PROBI

1.0

Generated by Doxygen 1.8.4

Wed Aug 28 2013 20:04:15

Contents

1 Main Page 1

2 Hierarchical Index 3

2.1 Class Hierarchy . . . 3

3 Class Index 5

3.1 Class List . . . 5

4 Class Documentation 7

4.1 AdaptiveSampling Class Reference . . . 7

4.1.1 Detailed Description . . . 7

4.1.2 Member Function Documentation . . . 7

4.1.2.1 computeCenterSet . . . 7

4.2 CenterOfGravity Class Reference . . . 8

4.2.1 Detailed Description . . . 8

4.2.2 Member Function Documentation . . . 8

4.2.2.1 cog . . . 8

4.3 DatastreamCoreset Class Reference . . . 8

4.3.1 Detailed Description . . . 9

4.3.2 Constructor & Destructor Documentation . . . 9

4.3.2.1 DatastreamCoreset . . . 9

4.4 EuclideanMetric Class Reference . . . 10

4.4.1 Detailed Description . . . 10

4.5 EuclideanNorm Class Reference . . . 11

4.5.1 Detailed Description . . . 11

4.6 EuclideanSpace Class Reference . . . 12

4.6.1 Detailed Description . . . 12

4.7 EuclideanSquaredMetric Class Reference . . . 12

4.7.1 Detailed Description . . . 13

4.8 EuclideanSquaredNorm Class Reference . . . 13

4.8.1 Detailed Description . . . 14

4.9 FastCoreset Class Reference . . . 14

4.9.1 Detailed Description . . . 15
4.9.2 Member Function Documentation . . . 15
4.9.2.1 computeCoreset . . . 15
4.9.2.2 getAllSamplesSize . . . 15
4.9.2.3 getK . . . 15
4.9.2.4 getKumarMedianIterations . . . 16
4.9.2.5 getMaxLloydClusteringIterations . . . 16
4.9.2.6 getWeiszfeldMedianIterations . . . 16
4.9.2.7 setAllSamplesSize . . . 16
4.9.2.8 setK . . . 16
4.9.2.9 setKumarMedianIterations . . . 16
4.9.2.10 setMaxLloydClusteringIterations . . . 16
4.9.2.11 setWeiszfeldMedianIterations . . . 17
4.10 Weiszfeld::IterationFailed Class Reference . . . 17
4.11 KMedian Class Reference . . . 17
4.11.1 Detailed Description . . . 18
4.12 KumarMedian Class Reference . . . 18
4.12.1 Detailed Description . . . 18
4.12.2 Member Function Documentation . . . 18
4.12.2.1 approximateOneMedian . . . 18
4.12.2.2 approximateOneMedianRounds . . . 18
4.13 LloydMedian Class Reference . . . 19
4.13.1 Detailed Description . . . 19
4.13.2 Member Function Documentation . . . 19
4.13.2.1 computeCenterSet . . . 19
4.14 LloydProbMedian Class Reference . . . 20
4.14.1 Detailed Description . . . 20
4.14.2 Member Function Documentation . . . 20
4.14.2.1 computeCenterSet . . . 20
4.15 MergeReduce< T > Class Template Reference . . . 20
4.15.1 Detailed Description . . . 21
4.15.2 Constructor & Destructor Documentation . . . 21
4.15.2.1 MergeReduce . . . 21
4.15.3 Member Function Documentation . . . 21
4.15.3.1 operator<< . . . 21
4.16 Metric< T > Class Template Reference . . . 21
4.16.1 Detailed Description . . . 21
4.17 Norm< T > Class Template Reference . . . 22
4.17.1 Detailed Description . . . 22
4.18 PKMedian Class Reference . . . 22

