kmeans

Generated by Doxygen 1.13.2

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

Chapter 2

Class Index

2.1 Class List

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

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

Chapter 4

Namespace Documentation

4.1 DM Namespace Reference

Classes

- class DistanceMetrics

  Class providing distance metric functions.

4.2 Kmeans Namespace Reference

Classes

- class Kmeans

  Class implementing the K-Means clustering algorithm.

4.3 KmeansParser Namespace Reference

Classes

- class Reader

  Class to read and parse data points from a file.

Chapter 5

Class Documentation

5.1 DM::DistanceMetrics Class Reference

Class providing distance metric functions.

```
#include <distanceMetrics.hpp>
```

Public Member Functions

- double euclideanDistance (const std::vector< double > &vec1, const std::vector< double > &vec2)

  Computes the Euclidean distance between two vectors.

5.1.1 Detailed Description

Class providing distance metric functions.

5.1.2 Member Function Documentation

5.1.2.1 euclideanDistance()

```
double DM::DistanceMetrics::euclideanDistance (
            const std::vector< double > & vec1,
            const std::vector< double > & vec2)
```

Computes the Euclidean distance between two vectors.

Parameters

vec1	First vector.
vec2	Second vector.

Returns

Euclidean distance between the vectors.

The documentation for this class was generated from the following files:

- include/distanceMetrics.hpp
- src/distanceMetrics.cpp

5.2 Kmeans::Kmeans Class Reference

Class implementing the K-Means clustering algorithm.

```
#include <kmeans.hpp>
```

Public Member Functions

- std::vector< std::vector< double > > getInitialCenters (const std::vector< std::vector< double > > &points, const int &k)

  Selects initial cluster centers from the data points.
- std::vector< std::vector< double > > computeDistance (const std::vector< std::vector< double > > &points, const std::vector< std::vector< double > > &centers, double(DM::DistanceMetrics::*func)(const std::vector< double > &, const std::vector< double > &), DM::DistanceMetrics &obj)

  Computes distances between points and centers using a specified metric.
- int minIndex (const std::vector< double > &distances)

  Finds the index of the minimum value in a distance vector.
- std::vector< std::vector< double > > computeNewCenters (const std::vector< std::vector< double > > &points, const std::vector< std::vector< double > > &distances, int k)

  Computes new cluster centers based on point assignments.
- std::vector< double > add (const std::vector< double > &a, const std::vector< double > &b)

  Adds two vectors element-wise.
- std::vector< double > div (const std::vector< double > &a, int n)

  Divides a vector by a scalar.

5.2.1 Detailed Description

Class implementing the K-Means clustering algorithm.

5.2.2 Member Function Documentation

5.2.2.1 add()

```
std::vector< double > Kmeans::Kmeans::add (
            const std::vector< double > & a,
            const std::vector< double > & b)
```

Adds two vectors element-wise.

Parameters

a	First vector.
b	Second vector.

Returns

Resulting vector from the addition.

5.2.2.2 computeDistance()

```
std::vector< std::vector< double > > Kmeans::Kmeans::computeDistance (
            const std::vector< std::vector< double > > & points,
            const std::vector< std::vector< double > > & centers,
            double(DM::DistanceMetrics::* func )(const std::vector< double > &, const std↩
::vector< double > &),
            DM::DistanceMetrics & obj)
```

Computes distances between points and centers using a specified metric.

Parameters

points	List of data points.
centers	Current cluster centers.
func	Pointer to the distance metric function.
obj	Reference to the DistanceMetrics object.

Returns

A vector of distance vectors, one per point.

5.2.2.3 computeNewCenters()

```
std::vector< std::vector< double > > Kmeans::Kmeans::computeNewCenters (
            const std::vector< std::vector< double > > & points,
            const std::vector< std::vector< double > > & distances,
            int k)
```

Computes new cluster centers based on point assignments.

Parameters

points	List of data points.
distances	Distances from points to centers.
k	Number of clusters.

Returns

Updated cluster centers.

5.2.2.4 div()

```
std::vector< double > Kmeans::Kmeans::div (
            const std::vector< double > & a,
            int n)
```

Divides a vector by a scalar.

Parameters

a	Vector to divide.
n	Scalar denominator.

Returns

Resulting vector after division.

5.2.2.5 getInitialCenters()

```
std::vector< std::vector< double > > Kmeans::Kmeans::getInitialCenters (
            const std::vector< std::vector< double > > & points,
            const int & k)
```

Selects initial cluster centers from the data points.

Parameters

points	List of all data points.
k	Number of clusters.

Returns

Initial centers as a vector of vectors.

5.2.2.6 minIndex()

```
int Kmeans::Kmeans::minIndex (
            const std::vector< double > & distances)
```

Finds the index of the minimum value in a distance vector.

Parameters

distances	Vector of distances.

Returns

Index of the minimum distance.

The documentation for this class was generated from the following files:

- include/kmeans.hpp
- src/kmeans.cpp

5.3 KmeansParser::Reader Class Reference

Class to read and parse data points from a file.

```
#include <read.hpp>
```

Public Member Functions

- Reader (std::string fileName)

  Constructor for the Reader class.
- std::vector< std::vector< double > > readAndParse ()

  Reads and parses the file into a vector of points.

5.3.1 Detailed Description

Class to read and parse data points from a file.

