

# Package: prim (via r-universe)

November 28, 2024

**Version** 1.0.22

**Date** 2024-11-28

**Title** Patient Rule Induction Method (PRIM)

**Maintainer** Tarn Duong <tarn.duong@gmail.com>

**Depends** R (>= 2.10.0)

**Imports** scales, tcltk, plot3D

**Suggests** knitr, rmarkdown, MASS

**VignetteBuilder** knitr

**Description** Patient Rule Induction Method (PRIM) for bump hunting in high-dimensional data.

**License** GPL-2 | GPL-3

**URL** <https://www.mvstat.net/tduong/>

**NeedsCompilation** no

**Author** Tarn Duong [aut, cre] (<<https://orcid.org/0000-0002-1198-3482>>)

**Repository** CRAN

**Date/Publication** 2024-11-28 13:50:13 UTC

## Contents

prim-package . . . . .	2
plot.prim . . . . .	2
prim S3 methods . . . . .	4
prim.box . . . . .	5
quasiflow . . . . .	7
<b>Index</b>	<b>8</b>

---

 prim-package

*Patient Rule Induction Method (PRIM)*


---

### Description

PRIM for bump-hunting for high-dimensional regression-type data.

### Details

The data are  $(\mathbf{X}_1, Y_1), \dots, (\mathbf{X}_n, Y_n)$  where  $\mathbf{X}_i$  is d-dimensional and  $Y_i$  is a scalar response. We wish to find the modal (and/or anti-modal) regions in the conditional expectation  $m(\mathbf{x}) = E(Y|\mathbf{x})$ .

PRIM is a bump-hunting technique introduced by Friedman & Fisher (1999), taken from data mining. PRIM estimates are a sequence of nested hyper-rectangles (boxes).

For an overview of this package, see `vignette("prim")` for PRIM estimation for 2- and 5-dimensional data.

### Author(s)

Tarn Duong <tarn.duong@gmail.com>

### References

Friedman, J.H. & Fisher, N.I. (1999) Bump-hunting for high dimensional data, *Statistics and Computing*, **9**, 123–143.

Hyndman, R.J. Computing and graphing highest density regions. *American Statistician*, **50**, 120–126.

---

 plot.prim

*PRIM plot for multivariate data*


---

### Description

PRIM plot for multivariate data.

### Usage

```
## S3 method for class 'prim'
plot(x, splom=TRUE, ...)
```

### Arguments

x	object of class prim
splom	flag for plotting 3-d data as scatter plot matrix. Default is TRUE.
...	other graphics parameters

**Details**

The function headers are

```
## bivariate
x, col, xlim, ylim, xlab, ylab, add=FALSE, add.legend=FALSE, cex.legend=1,
pos.legend, lwd=1, border, col.vec=c("blue", "orange"), alpha=1, ...)

## trivariate
plot(x, xlim, ylim, zlim, xlab, ylab, zlab, col.vec=c("blue","orange"),
alpha=1, theta=30, phi=40, d=4, ...)

## d-variate
plot(x, xmin, xmax, xlab, ylab, x.pt, m, col.vec=c("blue","orange"),
alpha=1, ...)
```

The arguments are

add.legend flag for adding legend (2-d plot)  
pos.legend (x,y) co-ordinates for legend (2-d plot)  
cex.legend cex graphics parameter for legend (2-d plot)  
col.vec vector of plotting colours, one for each box  
xlab,ylab,zlab,xlim,ylim,zlim,add,lwd,alpha,phi,theta,d usual graphics parameters  
xmin,xmax vector of minimum and maximum axis plotting values for scatter plot matrix  
x.pt data set to plot (other than x)

**Value**

Plot of 2-dim PRIM is a set of nested rectangles. Plot of 3-dim PRIM is a scatter point cloud. Plot of d-dim PRIM is a scatter plot matrix. The scatter plots indicate which points belong to which box.

**See Also**

[prim.box](#), [predict.prim](#)

**Examples**

```
## see ?predict.prim for bivariate example
## trivariate example
data(quasiflow)
qf <- quasiflow[1:1000,1:3]
qf.label <- quasiflow[1:1000,4]
thr <- c(0.25, -0.3)
qf.prim <- prim.box(x=qf, y=qf.label, threshold=thr, threshold.type=0)
plot(qf.prim, alpha=0.5)
plot(qf.prim, alpha=0.5, splom=FALSE, ticktype="detailed", colkey=FALSE)
```

---

prim S3 methods      *S3 methods for PRIM for multivariate data*

---

### Description

S3 methods PRIM for multivariate data.

