

ECL DOCUMENTATION



[Go Up](#)

Name	DBSCAN
Version	1.0.0
Description	DBSCAN Clustering Method
License	http://www.apache.org/licenses/LICENSE-2.0
Copyright	Copyright (C) 2019 HPCC Systems
Authors	HPCCSystems
DependsOn	ML_Core 3.2.2
Platform	7.4.0

OVERVIEW

DBSCAN

DBSCAN Clustering Method

Table of Contents

[DBSCAN.ecl](#)

Scalable Parallel DBSCAN Clustering Algorithm Implementation based on [1]

[DBSCAN_Types.ecl](#)

DBSCAN

[Go Up](#)

IMPORTS

ML_Core | ML_Core.Types | DBSCAN_Types | std.system.Thorlib | internal.locCluster |
internal.globalMerge |

DESCRIPTIONS

DBSCAN

/ EXPORT	DBSCAN
<pre>(REAL8 eps = 0.0, UNSIGNED4 minPts = 2, STRING dist = 'euclidian', SET OF REAL8 dist_params = [])</pre>	

Scalable Parallel DBSCAN Clustering Algorithm Implementation based on [1]. It's an extension of the original DBSCAN algorithm [2] to meet the challenge of clustering problems on the Big Data platforms such as HPCC Systems. Based on the algorithm, this implementation has three stages: 1. Data preparation, 2. Local clustering, 3. Global Merge. The details of stage 2 and 3 can be found in the /internal/locCluster.ecl and /internal/globalMerge.ecl. Reference [1] Patwary, Mostofa Ali, et al. "A new scalable parallel DBSCAN algorithm using the disjoint-set data structure." Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis. IEEE Computer Society Press, 2012. [2] Ester, Martin, et al. "A density-based algorithm for discovering clusters in large spatial databases with noise." Kdd. Vol. 96. No. 34. 1996.

PARAMETER **eps** ||| REAL8 — the maximum distance threshold to be considered as a neighbor of the other. Default value is 0.0.

PARAMETER **minPts** ||| UNSIGNED4 — the minimum number of points required for a point to become a core point. Default value is 2.

PARAMETER dist ||| STRING — a string describing the distance metrics used to calculate the distance between a pair of points. Default value is 'euclidean'. Other supported distance metrics includes 'cosine', 'haversine', 'chebyshev', 'manhattan', 'minkowski'.

PARAMETER dist_params ||| SET (REAL8) — a set of parameters for distance metrics that need extra setup. Default value is [] which should fit for most cases.

Children

1. **fit** : Fit function performs DBSCAN clustering on a dataset (ds) to find clusters and the cluster index (Label) of each sample in the dataset
2. **Num_Clusters** : Num_Clusters Provides the number of clusters that the given dataset will be divided into when clustered by the DBSCAN algorithm
3. **Num_Outliers** : Num_Outliers Provides the number of outliers that the given dataset will have when clustered by the DBSCAN algorithm

FIT

DBSCAN /

<code>DATASET(ML_Core.Types.ClusterLabels)</code>	fit
<code>(DATASET(Types.NumericField) ds)</code>	

Fit function performs DBSCAN clustering on a dataset (ds) to find clusters and the cluster index (Label) of each sample in the dataset.

PARAMETER ds ||| TABLE (NumericField) — The dataset in NumericField format to be clustered.

RETURN TABLE ({ UNSIGNED2 wi , UNSIGNED8 id , UNSIGNED8 label }) — result in ML_Core.Types.ClusterLabels format describing the cluster index of each sample.

SEE ML_Core.Types.NumericField, ML_Core.Types.ClusterLabels

NUM_CLUSTERS

DBSCAN /

<code>DATASET(Files.l_num_clusters)</code>	Num_Clusters
<code>(DATASET(ML_Core.Types.ClusterLabels) ds)</code>	

Num_Clusters Provides the number of clusters that the given dataset will be divided into when clustered by the DBSCAN algorithm.

PARAMETER **ds** ||| TABLE (ClusterLabels) — A dataset with cluster index information. Usually it's the result of Fit function.

RETURN **TABLE ({ UNSIGNED4 wi , UNSIGNED4 num })** — DATASET(l_num_clusters) The number of clusters, per work item.

