristopher Williams

**Maintainer** Kristopher Williams <kristopher.williams83@gmail.com>

**Description** Flexible procedures to compute local density-based outlier scores for ranking outliers. Both exact and approximate nearest neighbor search can be implemented, while also accommodating multiple neighborhood sizes and four different local density-based methods. It allows for referencing a random subsample of the input data or a user specified reference data set to compute outlier scores against, so both unsupervised and semi-supervised outlier detection can be implemented.

**Depends** R (>= 3.2.0)

**Imports** stats, RANN, mnormt

**License** GPL-3

**URL** <https://github.com/kwilliams83/ldbod>

**LazyData** TRUE

**RoxygenNote** 6.0.1

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2017-05-26 06:04:25 UTC

## Contents

ldbod . . . . .	2
ldbod.ref . . . . .	4
<b>Index</b>	<b>8</b>

---

ldbod *Local Density-Based Outlier Detection using Subsampling with Approximate Nearest Neighbor Search*

---

### Description

This function computes local density-based outlier scores for input data.

### Usage

```
ldbod(X, k = c(10, 20), nsub = nrow(X), method = c("lof", "ldf", "rkof",
  "lpdf"), ldf.param = c(h = 1, c = 0.1), rkof.param = c(alpha = 1, C = 1,
  sig2 = 1), lpdf.param = c(cov.type = "full", sigma2 = 1e-05, tmax = 1, v =
  1), treetype = "kd", searchtype = "standard", eps = 0,
  scale.data = TRUE)
```

### Arguments

X	An n x p data matrix to compute outlier scores
k	A vector of neighborhood sizes, k must be less than nsub
nsub	Subsample size, nsub must be greater than k. Usually nsub = 0.10*n or larger is recommended. Default is nsub = n
method	Character vector specifying the local density-based method(s) to compute. User can specify more than one method. By default all methods are computed
ldf.param	Vector of parameters for method LDF. h is the positive bandwidth parameter and c is a positive scaling constant. Default values are h=1 and c=0.1
rkof.param	Vector of parameters for method RKOF. C is the positive bandwidth parameter, alpha is a sensitivity parameter in the interval [0,1], and sig2 is the variance parameter. Default values are alpha=1, C=1, sig2=1
lpdf.param	Vector of parameters for method LPDF. cov.type is the covariance parameterization type, which users can specify as either 'full' or 'diag'. sigma2 is the positive regularization parameter, tmax is the maximum number of updates, and v is the degrees of freedom for the multivariate t distribution. Default values are cov.type = 'full', tmax=1, sigma2=1e-5, and v=1.
treetype	Character vector specifying tree method. Either 'kd' or 'bd' tree may be specified. Default is 'kd'. Refer to documentation for RANN package.
searchtype	Character vector specifying kNN search type. Default value is "standard". Refer to documentation for RANN package.
eps	Error bound. Default is 0.0 which implies exact nearest neighbour search. Refer to documentation for RANN package.
scale.data	Logical value indicating to scale each feature of X using standard normalization with mean 0 and standard deviation of 1

## Details

Computes the local density-based outlier scores for input data,  $X$ , referencing a random subsample of  $X$ . The subsampled data set is constructed by randomly drawing  $nsub$  samples from  $X$  without replacement.

Four different methods can be implemented LOF, LDF, RKOF, and LPDF. Each method specified returns densities and relative densities. Methods LDF and RKOF uses gaussian kernels, and method LPDF uses multivariate  $t$  distribution. Outlier scores returned are positive except for  $lpde$  and  $lpdr$  which are log scaled densities (natural log). Score  $lpdr$  has shown to be highly sensitive to  $k$ .

