

# Package ‘hyperoverlap’

February 7, 2019

**Title** Overlap Detection in n-Dimensional Space

**Version** 0.1

**Description** Uses support vector machines to identify a perfectly separating hyperplane (linear or curvilinear) between two entities in high-dimensional space. If this plane exists, the entities do not overlap. Applications include overlap detection in morphological, resource or environmental dimensions.

**Depends** R (>= 3.4.4)

**Imports** e1071, matlib, ggplot2, rgl, misc3d, methods, stats, MASS

**License** GPL-3

**Encoding** UTF-8

**LazyData** true

## R topics documented:

hyperoverlap . . . . .	1
hyperoverlap-class . . . . .	2
hyperoverlap_detect . . . . .	2
hyperoverlap_lda . . . . .	3
hyperoverlap_pairs_plot . . . . .	4
hyperoverlap_plot . . . . .	5
hyperoverlap_set . . . . .	6

<b>Index</b>	<b>7</b>
--------------	----------

---

hyperoverlap	<i>Hyperoverlap: Detection and Visualisation of Overlap in n-Dimensional Space</i>
--------------	--

---

## Description

Uses support vector machines to identify a perfectly separating hyperplane (linear or curvilinear) between two entities in high-dimensional space. If this plane exists, the entities do not overlap. Applications include overlap detection in morphological, resource or environmental dimensions.

**Details**

More details available in our forthcoming paper and vignette.

**Author(s)**

Matilda Brown <matilda.brown@utas.edu.au>

---

hyperoverlap-class	<i>Storage class for the description of hyperoverlaps</i>
--------------------	---

---

**Description**

Storage class for the description of hyperoverlaps

**Slots**

entity1 A length-one character vector  
 entity2 A length-one character vector  
 dimensions A length n character vector containing the variables used to define the space  
 occurrences A matrix containing the labelled input data  
 shape shape of the decision boundary; either "linear" or "curvilinear"  
 polynomial.order a length-one numeric vector showing the polynomial order of the most accurate kernel function. "0" if linear kernel.  
 result a length-one character vector, either "overlap" or "non-overlap"  
 accuracy a 2x2 table with the true (y) and predicted (pred) labels  
 number.of.points.misclassified a length-one numeric vector  
 decision.boundary list of 2; used for plotting decision boundary

---

hyperoverlap_detect	<i>Overlap detection in n-dimensional space using support vector machines (SVMs)</i>
---------------------	--

---

**Description**

Given a matrix containing the data and labels for two entities, a support vector machine is trained on these data and the predicted label of each point is evaluated. If every point has been classified correctly, the entities can be separated and they do not overlap.

**Usage**

```
hyperoverlap_detect(x, y, kernel="polynomial", kernel.degree = 3, cost = 50,
  stoppage.threshold = 0.2)
```

**Arguments**

<code>x</code>	A matrix or data.frame containing the data for both entities
<code>y</code>	A vector of labels for each observation
<code>kernel</code>	A length-one character vector, either "linear" or "polynomial"
<code>kernel.degree</code>	Parameter needed for kernel of type polynomial (default=3).
<code>cost</code>	Specifies the SVM margin 'hardness'. Default value is 50, but can be increased for improved accuracy, although this increases runtimes and memory usage.
<code>stoppage.threshold</code>	Numeric. If the number of points misclassified using a linear hyperplane exceeds this proportion of the number of observations, non-linear separation is not attempted. Must be between 0 and 1 (default = 0.2).

**Details**

Input data should be preprocessed so that all variables are comparable (e.g. same order of magnitude). Polynomial kernels allow curvilinear decision boundaries to be found between entities (see <https://www.cs.cmu.edu/~ggordon/SVMs/new-svms-and-kernels.pdf>). Smaller values of `kernel.degree` permit less complex decision boundaries; biological significance is likely to be lost at values > 5.

