

# Package ‘rtc’

August 15, 2014

**Type** Package

**Title** Tree Clustering

**Version** 0.1.0

**Date** 2014-08-14

**Author** Peter Kuma

**Maintainer** Peter Kuma <pkuma@pixelfederation.com>

**Description** Cluster analysis using decision trees

**License** Proprietary

## R topics documented:

rtc-package . . . . .	<a href="#">1</a>
tree.clustering . . . . .	<a href="#">2</a>

<b>Index</b>	<a href="#">4</a>
--------------	-------------------

---

rtc-package	<i>Tree Clustering</i>
-------------	------------------------

---

## Description

RTC performs cluster analysis using trees. A tree approximates the probability density function from which data has been drawn. A set of such trees is inferred using a Metropolis-Hastings sampler, and returned as a partitioning of the parameter space into rectangular segments.

## Details

Package: rtc  
Type: Package  
Version: 0.1.0  
Date: 2014-08-14  
License: <sup>1</sup>Proprietary

RTC provides a single function `tree.clustering`, which performs clustering of a given dataset.

### Author(s)

Peter Kuma <pkuma@pixelfederation.com>

---

<code>tree.clustering</code>	<i>Tree Clustering</i>
------------------------------	------------------------

---

### Description

Perform tree clustering of a data frame.

### Usage

```
tree.clustering(
  ds,
  nsamples=1,
  burnin=100,
  maxiter=NULL,
  limits=NULL,
  fragment.size=NULL,
  max.segments=0,
  nruns=1
)
```

### Arguments

<code>ds</code>	Data frame containing the data the be clustered. Only numeric columns are supported at the moment.
<code>nsamples</code>	Number of samples to generate using the Metropolis-Hastings algorithm.
<code>burnin</code>	Burn-in period, i.e. the number of initial samples to be ignored.
<code>maxiter</code>	Maximum number of iterations of the Metropolis-Hastings algorithm. Use <code>maxiter</code> to ensure <code>tree.clustering</code> finishes in a reasonable time even when the accept rate is low. When 0, the number of iterations is unlimited. Default: $(nsamples + burnin) * 1000$ .
<code>limits</code>	Parameter space bounds. When NULL, defaults to the range of data in <code>ds</code> .
<code>fragment.size</code>	Vector specifying fragment size of parameters. When set, segment ranges are constrained to be a multiple of fragment size in their respective parameters. The vector must have the same length as the number of columns in <code>ds</code> .
<code>max.segments</code>	Maximum number of segments. When 0, the number of segments is unlimited.
<code>nruns</code>	Number of runs. The algorithm is run <code>nruns</code> times, and segmentations are concatenated in the result.

## Details

This function performs clustering by constructing a tree approximating the probability density function from which data has been drawn. The tree partitions the parameter space hierarchically into a number of rectangular segments, on which the probability density function is assumed to have a uniform distribution. Possible trees are sampled using the Metropolis-Hastings algorithm.

## Value

`tree.clustering` returns a list of segmentations from the MCMC sampler. Segmentation is a list with the following elements:

likelihood	Segmentation likelihood, i.e. the likelihood of the data being drawn from the segmentation.
segments	List of <i>segments</i> .

*Segment* is a list with the following elements:

NX	Number of data elements in the segment.
V	Volume of the segment calculated by multiplying the segment lengths over each parameters.
ranges	List of segment ranges over parameters. Range is a vector of two values: the lower and upper bound (resp.) of the segment in the respective parameter.

## Examples

```
data(trees)
tree.clustering(trees, fragment.size=c(0.1,1,0.1))
```

# Index

\*Topic **file**

tree.clustering, [2](#)

\*Topic **package**

rtc-package, [1](#)

rtc (rtc-package), [1](#)

rtc-package, [1](#)

tree.clustering, [2](#)