4.18.1 Detailed Description . . . 22
4.19 Point Class Reference . . . 22
4.19.1 Detailed Description . . . 23
4.20 ProbabilisticPoint Class Reference . . . 23
4.20.1 Detailed Description . . . 24
4.21 Randomness Class Reference . . . 24
4.21.1 Detailed Description . . . 24
4.22 Sampling Class Reference . . . 24
4.22.1 Detailed Description . . . 25
4.22.2 Member Function Documentation . . . 25
4.22.2.1 sampleWithoutReplacement . . . 25
4.22.2.2 sampleWithoutReplacementFast . . . 26
4.22.2.3 sampleWithReplacement . . . 26
4.22.2.4 sampleWithReplacement . . . 26
4.23 WeightedPoint Class Reference . . . 26
4.23.1 Detailed Description . . . 27
4.24 Weiszfeld Class Reference . . . 27
4.24.1 Detailed Description . . . 28

Index **29**

Chapter 1

Main Page

PROBI is a data stream algorithm for the probabilistic Euclidean k-median problem. This implementation is an heuristic and fast version of PROBI. It also features a second algorithm for the probabilistic Euclidean k-means problem.

Class hierarchy

FastCoreset implements PROBI. Points are stored and delivered to FastCoreset as instances of Point and ProbabilisticPoint. Metric<Point> is implemented by EuclideanMetric and EuclideanSquaredMetric for k-median and k-means, respectively. Norm<Point> is implemented by EuclideanNorm and EuclideanSquaredNorm. LloydMedian and LloydProbMedian are adaptions of Lloyd's algorithm for k-median and probabilistic k-median, respectively.

Building PROBI

The PROBI sample applications can be built by generating the project files with Premake (see below) in probi-environment and compiling them. A generated Makefile using GCC is provided for convenience.

Example: Generation of a Makefile using Linux

```
premake4 gmake
```

Four configurations are available

- "Debug" and "Release" for k-median
- "DebugKmeans" and "ReleaseKmeans" for k-means

Example: Compiling using the "Release" configuration

```
make config=release
```

Attention

Debug configurations need header files which are ordinarily available only on Unix based operating systems.

References

- PROBI website
- Premake website

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

AdaptiveSampling . . . 7
CenterOfGravity . . . 8
EuclideanSpace . . . 12
FastCoreset . . . 14
Weiszfeld::IterationFailed . . . 17
KMedian . . . 17
KumarMedian . . . 18
LloydMedian . . . 19
LloydProbMedian . . . 20
MergeReduce< T > . . . 20
MergeReduce< ProbabilisticPoint > . . . 20
DatastreamCoreset . . . 8
Metric< T > . . . 21
Metric< Point > . . . 21
EuclideanMetric . . . 10
EuclideanSquaredMetric . . . 12
Norm< T > . . . 22
Norm< Point > . . . 22
EuclideanNorm . . . 11
EuclideanSquaredNorm . . . 13
PKMedian . . . 22
Point . . . 22
WeightedPoint . . . 26
ProbabilisticPoint . . . 23
Randomness . . . 24
Sampling . . . 24
Weiszfeld . . . 27

Chapter 3

Class Index

3.1 Class List

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

AdaptiveSampling
: Weighted sampling for k-means++ and similar algorithms . . . 7

CenterOfGravity
: Computes the center of gravity / centroid / 1-means . . . 8

DatastreamCoreset
: Wraps FastCoreset into the Merge & Reduce framework . . . 8

EuclideanMetric
: Euclidean metric for Point objetcs . . . 10

EuclideanNorm
: Euclidean norm for Point objetcs . . . 11

EuclideanSpace
: Euclidean space operations for Point objetcs . . . 12

EuclideanSquaredMetric
: Euclidean squared metric for Point objetcs . . . 12

EuclideanSquaredNorm
: Euclidean squared norm for Point objetcs . . . 13

FastCoreset
: Fast implementation of PROBI . . . 14

Weiszfeld::IterationFailed . . . 17

KMedian
: K-median evaluation for Point objects . . . 17

KumarMedian
: 1-median approximation . . . 18

LloydMedian
: Adaption of Lloyd's algorithm for k-median . . . 19

LloydProbMedian
: Adaption of Lloyd's algorithm for probabilistic k-median . . . 20

MergeReduce< T >
: Merge & Reduce framework template . . . 20

Metric< T >
: Metric interface . . . 21

Norm< T >
: Norm interface . . . 22

PKMedian
: Probabilistic k-median evaluator . . . 22

Point
: Point class . . . 22

ProbabilisticPoint
Probabilistic point 23
Randomness
Encapsulates an STL random generator 24
Sampling
Sampling operations 24
WeightedPoint
Weighted point class 26
Weiszfeld
1-median approximation 27

Chapter 4

Class Documentation

4.1 AdaptiveSampling Class Reference

Weighted sampling for k-means++ and similar algorithms.