5.3.2 Constructor & Destructor Documentation

5.3.2.1 Reader()

```
KmeansParser::Reader::Reader (
            std::string fileName)
```

Constructor for the Reader class.

Parameters

fileName	Name of the file to read.

5.3.3 Member Function Documentation

5.3.3.1 readAndParse()

```
std::vector< std::vector< double > > KmeansParser::Reader::readAndParse ()
```

Reads and parses the file into a vector of points.

Returns

A vector of vectors, where each inner vector represents a point.

The documentation for this class was generated from the following files:

- include/read.hpp
- src/read.cpp

Chapter 6

File Documentation

6.1 app/main.cpp File Reference

Main entry point for the K-Means clustering algorithm in C++.

```
#include <vector>
#include <iostream>
#include <chrono>
#include "read.hpp"
#include "distanceMetrics.hpp"
#include "kmeans.hpp"
```

Functions

- int main ()

  Main function to execute K-Means clustering.

6.1.1 Detailed Description

Main entry point for the K-Means clustering algorithm in C++.

This file drives the K-Means clustering process by reading data points from a file, initializing centers, and iterating until convergence. It measures and outputs the execution time.

6.1.2 Function Documentation

6.1.2.1 main()

```
int main ()
```

Main function to execute K-Means clustering.

Returns

0 on successful execution.

6.2 include/distanceMetrics.hpp File Reference

Header file for distance metric calculations.

```
#include <vector>
```

Classes

- class DM::DistanceMetrics

  Class providing distance metric functions.

Namespaces

- namespace DM

6.2.1 Detailed Description

Header file for distance metric calculations.

Defines the DistanceMetrics class in the DM namespace, providing methods for distance computation.

6.3 distanceMetrics.hpp

Go to the documentation of this file.

```


#pragma once
#include <vector>

namespace DM {
    class DistanceMetrics {
        public:
            double euclideanDistance(const std::vector<double>& vec1, const std::vector<double>& vec2);
    };
}
```

6.4 include/kmeans.hpp File Reference

Header file for the K-Means clustering algorithm implementation.

```
#include <vector>
#include "distanceMetrics.hpp"
```

Classes

- class Kmeans::Kmeans

  Class implementing the K-Means clustering algorithm.

Namespaces

- namespace Kmeans

6.4.1 Detailed Description

Header file for the K-Means clustering algorithm implementation.

Defines the Kmeans class in the Kmeans namespace, providing methods for clustering.

6.5 kmeans.hpp

Go to the documentation of this file.

```


#pragma once
#include <vector>
#include "distanceMetrics.hpp"

namespace Kmeans {
    class Kmeans {
        public:
            std::vector<std::vector<double>> getInitialCenters(const std::vector<std::vector<double>>& points, const int& k);

            std::vector<std::vector<double>> computeDistance(const std::vector<std::vector<double>>& points,
                const std::vector<std::vector<double>>& centers,
                double (DM::DistanceMetrics::*func)(const std::vector<double>&, const std::vector<double>&),
                DM::DistanceMetrics& obj);

            int minIndex(const std::vector<double>& distances);

            std::vector<std::vector<double>> computeNewCenters(const std::vector<std::vector<double>>& points,
                const std::vector<std::vector<double>>& distances, int k);

            std::vector<double> add(const std::vector<double>& a, const std::vector<double>& b);

            std::vector<double> div(const std::vector<double>& a, int n);
    };
}
```

6.6 include/read.hpp File Reference

Header file for reading and parsing input data for K-Means clustering.

```
#include <string>
#include <vector>
```

Classes

- class KmeansParser::Reader

    Class to read and parse data points from a file.

Namespaces

- namespace KmeansParser

6.6.1 Detailed Description

Header file for reading and parsing input data for K-Means clustering.

Defines the Reader class in the KmeansParser namespace, responsible for reading data points from a file.

6.7 read.hpp

Go to the documentation of this file.

```

#pragma once
#include <string>
#include <vector>

namespace KmeansParser {
    class Reader {
        private:
            std::string fileName{};
            std::vector<std::vector<double» all_points{};
        public:
            Reader(std::string fileName);

            std::vector<std::vector<double» readAndParse();
    };
}
```

6.8 src/distanceMetrics.cpp File Reference

Implementation of distance metric functions.

```
#include <cmath>
#include <vector>
#include "distanceMetrics.hpp"
```

Namespaces

- namespace DM

6.8.1 Detailed Description

Implementation of distance metric functions.

Provides the implementation for the DistanceMetrics class, specifically the Euclidean distance calculation.

6.9 src/kmeans.cpp File Reference

Implementation of the K-Means clustering algorithm.

```
#include <vector>
#include <limits.h>
#include "kmeans.hpp"
#include "distanceMetrics.hpp"
```

Namespaces

- namespace Kmeans

6.9.1 Detailed Description

Implementation of the K-Means clustering algorithm.

Provides the implementation for the Kmeans class, handling center initialization, distance computation, and center updates.

6.10 src/read.cpp File Reference

Implementation of file reading and parsing for K-Means clustering.

```
#include <string>
#include <fstream>
#include <vector>
#include <sstream>
#include "read.hpp"
```

Namespaces

- namespace KmeansParser

6.10.1 Detailed Description

Implementation of file reading and parsing for K-Means clustering.

Provides the implementation for the Reader class, reading data points from a file into a vector of vectors.

Index