Grande Omega

1.0.0

Generated by Doxygen 1.8.14

Contents

1 Todo List

Class KnearestClassification

In order de reduce the time complexity, a KD Tree could be implemented. Searching a KD Tree for nearest neighbours has a time complexity of `O(log n)`

2 Namespace Index

2.1 Namespace List

Here is a list of all documented namespaces with brief descriptions:

3 Hierarchical Index

3.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

4 Class Index

4.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

5 Namespace Documentation

5.1 DataTools Namespace Reference

Namespaces

- 1

Classes

- class GenericVector
  N dimensional vector.
- class MatrixUtils
  Utilities for Math algebra.
- class Vector2
  Two dimensional vector.

5.2 DataTools.classification Namespace Reference

Classes

- class KnearestClassification
  Classification algorithm using K Nearest.

5.3 DataTools.clustering Namespace Reference

Classes

- class **ClusterPoint**
  Vector that is assigned to a cluster.
- class Dbscan
  Class featuring the DB Scan algorithm.
- class Kmeans
  Class featuring de KMeans algorithm.

5.4 DataTools.correlation Namespace Reference

Classes

- class Correlation
- class PearsonCorrelation
- class SpearmanCorrelation

5.5 DataTools.regression Namespace Reference

Classes

- class LinearRegression
- class PolynomialRegression

5.6 DataTools.utils Namespace Reference

Classes

- class PriorityQue
- class QueItem
- class Utils

5.7 Smoothing Namespace Reference

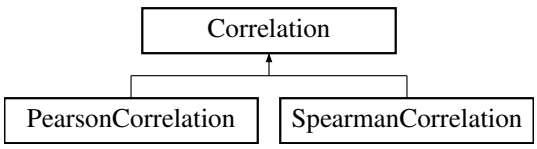
Classes

- class DES

6 Class Documentation

6.1 Correlation Class Reference

Inheritance diagram for Correlation:



Public Member Functions

- abstract double **GetCorrelationCoefficient** ()

Protected Member Functions

- **Correlation** (IEnumerable< Vector2 > data)

Protected Attributes

- readonly IEnumerable< Vector2 > **Data**

6.2 Dbscan Class Reference

Class featuring the DB Scan algorithm.

Public Member Functions

- Dbscan (float eps, int minPoints, IEnumerable< GenericVector > data)

Properties

- Dictionary< int, IEnumerable< GenericVector > > **DataClusters** `[get]`

6.2.1 Detailed Description

Class featuring the DB Scan algorithm.

The DB Scan algorithm is capable of clustering vectors of `n` dimensions. As input, it needs a radius within neighbours should be together, the minimum amount of point in a cluster and ofcourse the dataset, in this case, a `IEnumerable<GenericVector>`. As output it produces a Dictionary DataClusters with the key as cluster and the value the vectors belonging to the cluster.

6.2.2 Constructor & Destructor Documentation

6.2.2.1 Dbscan()

```
Dbscan (
            float eps,
            int minPoints,
            IEnumerable< GenericVector > data )
```

Constructor for DB Scan

Parameters

eps	Radius within neighbours should be
minPoints	Min amount of points in a cluster
data	Dataset to cluster

6.3 DES Class Reference

Static Public Member Functions

- static double [ ] **ForeCast** (double[ ] values, int start, int end, double alpha, double beta)
- static Tuple< double, double > **FindBestAlphaBeta** (double[ ] values, double stepSize)

6.4 GenericVector Class Reference

N dimensional vector.

Public Member Functions

- **GenericVector** (int size)
- **GenericVector** (params double[ ] args)
- double [ ] **ToArray** ()
- override string **ToString** ()
- Vector2 **ToVector2** (int indexOne=0, int indexTwo=1)

Static Public Member Functions

- static GenericVector **Sum** (GenericVector a, GenericVector b)
- static GenericVector **Devide** (GenericVector a, int devider)
- static double **Distance** (GenericVector a, GenericVector b)

Public Attributes

- int **Size** => _points.Length
- double **BiggestPoint** => _points.Max()
- double **this[int x]** => _points[x]

6.4.1 Detailed Description

N dimensional vector.

This class is a vector with N dimensions. The amount of dimensions is created upon initialization and can not be changed afterwards. It implements some basic vector algebra like summation, multiplication and deviding.

6.5 Kmeans Class Reference

Class featuring de KMeans algorithm.