`#include <AdaptiveSampling.hpp>`

Public Member Functions

- **AdaptiveSampling** (std::function< Metric< Point > *() > createMetric)
- template<typename ForwardIterator >
  std::unique_ptr< std::vector
  < Point > > computeCenterSet (ForwardIterator begin, ForwardIterator end, size_t k, size_t n=0)

  Computes a center set.

4.1.1 Detailed Description

Weighted sampling for k-means++ and similar algorithms.

4.1.2 Member Function Documentation

4.1.2.1 template<typename ForwardIterator > std::unique_ptr< std::vector< Point > > AdaptiveSampling::computeCenterSet (ForwardIterator *begin*, ForwardIterator *end*, size_t *k*, size_t *n* = 0)

Computes a center set.

Parameters

begin	Input point set iterator: begin
end	Input point set iterator: end
k	Number of centers
n	Number of points (optional)

Returns

k centers

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

- AdaptiveSampling.hpp

- AdaptiveSampling.cpp

4.2 CenterOfGravity Class Reference

Computes the center of gravity / centroid / 1-means.

`#include <CenterOfGravity.hpp>`

Public Member Functions

- **CenterOfGravity** (std::function< Metric< Point > *() > createMetric)
- template<typename ForwardIterator >
  Point cog (ForwardIterator begin, ForwardIterator end)

  Computes the center of gravity / centroid / 1-means.

4.2.1 Detailed Description

Computes the center of gravity / centroid / 1-means.

4.2.2 Member Function Documentation

4.2.2.1 template<typename ForwardIterator > Point CenterOfGravity::cog (ForwardIterator *begin*, ForwardIterator *end*)

Computes the center of gravity / centroid / 1-means.

Parameters

begin	Input point set iterator: begin
end	Input point set iterator: end

Returns

Center of gravity

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

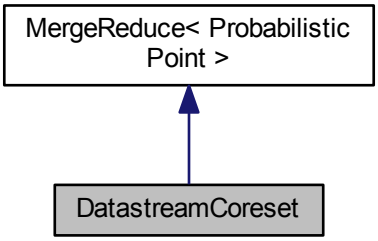
- CenterOfGravity.hpp
- CenterOfGravity.cpp

4.3 DatastreamCoreset Class Reference

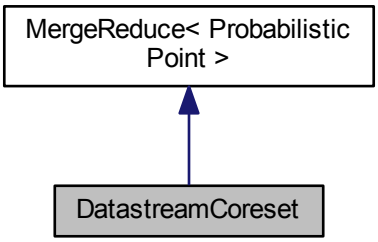
Wraps FastCoreset into the Merge & Reduce framework.

`#include <DatastreamCoreset.hpp>`

Inheritance diagram for DatastreamCoreset:



Collaboration diagram for DatastreamCoreset:



Public Member Functions

- DatastreamCoreset (FastCoreset *fastCoreset, int firstBucketSize)

4.3.1 Detailed Description

Wraps FastCoreset into the Merge & Reduce framework.

4.3.2 Constructor & Destructor Documentation

4.3.2.1 DatastreamCoreset::DatastreamCoreset (FastCoreset * *fastCoreset*, int *firstBucketSize*) [inline]

Parameters

fastCoreset	FastCoreset instance to be wrapped

firstBucketSize	Merge & Reduce bucket size

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

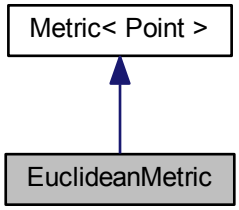
- DatastreamCoreset.hpp
- DatastreamCoreset.cpp

4.4 EuclideanMetric Class Reference

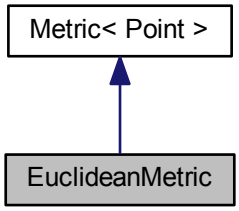
Euclidean metric for Point objetcs.