**Value**

A `hyperoverlap-class` object

**Examples**

```
## Not run:
data = iris[which(iris$Species!="versicolor"),]
x = hyperoverlap_detect(data[,1:3],data$Species, kernel="linear")

## End(Not run)
```

---

hyperoverlap\_lda

*Hyperoverlap visualisation using linear discriminant analysis*


---

**Description**

Hyperoverlap visualisation using linear discriminant analysis

**Usage**

```
hyperoverlap_lda(x, return.plot=TRUE, visualise3d=FALSE)
```

**Arguments**

x	An object of class "Hyperoverlap"
return.plot	Logical. If TRUE, data are plotted using plot()
visualise3d	Logical. If FALSE, data are projected onto two axes (LDA1, residualPCA1). If TRUE, data are projected onto three axes (LDA1, residualPCA1, residualPCA2)

**Details**

This function provides a way to visualise overlap (or non-overlap) between classes of high dimensional data. For inspection, it is useful to use the base graphics package (implemented by return.plot=TRUE). The transformed coordinates of each point are also returned as a dataframe, which can be plotted with user-defined parameters.

**Value**

Returns a dataframe with columns "Entity", "LDA1", "residualPCA1", "residualPCA2" (if visualise3d = TRUE)

**See Also**

hyperoverlap\_detect

**Examples**

```
## Not run:
#using iris dataset reduced to two species
data = iris[which(iris$Species!="versicolor"),]
x = hyperoverlap_detect(data[1:4], data$Species)
hyperoverlap_lda(x)

## End(Not run)
```

---

hyperoverlap\_pairs\_plot

*Overlap heatmap plots for comparison of multiple entities*

---

**Description**

This function plots a matrix of overlap.

**Usage**

```
hyperoverlap_pairs_plot(x, cols = pal)
```

**Arguments**

x	A matrix of the form produced by produced by hyperoverlap_set() which contains the columns entity1, entity2, and result
cols	A vector of colours (defaults to red (a)non-overlap) and blue (overlap))

**Value**

A ggplot object

**Examples**

```
## Not run:  
hyperoverlap_pairs_plot(hyperoverlap.set.iris)  
  
## End(Not run)
```

---

hyperoverlap_plot	<i>Overlap plotting</i>
-------------------	-------------------------

---

**Description**

Plot the optimal separating hyperplane found by `hyperoverlap_detect()`. Only used for data of three dimensions or fewer.

**Usage**

```
hyperoverlap_plot(x)
```

**Arguments**

x                    an [hyperoverlap-class](#) object

**Value**

plot of data and decision boundary

**See Also**

[hyperoverlap\\_detect](#)

**Examples**

```
## Not run:  
hyperoverlap_plot(hyperoverlap_iris)  
  
## End(Not run)
```

---

hyperoverlap_set	<i>Pairwise overlap detection in n-dimensional space of multiple entities using support vector machines (SVMs)</i>
------------------	--

---

## Description

This function is a wrapper for [hyperoverlap\\_detect](#)

## Usage

```
hyperoverlap_set(x,y,kernel="polynomial",kernel.degree=3, cost=50,
stoppage.threshold=0.2, write.to.file=TRUE,
path=paste0("hyperoverlap_",Sys.time(),"/"))
```

## Arguments

x	A matrix or data.frame containing the ecological data
y	A vector of labels for each observation
kernel	A length-one character vector, either "linear" or "polynomial"
kernel.degree	For kernel="polynomial", specifies the order of polynomial kernel function. Smaller orders allow less complex boundary shapes. Values greater than 5 are not recommended for this parameter.
cost	Specifies the SVM margin 'hardness'. Default value is 50, but can be increased for improved accuracy, although this increases runtimes and memory usage.
stoppage.threshold	Default = 0.2. If the number of points misclassified using a linear hyperplane exceeds this proportion of the number of observations, non-linear separation is not attempted. Can be made 1 to force non-linear separation.
write.to.file	TRUE or FALSE, specifies whether each Hyperoverlap object should be saved as a .rds file
path	where to write .rds files to. Defaults to a folder in the working directory titled by Sys.time()

## Value

A long-form matrix with the following columns: entity1, entity2, shape,polynomial.order (if kernel=polynomial), result, number.of.points.misclassified. If specified, individual Hyperoverlap-class objects are written to file

## Examples

```
## Not run:
data(iris)
hyperoverlap.iris.set = hyperoverlap_set(iris[1:3],iris$Species, kernel="linear")

## End(Not run)
```

# Index

hyperoverlap, [1](#)  
hyperoverlap-class, [2](#)  
hyperoverlap-package (hyperoverlap), [1](#)  
hyperoverlap\_detect, [2](#), [5](#), [6](#)  
hyperoverlap\_lda, [3](#)  
hyperoverlap\_pairs\_plot, [4](#)  
hyperoverlap\_plot, [5](#)  
hyperoverlap\_set, [6](#)