### Usage

```
## S3 method for class 'prim'
predict(object, newdata, y.fun.flag=FALSE, ...)
## S3 method for class 'prim'
summary(object, ..., print.box=FALSE)
```

### Arguments

object	object of class prim
newdata	data matrix
y.fun.flag	flag to return y value of PRIM box rather than box label. Default is FALSE.
print.box	flag to print out limits of all PRIM boxes. Default is FALSE.
...	other parameters

### Details

–The predict method returns the value of PRIM box number in which newdata are located.

–The summary method displays a table with three columns: box-fun is the y value, box-mass is the mass of the box, threshold.type is the threshold direction indicator: 1 = ">= threshold", -1 = "<=threshold". Each box corresponds to a row. The second last row marked with an asterisk is the box which collates the remaining data points not belonging to a specific PRIM box. The final row is an overall summary, i.e. box-fun is the overall mean of y and box-mass is 1.

### Examples

```
data(quasiflow)
qf <- quasiflow[1:1000,1:2]
qf.label <- quasiflow[1:1000,3]*quasiflow[1:1000,4]

qf.prim <- prim.box(x=qf, y=qf.label, threshold=c(0.3, -0.1), threshold.type=0,
  verbose=TRUE)
## verbose=TRUE prints out extra informaton about peeling and pasting

summary(qf.prim)
predict(qf.prim, newdata=c(0.6,0.2))

## using median insted of mean for the response y

qf.prim2 <- prim.box(x=qf, y=qf.label, threshold=c(0.5, -0.2),
```

```

    threshold.type=0, y.fun=median)
summary(qf.prim2)
predict(qf.prim2, newdata=c(0.6,0.2))

```

---

prim.box	<i>PRIM for multivariate data</i>
----------	-----------------------------------

---

## Description

PRIM for multivariate data.

## Usage

```

prim.box(x, y, box.init=NULL, peel.alpha=0.05, paste.alpha=0.01,
        mass.min=0.05, threshold, pasting=TRUE, verbose=FALSE,
        threshold.type=0, y.fun=mean)

prim.hdr(prim, threshold, threshold.type, y.fun=mean)
prim.combine(prim1, prim2, y.fun=mean)

```

## Arguments

x	matrix of data values
y	vector of response values
y.fun	function applied to response y. Default is mean.
box.init	initial covering box
peel.alpha	peeling quantile tuning parameter
paste.alpha	pasting quantile tuning parameter
mass.min	minimum mass tuning parameter
threshold	threshold tuning parameter(s)
threshold.type	threshold direction indicator: 1 = ">= threshold", -1 = "<= threshold", 0 = ">= threshold[1] & <= threshold[2]"
pasting	flag for pasting
verbose	flag for printing output during execution
prim, prim1, prim2	objects of type prim

## Details

The data are  $(\mathbf{X}_1, Y_1), \dots, (\mathbf{X}_n, Y_n)$  where  $\mathbf{X}_i$  is d-dimensional and  $Y_i$  is a scalar response. PRIM finds modal (and/or anti-modal) regions in the conditional expectation  $m(\mathbf{x}) = \mathbf{E}(Y|\mathbf{x})$ .

In general,  $Y_i$  can be real-valued. See vignette("prim"). Here, we focus on the special case for binary  $Y_i$ . Let  $Y_i = 1$  when  $\mathbf{X}_i \sim F^+$ ; and  $Y_i = -1$  when  $\mathbf{X}_i \sim F^-$  where  $F^+$  and  $F^-$  are different distribution functions. In this set-up, PRIM finds the regions where  $F^+$  and  $F^-$  are most different.

The tuning parameters `peel.alpha` and `paste.alpha` control the ‘patience’ of PRIM. Smaller values involve more patience. Larger values less patience. The peeling steps remove data from a box till either the box mean is smaller than `threshold` or the box mass is less than `mass.min`. Pasting is optional, and is used to correct any possible over-peeling. The default values for `peel.alpha`, `paste.alpha` and `mass.min` are taken from Friedman & Fisher (1999).