`#include <EuclideanMetric.hpp>`

Inheritance diagram for EuclideanMetric:



Collaboration diagram for EuclideanMetric:



Public Member Functions

- virtual double **distance** (Point const &x, Point const &y) const

4.4.1 Detailed Description

Euclidean metric for Point objetcs.

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

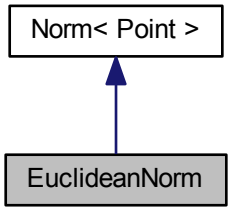
- EuclideanMetric.hpp
- EuclideanMetric.cpp

4.5 EuclideanNorm Class Reference

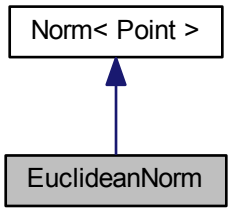
Euclidean norm for Point objetcs.

`#include <EuclideanNorm.hpp>`

Inheritance diagram for EuclideanNorm:



Collaboration diagram for EuclideanNorm:



Public Member Functions

- virtual double **length** (Point const &x) const

4.5.1 Detailed Description

Euclidean norm for Point objetcs.

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

- EuclideanNorm.hpp
- EuclideanNorm.cpp

4.6 EuclideanSpace Class Reference

Euclidean space operations for Point objetcs.

`#include <EuclideanSpace.hpp>`

Public Member Functions

- **EuclideanSpace** (std::function< Norm< Point > *()> createNorm)
- template<typename InputIterator >
  std::unique_ptr< std::vector
  < Point > > **orthonormalize** (InputIterator begin, InputIterator end)

4.6.1 Detailed Description

Euclidean space operations for Point objetcs.

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

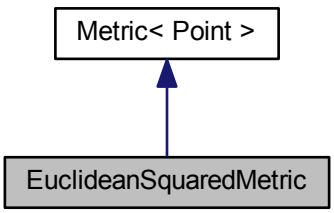
- EuclideanSpace.hpp
- EuclideanSpace.cpp

4.7 EuclideanSquaredMetric Class Reference

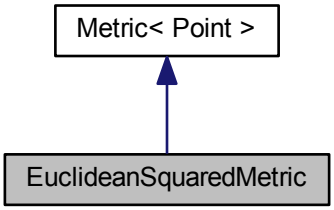
Euclidean squared metric for Point objetcs.

`#include <EuclideanSquaredMetric.hpp>`

Inheritance diagram for EuclideanSquaredMetric:



Collaboration diagram for EuclideanSquaredMetric:



Public Member Functions

- virtual double **distance** (Point const &x, Point const &y) const

4.7.1 Detailed Description

Euclidean squared metric for Point objetcs.

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

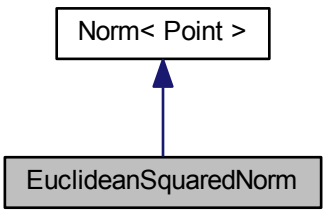
- EuclideanSquaredMetric.hpp
- EuclideanSquaredMetric.cpp

4.8 EuclideanSquaredNorm Class Reference

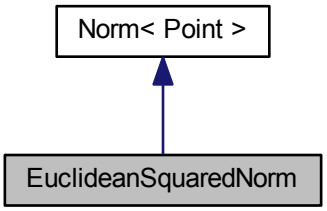
Euclidean squared norm for Point objetcs.

`#include <EuclideanSquaredNorm.hpp>`

Inheritance diagram for EuclideanSquaredNorm:



Collaboration diagram for EuclideanSquaredNorm:



Public Member Functions

- virtual double **length** (Point const &x) const

4.8.1 Detailed Description

Euclidean squared norm for Point objetcs.

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

- EuclideanSquaredNorm.hpp
- EuclideanSquaredNorm.cpp

4.9 FastCoreset Class Reference

Fast implementation of PROBI.