All kNN computations are carried out using the `nn2()` function from the RANN package. Multivariate  $t$  densities are computed using the `dmt()` function from the `mnormt` package. Refer to specific packages for more details. Note: all neighborhoods are strictly of size  $k$ ; therefore, the algorithms for LOF, LDF, and RKOF are not exact implementations, but algorithms are similar for most situation and are equivalent when distance to  $k$ -th nearest neighbor is unique. If there are many duplicate data points, then implementation of algorithms could lead to dramatically different (positive or negative) results than those that allow neighborhood sizes larger than  $k$ , especially if  $k$  is relatively small. Removing duplicates is recommended before computing outlier scores unless there is good reason to keep them.

The algorithm can be used to compute an ensemble of outlier scores by using multiple  $k$  values and/or iterating over multiple subsamples.

## Value

A list of length 9 with the elements:

`lrd` –An  $n \times \text{length}(k)$  matrix where each column vector represents the local reachability density (LRD) outlier scores for each specified  $k$  value. Smaller values indicate a point in more outlying.

`lof` –An  $n \times \text{length}(k)$  matrix where each column vector represents the local outlier factor (LOF) outlier scores for each specified  $k$  value. Larger values indicate a point in more outlying.

`lde` –An  $n \times \text{length}(k)$  matrix where each column vector represents the local density estimate (LDE) outlier scores for each specified  $k$  value. Smaller values indicate a point in more outlying.

`ldf` –An  $n \times \text{length}(k)$  matrix where each column vector represents the local density factor (LDF) outlier scores for each specified  $k$  value. Larger values indicate a point in more outlying.

`kde` –An  $n \times \text{length}(k)$  matrix where each column vector represents the kernel density estimate (KDE) outlier scores for each specified  $k$  value. Smaller values indicate a point in more outlying.

`rkof` –An  $n \times \text{length}(k)$  matrix where each column vector represents the robust kernel density factor (RKOF) outlier scores for each specified  $k$  value. Larger values indicate a point in more outlying.

`lpde` –An  $n \times \text{length}(k)$  matrix where each column vector represents the local parametric density estimate (LPDE) outlier scores for each specified  $k$  value on log scale. Smaller values indicate a point in more outlying.

`lpdf` –An  $n \times \text{length}(k)$  matrix where each column vector represents the local parametric density factor (LPDF) outlier scores for each specified  $k$  value. Smaller values indicate a point in more outlying.

`lpdr` –An  $n \times \text{length}(k)$  matrix where each column vector represents the local parametric density ratio (LPDR) outlier scores for each specified  $k$  value. Smaller values indicate a point in more outlying. LPDR is typically used to detect groups of outliers.

If a method is not specified then returns NULL

## References

- M. M. Breunig, H-P. Kriegel, R.T. Ng, and J. Sander (2000). LOF: Identifying density-based local outliers. In Proc. of ACM International Conference on Knowledge Discovery and Data Mining, 93-104.
- L. J. Latecki, A. Lazarevic, and D. Pokrajac (2007). Outlier Detection with kernel density functions. In Proc. of Machine Learning and Data Mining in Pattern Recognition, 61-75
- J. Gao, W. Hu, Z. Zhang, X. Zhang, and O. Wu (2011). RKOF: Robust kernel-based local outlier detection. In Proc. of Advances in Knowledge Discovery and Data Mining, 270-283.
- K. T. Williams (2016). Local parametric density-based outlier detection and ensemble learning with application to malware detection. PhD Dissertation. The University of Texas at San Antonio.

## Examples

```
# 500 x 2 data matrix
X <- matrix(rnorm(1000),500,2)

# five outliers
outliers <- matrix(c(rnorm(2,20),rnorm(2,-12),rnorm(2,-8),rnorm(2,-5),rnorm(2,9)),5,2)
X <- rbind(X,outliers)

# compute outlier scores without subsampling for all methods using neighborhood size of 50
scores <- ldbod(X, k=50)

head(scores$lrd); head(scores$rkof)

# plot data and highlight top 5 outliers returned by lof
plot(X)
top5outliers <- X[order(scores$lof,decreasing=TRUE)[1:5],]
points(top5outliers,col=2)

# plot data and highlight top 5 outliers returned by outlier score lpde
plot(X)
top5outliers <- X[order(scores$lpde,decreasing=FALSE)[1:5],]
points(top5outliers,col=2)

# compute outlier scores for k= 10,20 with 10% subsampling for methods 'lof' and 'lpdf'
scores <- ldbod(X, k = c(10,20), nsub = 0.10*nrow(X), method = c('lof','lpdf'))

# plot data and highlight top 5 outliers returned by lof for k=20
plot(X)
top5outliers <- X[order(scores$lof[,2],decreasing=TRUE)[1:5],]
points(top5outliers,col=2)
```

