

# Package ‘streamMOA’

July 2, 2014

**Version** 0.1-1

**Date** 2014-06-17

**Title** stream - Interface to MOA Algorithms

**Description** This package provides an interface to algorithms implemented in the MOA (Massive On-line Analysis) framework for data stream mining.

**Depends** stream (>= 1.0-2), methods, rJava (>= 0.6-3)

**SystemRequirements** Oracle Java (>= 5.0)

**License** GPL-3

**Copyright** MOA code in inst/java/moa.jar is Copyright (C) The University of Waikato and distributed under the Apache License, version 2.0. All other code is Copyright (C) Matthew Bolanos, John Forrest and Michael Hahsler

**Author** Michael Hahsler [aut, cre, cph], Matthew Bolanos [aut, cph], John Forrest [aut, cph]

**Maintainer** Michael Hahsler <mhahsler@lyle.smu.edu>

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2014-06-17 10:54:36

## R topics documented:

DSC_CluStream	2
DSC_ClusTree	3
DSC_DenStream	4
DSC_MOA	6
DSD_RandomRBFGeneratorEvents	6

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

---

DSC\_CluStream

*CluStream Data Stream Clusterer*

---

### Description

Class implements the CluStream cluster algorithm for data streams.

### Usage

```
DSC_CluStream(m = 100, horizon = 1000, t = 2, k=NULL)
```

### Arguments

m	Defines the maximum number of micro-clusters used in CluStream
horizon	Defines the time window to be used in CluStream
t	Maximal boundary factor (=Kernel radius factor). When deciding to add a new data point to a micro-cluster, the maximum boundary is defined as a factor of t of the RMS deviation of the data points in the micro-cluster from the centroid.
k	Number of macro-clusters to produce using weighted k-means. NULL disables automatic reclustering.

### Details

This is an interface to the MOA implementation of CluStream.

CluStream applies a weighted k-means algorithm for reclustering (see Examples section below).

### Value

An object of class DSC\_CluStream (subclass of DSC\_Micro, DSC\_MOA and DSC).

### References

Aggarwal CC, Han J, Wang J, Yu PS (2003). "A Framework for Clustering Evolving Data Streams." In "Proceedings of the International Conference on Very Large Data Bases (VLDB '03)," pp. 81-92.

Bifet A, Holmes G, Pfahringer B, Kranen P, Kremer H, Jansen T, Seidl T (2010). MOA: Massive Online Analysis, a Framework for Stream Classification and Clustering. In Journal of Machine Learning Research (JMLR).

### See Also

[DSC](#), [DSC\\_Micro](#), [DSC\\_MOA](#)

## Examples

```
# 3 clusters with 5% noise
dsd <- DSD_Gaussians(k=3, d=2, noise=.05)

# cluster with CluStream
dsc <- DSC_CluStream(m=50)
cluster(dsc, dsd, 500)
dsc

# plot micro-clusters
plot(dsc, dsd)

# plot assignment area (micro-cluster radius)
plot(dsc, dsd, assignment=TRUE, weights=FALSE)

# reclustering. Use weighted k-means for CluStream
kmeans <- DSC_Kmeans(k=3, weighted=TRUE)
recluster(kmeans, dsc)
plot(kmeans, dsd, type="both")

# use k-means automatically
dsc <- DSC_CluStream(m=50, k=3)
cluster(dsc, dsd, 500)
dsc

plot(dsc, dsd, type="both")
```

---

DSC\_ClusTree

*ClusTree Data Stream Clusterer*

---

## Description

Class implements the ClusTree cluster algorithm for data streams.

## Usage

```
DSC_ClusTree(horizon = 1000, maxHeight = 8, lambda = NULL)
```

## Arguments

horizon	Range of the (time) window.
maxHeight	The maximum height of the tree.
lambda	number used to override computed lambda (decay).

## Details

This is an interface to the MOA implementation of ClusTree.