`#include <FastCoreset.hpp>`

Public Member Functions

- **FastCoreset** (std::function< Metric< Point > *() > createMetric, std::function< Norm< Point > *() > createNorm)
- void setK (int k)

  Sets number of centers.
- int getK () const

  Gets number of centers.
- void setWeiszfeldMedianIterations (int weiszfeldMedianIterations)

  Sets number of Weiszfeld iterations (approximation of 1-median)
- int getWeiszfeldMedianIterations () const

  Gets number of Weiszfeld iterations (approximation of 1-median)
- void setMaxLloydClusteringIterations (int maxLloydClusteringIterations)

  Sets number of "probabilistic Lloyd" iterations.
- int getMaxLloydClusteringIterations () const

  Gets number of "probabilistic Lloyd" iterations.
- void setKumarMedianIterations (int kumarMedianIterations)

Sets number of iterations in Kumar's k-median algorithm (fallback)

- int getKumarMedianIterations () const

  Gets number of iterations in Kumar's k-median algorithm (fallback)

- void setAllSamplesSize (int allSamplesSize)

  Sets the ring sample size.

- int getAllSamplesSize () const

  Gets the ring sample size.

- template<typename Iterator1 , typename Iterator2 >
  void computeCoreset (Iterator1 inputBegin, Iterator1 inputEnd, Iterator2 output, size_t n=0)
- template<typename RandomAccessIterator1 , typename Iterator2 >
  void **computeCoreset** (RandomAccessIterator1 inputBegin, RandomAccessIterator1 inputEnd, Iterator2 output, size_t n)

4.9.1 Detailed Description

Fast implementation of PROBI.

PROBI is a clustering algorithm for the probabilistic Euclidean k-median problem.

4.9.2 Member Function Documentation

4.9.2.1 template<typename Iterator1 , typename Iterator2 > void FastCoreset::computeCoreset (Iterator1 *inputBegin,* Iterator1 *inputEnd,* Iterator2 *output,* size_t *n* = 0)

Computes a k-median coreset

Parameters

begin	Input point set: begin
end	Input point set: end
output	Output iterator
n	Size of input (optional)

Returns

k-median coreset

4.9.2.2 int FastCoreset::getAllSamplesSize ( ) const

Gets the ring sample size.

Returns

Ring sample size

4.9.2.3 int FastCoreset::getK ( ) const

Gets number of centers.

Returns

Number of centers

4.9.2.4 int FastCoreset::getKumarMedianIterations ( ) const

Gets number of iterations in Kumar's k-median algorithm (fallback)

Returns

Maximum number of iterations

4.9.2.5 int FastCoreset::getMaxLloydClusteringIterations ( ) const

Gets number of "probabilistic Lloyd" iterations.

Returns

Maximum number of iterations

4.9.2.6 int FastCoreset::getWeiszfeldMedianIterations ( ) const

Gets number of Weiszfeld iterations (approximation of 1-median)

Returns

Maximum number of iterations

4.9.2.7 void FastCoreset::setAllSamplesSize (int *allSamplesSize*)

Sets the ring sample size.

Parameters

allSamplesSize	Ring sample size

4.9.2.8 void FastCoreset::setK (int *k*)

Sets number of centers.

Parameters

k	Number of centers

4.9.2.9 void FastCoreset::setKumarMedianIterations (int *kumarMedianIterations*)

Sets number of iterations in Kumar's k-median algorithm (fallback)

Parameters

kumarMedian-Iterations	Maximum number of iterations

4.9.2.10 void FastCoreset::setMaxLloydClusteringIterations (int *maxLloydClusteringIterations*)

Sets number of "probabilistic Lloyd" iterations.