SEE DBSCAN_Types.l_num_clusters

NUM_OUTLIERS

DBSCAN /

<code>DATASET(Files.l_num_clusters)</code>	Num_Outliers
<code>(DATASET(ML_Core.Types.ClusterLabels) ds)</code>	

Num_Outliers Provides the number of outliers that the given dataset will have when clustered by the DBSCAN algorithm.

PARAMETER **ds** ||| TABLE (ClusterLabels) — A dataset with cluster index information. Usually it's the result of Fit function.

RETURN **TABLE ({ UNSIGNED4 wi , UNSIGNED4 num })** — DATASET(l_num_clusters) The number of outliers, per work item.

SEE DBSCAN_Types.l_num_clusters

DBSCAN_Types

[Go Up](#)

IMPORTS

ML_Core | ML_Core.Types |

DESCRIPTIONS

DBSCAN_TYPES

DBSCAN_Types

No Documentation Found

Children

1. [l_stage1](#) : l_stage1 extends NumericField by adding a nodeID field and a fields field for the data preparation of stage 2 local clustering
2. [l_stage2](#) : l_stage2 is the data structure for the local clustering of locDBSCAN() function
3. [l_stage3](#) : l_stage3 is the data structure for global merging of globalMerge() function
4. [l_num_clusters](#) : l_num_clusters This record structure holds the results of functions that return statistics about the clusters formed in DBSCAN clustering, that is, it is the result structure for num_clusters and num_outliers

L_STAGE1

[DBSCAN_Types](#) /

l_stage1

`l_stage1` extends `NumericField` by adding a `nodeID` field and a `fields` field for the data preparation of stage 2 local clustering. The `nodeID` field records the physical cluster node to which the data point is assigned to. The `fields` field allows each data point to be stored as a vector for embedded C++ computing at stage 2.

FIELD wi ||| UNSIGNED2 — The work-item identifier for this cell.

FIELD id ||| UNSIGNED8 — The record-identifier for this cell.

FIELD number ||| UNSIGNED4 — The field number (i.e. `featureId`) of this cell.

FIELD value ||| REAL8 — The numerical value of this cell.

FIELD nodeID ||| UNSIGNED4 — The physical cluster node to which the data point is assigned to. It's 0-based index by default.

FIELD fields ||| SET (REAL4) — The SET of feature values of each data point. It's similar to the vector definition in C++.

SEE `ML_Core.Types.NumericField`.

L_STAGE2

[DBSCAN_Types](#) /

l_stage2

`l_stage2` is the data structure for the local clustering of `locDBSCAN()` function.

FIELD wi ||| UNSIGNED2 — The work-item identifier for this cell.

FIELD id ||| UNSIGNED8 — The record-identifier for this cell.

FIELD parentID ||| UNSIGNED8 — the largest core points a data point belongs to.

FIELD nodeID ||| UNSIGNED8 — The physical cluster node to which the data point is assigned to.

FIELD fields ||| SET (REAL4) — The SET of feature values of each data point. It's similar to the vector definition in C++.

FIELD if_local ||| BOOLEAN — TRUE if the data point is physically located at the current cluster. Otherwise FALSE.

FIELD if_core ||| BOOLEAN — TRUE if the data point is a core point. Otherwise FALSE.

L_STAGE3

[DBSCAN_Types /](#)

<u>l_stage3</u>

`l_stage3` is the data structure for global merging of `globalMerge()` function.

FIELD wi ||| UNSIGNED4 — The work-item identifier for this cell.

FIELD id ||| UNSIGNED4 — The record-identifier for this cell.

FIELD parentID ||| UNSIGNED4 — the largest core points a data point belongs to.

FIELD nodeID ||| UNSIGNED4 — The physical cluster node it's located. It's 0-based index by default.

FIELD if_local ||| BOOLEAN — TRUE if the data point is physically located at the current cluster. Otherwise FALSE.

FIELD if_core ||| BOOLEAN — TRUE if the data point is a core point. Otherwise FALSE.

SEE `ML_Core.Types.NumericField`.

L_NUM_CLUSTERS

[DBSCAN_Types /](#)

<u>l_num_clusters</u>

`l_num_clusters` This record structure holds the results of functions that return statistics about the clusters formed in DBSCAN clustering, that is, it is the result structure for `num_clusters` and `num_outliers`. It contains the value of the statistic, per work-item

FIELD wi ||| UNSIGNED4 — The work-item identifier

FIELD num ||| UNSIGNED4 — The value of the statistic (Number of clusters / outliers)