**Value**

An object of class DSC\_ClusTree (subclass of DSC, DSC\_MOA, DSC\_Micro).

**References**

Philipp Kranen, Ira Assent, Corinna Baldauf, and Thomas Seidl. 2009. Self-Adaptive Anytime Stream Clustering. In Proceedings of the 2009 Ninth IEEE International Conference on Data Mining (ICDM '09). IEEE Computer Society, Washington, DC, USA, 249-258. DOI=10.1109/ICDM.2009.47 <http://dx.doi.org/10.1109/ICDM.2009.47>

Bifet A, Holmes G, Pfahringer B, Kranen P, Kremer H, Jansen T, Seidl T (2010). MOA: Massive Online Analysis, a Framework for Stream Classification and Clustering. In Journal of Machine Learning Research (JMLR).

**See Also**

[DSC](#), [DSC\\_Micro](#), [DSC\\_MOA](#)

**Examples**

```
# 3 clusters with 5% noise
dsd <- DSD_Gaussians(k=3, d=2, noise=0.05)

dsc <- DSC_ClusTree(maxHeight=3)
cluster(dsc, dsd, 500)
dsc

# plot micro-clusters
plot(dsc, dsd)

# reclustering
kmeans <- DSC_Kmeans(k=3)
recluster(kmeans,dsc)
plot(kmeans, dsd, type="both")
```

---

DSC\_DenStream

*DenStream Data Stream Clusterer*

---

**Description**

Class implements the DenStream cluster algorithm for data streams.

**Usage**

```
DSC_DenStream(epsilon, mu = 1, beta = 0.2, lambda = 0.001,
  initPoints = 100, offline = 2, processingSpeed=1)
```

**Arguments**

epsilon	defines the epsilon neighbourhood which is the maximal radius of micro-clusters ( $r \leq \epsilon$ ). Range: 0 to 1.
mu	minimum weight $w$ for core-micro-clusters ( $w \geq \mu$ ). Range: 0 to $\max(\text{double})$ .
beta	multiplier for $\mu$ to detect outlier micro-clusters given their weight $w$ ( $w < \beta \times \mu$ ). Range: 0 to 1
lambda	decay constant.
initPoints	number of points to use for initialization via DBSCAN.
offline	offline multiplier for epsilon. Range: between 2 and 20). Used for reachability reclustering
processingSpeed	Number of incoming points per time unit (important for decay). Range: between 1 and 1000.

**Details**

Interface to the DenStream implementation in MOA.

DenStream applies weighted DBSCAN for reclustering (see Examples section below).

**Value**

An object of class DSC\_DenStream (subclass of DSC, DSC\_MOA, DSC\_Micro)

**References**

Cao F, Ester M, Qian W, Zhou A (2006). Density-Based Clustering over an Evolving Data Stream with Noise. In Proceedings of the 2006 SIAM International Conference on Data Mining, pp 326-337. SIAM.

Bifet A, Holmes G, Pfahringer B, Kranen P, Kremer H, Jansen T, Seidl T (2010). MOA: Massive Online Analysis, a Framework for Stream Classification and Clustering. In Journal of Machine Learning Research (JMLR).

**See Also**

[DSC](#), [DSC\\_Micro](#), [DSC\\_MOA](#)

**Examples**

```
# 3 clusters with 5% noise
dsd <- DSD_Gaussians(k=3, d=2, noise=0.05)

dsc <- DSC_DenStream(epsilon=.05)
cluster(dsc, dsd, 500)
dsc

# plot micro-clusters
plot(dsc, dsd)
```

```

# plot the micro-cluster assignment area
plot(dsc, dsd, assignment=TRUE, weights=FALSE)

# show macro-clusters (using density reachability with epsilon x offline)
plot(dsc, dsd, type="both")

# reclustering. DenStream micro-clusters with k-means instead
km <- DSC_Kmeans(k=3, weighted=TRUE)
recluster(km, dsc)
plot(km, dsd, type="both")

```

---

DSC\_MOA

*DSC\_MOA Class*


---

### Description

An abstract class that inherits from the base class DSC and provides the common functions needed to interface MOA clusterers.