The type of PRIM estimate is controlled `threshold` and `threshold.type`:

`threshold.type=1` search for  $\{m(\mathbf{x}) \geq \text{threshold}\}$ .

`threshold.type=-1` search for  $\{m(\mathbf{x}) \leq \text{threshold}\}$ .

`threshold.type=0` search for both  $\{m(\mathbf{x}) \geq \text{threshold}[1]\}$  and  $\{m(\mathbf{x}) \leq \text{threshold}[2]\}$ .

There are two ways of using PRIM. One is `prim.box` with pre-specified `threshold(s)`. This is appropriate when the `threshold(s)` are known to produce good estimates.

On the other hand, if the user doesn’t provide `threshold` values then `prim.box` computes box sequences which cover the data range. These can then be pruned at a later stage. `prim.hdr` allows the user to specify many different `threshold` values in an efficient manner, without having to recompute the entire PRIM box sequence. `prim.combine` can be used to join the regions computed from `prim.hdr`. See the examples below.

## Value

– `prim.box` produces a PRIM estimate, an object of type `prim`, which is a list with 8 fields:

<code>x</code>	list of data matrices
<code>y</code>	list of response variable vectors
<code>y.mean</code>	list of vectors of box mean for <code>y</code>
<code>box</code>	list of matrices of box limits (first row = minima, second row = maxima)
<code>mass</code>	vector of box masses (proportion of points inside a box)
<code>num.class</code>	total number of PRIM boxes
<code>num.hdr.class</code>	total number of PRIM boxes which form the HDR
<code>ind</code>	threshold direction indicator: 1 = " $\geq$ threshold", -1 = " $\leq$ threshold"

The above lists have `num.class` fields, one for each box.

– `prim.hdr` takes a `prim` object and prunes it using different `threshold` values. Returns another `prim` object. This is much faster for experimenting with different `threshold` values than calling `prim.box` each time.

– `prim.combine` combines two `prim` objects into a single `prim` object. Usually used in conjunction with `prim.hdr`. See examples below.

## Examples

```
data(quasiflow)
qf <- quasiflow[1:1000,1:2]
qf.label <- quasiflow[1:1000,4]

## using only one command
```

```
thr <- c(0.25, -0.3)
qf.prim1 <- prim.box(x=qf, y=qf.label, threshold=thr, threshold.type=0)

## alternative - requires more commands but allows more control
## in intermediate stages
qf.primp <- prim.box(x=qf, y=qf.label, threshold.type=1)
  ## default threshold too low, try higher one

qf.primp.hdr <- prim.hdr(prim=qf.primp, threshold=0.25, threshold.type=1)
qf.primn <- prim.box(x=qf, y=qf.label, threshold=-0.3, threshold.type=-1)
qf.prim2 <- prim.combine(qf.primp.hdr, qf.primn)

plot(qf.prim1, alpha=0.2)  ## orange=x1>x2, blue x2<x1
points(qf[qf.label==1,], cex=0.5)
points(qf[qf.label==-1,], cex=0.5, col=2)
```

---

quasiflow

*Quasi flow cytometry data*

---

## Description

This data set is simulated data from two normal mixture distributions, mimicking a flow cytometry data set. It contains 10000 observations from an HIV+ patient and 10000 observations an HIV-patient.

## Usage

```
data(quasiflow)
```

## Format

quasiflow is a matrix with 6 columns and 20000 rows. Each row corresponds to measurements for one cell. The first 5 columns are flow cytometric measurements and the sixth column is a binary indicator, with 1 = HIV+ and -1 = HIV-.

## Source

Generated by package author.

# Index

- \* **datasets**
  - quasiflow, [7](#)
- \* **hplot**
  - plot.prim, [2](#)
- \* **multivariate**
  - prim S3 methods, [4](#)
  - prim.box, [5](#)
- \* **package**
  - prim-package, [2](#)

plot.prim, [2](#)  
predict.prim, [3](#)  
predict.prim(prim S3 methods), [4](#)  
prim(prim-package), [2](#)  
prim S3 methods, [4](#)  
prim-package, [2](#)  
prim.box, [3](#), [5](#)  
prim.combine(prim.box), [5](#)  
prim.hdr(prim.box), [5](#)

quasiflow, [7](#)

summary.prim(prim S3 methods), [4](#)