**Description**

This function computes local density-based outlier scores for input data and user specified reference set.

**Usage**

```
ldbod.ref(X, Y, k = c(10, 20), method = c("lof", "ldf", "rkof", "lpdf"),
  ldf.param = c(h = 1, c = 0.1), rkof.param = c(alpha = 1, C = 1, sig2 = 1),
  lpdf.param = c(cov.type = "full", sigma2 = 1e-05, tmax = 1, v = 1),
  treetype = "kd", searchtype = "standard", eps = 0, scale.data = TRUE)
```

**Arguments**

X	An n x p data matrix to compute outlier scores
Y	An m x p reference data matrix.
k	A vector of neighborhood sizes, k must be less than m.
method	Character vector specifying the local density-based method(s) to compute. User can specify more than one method. By default all methods are computed
ldf.param	Vector of parameters for method LDF. h is the positive bandwidth parameter and c is a positive scaling constant. Default values are h=1 and c=0.1
rkof.param	Vector of parameters for method RKOF. C is the positive bandwidth parameter, alpha is a sensitivity parameter in the interval [0,1], and sig2 is the variance parameter. Default values are alpha=1, C=1, sig2=1
lpdf.param	Vector of parameters for method LPDF. cov.type is the covariance parameterization type, which users can specify as either 'full' or 'diag'. sigma2 is the positive regularization parameter, tmax is the maximum number of updates, and v is the degrees of freedom for the multivariate t distribution. Default values are cov.type = 'full', tmax=1, sigma2=1e-5, and v=1.
treetype	Character vector specifying tree method. Either 'kd' or 'bd' tree may be specified. Default is 'kd'. Refer to documentation for RANN package.
searchtype	Character vector specifying kNN search type. Default value is "standard". Refer to documentation for RANN package.
eps	Error bound. Default is 0.0 which implies exact nearest neighbour search. Refer to documentation for RANN package.
scale.data	Logical value indicating to scale each feature of X using standard normalization based on mean and standard deviation for features of Y.

**Details**

Computes local density-based outlier scores for input data, X, referencing data Y. For semi-supervised outlier detection Y would be a set of "normal" reference points; otherwise, Y can be any other set of reference points of interest. This allows users the flexibility to reference other data sets besides X or a subset of X. Four different methods can be implemented LOF, LDF, RKOF, and LPDF. Each method specified returns densities and relative densities. Methods LDF and RKOF uses gaussian kernels, and method LPDF uses multivariate t distribution. Outlier scores returned are non-negative

except for `lpde` and `lpdr` which are log scaled densities (natural log). Note: Outlier score `lpdr` is strictly designed for unsupervised outlier detection and should not be used in the semi-supervised setting. Refer to references for more details about each method.

All kNN computations are carried out using the `nn2()` function from the RANN package. Multivariate t densities are computed using the `dmt()` function from the `mnormt` package. Refer to specific packages for more details. Note: all neighborhoods are strictly of size `k`; therefore, the algorithms for LOF, LDF, and RKOF are not exact implementations, but algorithms are similar for most situation and are equivalent when distance to `k`-th nearest neighbor is unique. If there are many duplicate data points in `Y`, then implementation of algorithms could lead to dramatically different (positive or negative) results than those that allow neighborhood sizes larger than `k`, especially if `k` is relatively small. Removing duplicates is recommended before computing outlier scores unless there is good reason to keep them.