### Details

DSC\_MOA classes operate in a different way in that the centers of the micro-clusters have to be extracted from the underlying Java object. This is done by using rJava to perform method calls directly in the JRI and converting the multi-dimensional Java array into a local R data type.

### References

MOA: Massive Online Analysis, a Framework for Stream Classification and Clustering Albert Bifet, Geoff Holmes, Bernhard Pfahringer, Philipp Kranen, Hardy Kremer, Timm Jansen, Thomas Seidl. Journal of Machine Learning Research (JMLR).

### See Also

[DSC](#)

---

DSD\_RandomRBFGeneratorEvents

*Random RBF Generator Events Data Stream Generator*


---

### Description

A class that generates random data based on RandomRBFGeneratorEvents implemented in MOA.

**Usage**

```
DSD_RandomRBFGeneratorEvents(k = 3, d = 2, numClusterRange = 3L,
  kernelRadius = 0.07, kernelRadiusRange = 0, densityRange = 0,
  speed = 100L, speedRange = 0L, noiseLevel = 0.1,
  noiseInCluster = FALSE, eventFrequency = 30000L,
  eventMergeSplitOptions = FALSE, eventDeleteCreate = FALSE,
  modelSeed = NULL, instanceSeed = NULL)
```

**Arguments**

k	The number of clusters.
d	The dimensionality of the data.
numClusterRange	Number of clusters.
kernelRadius	The average radius of the micro-clusters.
kernelRadiusRange	Radius range of the micro-clusters.
densityRange	Density range.
speed	Speed.
speedRange	Speed range.
noiseLevel	Noise level.
noiseInCluster	Noise in cluster.
eventFrequency	Frequency of events.
eventMergeSplitOptions	Merge and split?
eventDeleteCreate	Delete and create?
modelSeed	Random seed for the model.
instanceSeed	Random seed for the instances.

**Details**

There are an assortment of parameters available for the underlying MOA data structure, however, we have currently limited the available parameters to the arguments above. Currently the `modelSeed` and `instanceSeed` are set to default values every time a `DSD_MOA` is created, therefore the generated data will be the same. Because of this, it is important to set the seed manually when different data is needed.

The default behavior is to create a data stream with 3 clusters and concept drift. The locations of the clusters will change slightly, and they will merge with one another as time progresses.

**Value**

An object of class `DSD_RandomRBFGeneratorEvent` (subclass of `DSD_MOA`, `DSD`).

**References**

MOA: Massive Online Analysis, a Framework for Stream Classification and Clustering Albert Bifet, Geoff Holmes, Bernhard Pfahringer, Philipp Kranen, Hardy Kremer, Timm Jansen, Thomas Seidl. Journal of Machine Learning Research (JMLR).

**See Also**

[DSD](#)

**Examples**

```
dsd <- DSD_RandomRBFGeneratorEvents()

## Not run:
animate_data(dsd, n=5000, pointInterval=100, xlim=c(0,1), ylim=c(0,1))

## End(Not run)
```



# Index

CluStream (DSC\_CluStream), [2](#)  
clustream (DSC\_CluStream), [2](#)  
ClusTree (DSC\_ClusTree), [3](#)  
clustree (DSC\_ClusTree), [3](#)

DenStream (DSC\_DenStream), [4](#)  
denstream (DSC\_DenStream), [4](#)  
DSC, [2](#), [4-6](#)  
DSC\_CluStream, [2](#)  
DSC\_ClusTree, [3](#)  
DSC\_DenStream, [4](#)  
DSC\_Micro, [2](#), [4](#), [5](#)  
DSC\_MOA, [2](#), [4](#), [5](#), [6](#)  
DSD, [8](#)  
DSD\_RandomRBFGeneratorEvents, [6](#)