Public Member Functions

- Kmeans (int k, int iterations, IEnumerable< GenericVector > dataSet)

Properties

- Dictionary< int, IEnumerable< GenericVector > > **DataClusters** `[get]`

6.5.1 Detailed Description

Class featuring de KMeans algorithm.

KMeans is a clustering algorithm. As input it needs `k` (the amount of clusters), the max amount of iterations and the dataset to cluster. It outputs the clustered values.

6.5.2 Constructor & Destructor Documentation

6.5.2.1 Kmeans()

```
Kmeans (
            int k,
            int iterations,
            IEnumerable< GenericVector > dataSet )
```

Constructor for Kmeans

Parameters

k	Amount of clusters
iterations	Max amount of iterations
dataSet	The dataset to cluster

6.6 KnearestClassification Class Reference

Classification algorithm using K Nearest.

Public Member Functions

- **KnearestClassification** (Dictionary< int, IEnumerable< GenericVector >> traingingData, int k)
- int **ClassifyPoint** (GenericVector point)

6.6.1 Detailed Description

Classification algorithm using K Nearest.

K Nearest is a classification algorithm that uses a training set in order te classify new points. To do so, K Nearest computes the distance from the new point to all the points in the training set. With a time complexity of `O(m*n)` where `n` is the length of the training set and `m` is the length of the set of new points, this a CPU heavy operation.

Todo In order de reduce the time complexity, a KD Tree could be implemented. Searching a KD Tree for nearest neighbours has a time complexity of `O(log n)`

6.7 LinearRegression Class Reference

Public Member Functions

- **LinearRegression** (IEnumerable< Vector2 > data)
- IEnumerable< GenericVector > **GetLinearRegressionLine** ()

Public Attributes

- double **Slope** => GetSlope()
- double **YIntercept** => MeanY - (Slope * MeanX)

6.8 MatrixUtils Class Reference

Utilities for Math algebra.

Static Public Member Functions

- static double [ ][ ] **MatrixInverse** (double[ ][ ] matrix)
- static double [ ][ ] **MatrixCreate** (int rows, int cols)

6.8.1 Detailed Description

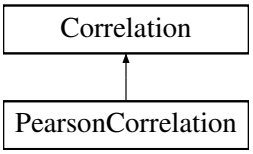
Utilities for Math algebra.

!NOTE the code used in this class is written by James D. McCraffey. All credites goes to him. More info (https://jamesmccaffrey.wordpress.com/2015/03/06/inverting-a-matrix-using-c/)

In this class are included several Matrix functions, that supports the matrix inverse function.

6.9 PearsonCorrelation Class Reference

Inheritance diagram for PearsonCorrelation:



Public Member Functions

- **PearsonCorrelation** (IEnumerable< Vector2 > data)
- override double **GetCorrelationCoefficient** ()

Additional Inherited Members

6.10 PolynomialRegression Class Reference

Public Member Functions

- **PolynomialRegression** (IEnumerable< Vector2 > data, int polynomialDegree)
- double **PredictPoint** (double x)
- GenericVector [ ] **GetPolynomialPoints** ()

6.11 PriorityQue< T > Class Template Reference

Public Member Functions

- **PriorityQue** (IEnumerable< Tuple< double, T >> entry)
- void **Insert** (double priority, T queItem)
- QueItem< T > **Peek** ()
- QueItem< T > **Pop** ()
- List< Tuple< double, T > > **ToList** ()

Public Attributes

- int **Count** => _entries.Count
- bool **IsEmpty** => Count == 0

6.12 QueItem< T > Class Template Reference

Public Member Functions

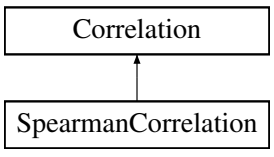
- **QueItem** (Tuple< double, T > queItem)

Public Attributes

- double **Priority** => _queItem.Item1
- T **Item** => _queItem.Item2

6.13 SpearmanCorrelation Class Reference

Inheritance diagram for SpearmanCorrelation:



Public Member Functions

- **SpearmanCorrelation** (IEnumerable< Vector2 > data)
- override double **GetCorrelationCoefficient** ()

Additional Inherited Members

6.14 Utils Class Reference

Static Public Member Functions

- static void **Times** (this int count, System.Action action)

6.15 Vector2 Class Reference

Two dimensional vector.

Public Member Functions

- **Vector2** (double x, double y)

Properties

- double **X** `[get, set]`
- double **Y** `[get, set]`

6.15.1 Detailed Description

Two dimensional vector.

Two dimensional vector having a X and Y dimension.

Index