The algorithm can be used to compute an ensemble of unsupervised outlier scores by using multiple `k` values and/or multiple iterations of reference data.

### Value

A list of length 9 with the elements:

`lrd` – An  $n \times \text{length}(k)$  matrix where each column vector represents outlier scores for each specified `k` value. Smaller values indicate a point in more outlying.

`lof` – An  $n \times \text{length}(k)$  matrix where each column vector represents outlier scores for each specified `k` value. Larger values indicate a point in more outlying.

`lde` – An  $n \times \text{length}(k)$  matrix where each column vector represents outlier scores for each specified `k` value. Smaller values indicate a point in more outlying.

`ldf` – An  $n \times \text{length}(k)$  matrix where each column vector represents outlier scores for each specified `k` value. Larger values indicate a point in more outlying.

`kde` – An  $n \times \text{length}(k)$  matrix where each column vector represents outlier scores for each specified `k` value. Smaller values indicate a point in more outlying.

`rkof` – An  $n \times \text{length}(k)$  matrix where each column vector represents outlier scores for each specified `k` value. Larger values indicate a point in more outlying.

`lpde` – An  $n \times \text{length}(k)$  matrix where each column vector represents outlier scores for each specified `k` value. Smaller values indicate a point in more outlying.

`lpdf` – An  $n \times \text{length}(k)$  matrix where each column vector represents outlier scores for each specified `k` value. Smaller values indicate a point in more outlying.

`lpdr` – An  $n \times \text{length}(k)$  matrix where each column vector represents outlier scores for each specified `k` value. Smaller values indicate a point in more outlying.

If a method is not specified then returns NULL

### References

M. M. Breunig, H-P. Kriegel, R.T. Ng, and J. Sander (2000). LOF: Identifying density-based local outliers. In Proc. of ACM International Conference on Knowledge Discovery and Data Mining, 93-104.

L. J. Latecki, A. Lazarevic, and D. Pokrajac (2007). Outlier Detection with kernel density functions. In Proc. of Machine Learning and Data Mining in Pattern Recognition, 61-75

J. Gao, W. Hu, Z. Zhang, X. Zhang, and O. Wu (2011). RKOF: Robust kernel-based local outlier detection. In Proc. of Advances in Knowledge Discovery and Data Mining, 270-283.

K. T. Williams (2016). Local parametric density-based outlier detection and ensemble learning with application to malware detection. PhD Dissertation. The University of Texas at San Antonio.

## Examples

```
# 500 x 2 data matrix
X <- matrix(rnorm(1000),500,2)
Y <- X
# five outliers
outliers <- matrix(c(rnorm(2,20),rnorm(2,-12),rnorm(2,-8),rnorm(2,-5),rnorm(2,9)),5,2)
X <- rbind(X,outliers)

# compute outlier scores referencing Y for all methods using a neighborhood size of 50
scores <- ldbod.ref(X,Y, k=50)

head(scores$lrd); head(scores$rkof)

# plot data and highlight top 5 outliers returned by lof
plot(X)
top5outliers <- X[order(scores$lof,decreasing=TRUE)[1:5],]
points(top5outliers,col=2)

# plot data and highlight top 5 outliers returned by outlier score lpde
plot(X)
top5outliers <- X[order(scores$lpde,decreasing=FALSE)[1:5],]
points(top5outliers,col=2)

# compute outlier scores for k= 10,20 referencing Y for methods 'lof' and 'lpdf'
scores <- ldbod.ref(X,Y, k = c(10,20), method = c('lof','lpdf'))

# plot data and highlight top 5 outliers returned by lof for k=20
plot(X)
top5outliers <- X[order(scores$lof[,2],decreasing=TRUE)[1:5],]
points(top5outliers,col=2)
```

# Index

ldbod, [2](#)  
ldbod.ref, [4](#)