Parameters

maxLloyd-Clustering-Iterations	Maximum number of iterations

4.9.2.11 void FastCoreset::setWeiszfeldMedianIterations (int *weiszfeldMedianIterations*)

Sets number of Weiszfeld iterations (approximation of 1-median)

Parameters

weiszfeld-MedianIterations	Maximum number of iterations

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

- FastCoreset.hpp
- FastCoreset.cpp

4.10 Weiszfeld::IterationFailed Class Reference

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

- Weiszfeld.hpp

4.11 KMedian Class Reference

k-median evaluation for Point objects

`#include <KMedian.hpp>`

Public Member Functions

- **KMedian** (std::function< Metric< Point > *()> createMetric)
- template<typename ForwardIterator >
  double cost (ForwardIterator first, ForwardIterator last, Point Center)

  1-median cost
- template<typename ForwardIteratorPoint , typename ForwardIteratorCenter >
  double cost (ForwardIteratorPoint beginP, ForwardIteratorPoint endP, ForwardIteratorCenter beginC, ForwardIteratorCenter endC)

  k-median cost
- template<typename ForwardIterator >
  double weightedCost (ForwardIterator first, ForwardIterator last, Point Center)

  1-median weighted cost
- template<typename ForwardIteratorPoint , typename ForwardIteratorCenter >
  double weightedCost (ForwardIteratorPoint beginP, ForwardIteratorPoint endP, ForwardIteratorCenter beginC, ForwardIteratorCenter endC)

  k-median weighted cost

4.11.1 Detailed Description

k-median evaluation for Point objects

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

- KMedian.hpp
- KMedian.cpp

4.12 KumarMedian Class Reference

1-median approximation

`#include <KumarMedian.hpp>`

Public Member Functions

- **KumarMedian** (std::function< Metric< Point > *()> createMetric, std::function< Norm< Point > *()> createNorm)
- template<typename InputIterator >
  Point approximateOneMedianRounds (InputIterator begin, InputIterator end, double eps, int rounds=0)

  Approximate k-median (choose best out of n)
- template<typename InputIterator >
  Point approximateOneMedian (InputIterator begin, InputIterator end, double eps)

  Approximate k-median.

4.12.1 Detailed Description

1-median approximation

Kumar, Sabharwal, Sen: Linear-time approximation schemes for clustering problems in any dimensions

4.12.2 Member Function Documentation

4.12.2.1 template<typename InputIterator > Point KumarMedian::approximateOneMedian (InputIterator *begin*, InputIterator *end*, double *eps*)

Approximate k-median.

Parameters

begin	Point iterator
end	Point iterator
eps	Factor of approximation

4.12.2.2 template<typename InputIterator > Point KumarMedian::approximateOneMedianRounds (InputIterator *begin*, InputIterator *end*, double *eps*, int *rounds* = `0`)

Approximate k-median (choose best out of n)

Parameters

begin	Point iterator
end	Point iterator
eps	Factor of approximation
rounds	Number of rounds

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

- KumarMedian.hpp
- KumarMedian.cpp

4.13 LloydMedian Class Reference

Adaption of Lloyd's algorithm for k-median.

`#include <LloydMedian.hpp>`

Public Member Functions

- **LloydMedian** (std::function< Metric< Point > *() > createMetric, std::function< Norm< Point > *() > createNorm)
- template<typename ForwardIterator , typename OutputIterator >
  void computeCenterSet (ForwardIterator begin, ForwardIterator end, OutputIterator output, size_t k, size_t maxIterations, size_t n=0)

4.13.1 Detailed Description

Adaption of Lloyd's algorithm for k-median.

Uses k-means++-like sampling

4.13.2 Member Function Documentation

4.13.2.1 template<typename ForwardIterator , typename OutputIterator > void LloydMedian::computeCenterSet (ForwardIterator *begin,* ForwardIterator *end,* OutputIterator *output,* size_t *k,* size_t *maxIterations,* size_t *n* = 0)

Computes a center set

Parameters

begin	Input point set: begin
end	Input point set: end
output	Output iterator
k	Number of centers
maxIterations	Maximum number of iterations
n	Size of input set (optional)

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

- LloydMedian.hpp
- LloydMedian.cpp

4.14 LloydProbMedian Class Reference

Adaption of Lloyd's algorithm for probabilistic k-median.

`#include <LloydProbMedian.hpp>`

Public Member Functions

- **LloydProbMedian** (std::function< Metric< Point > *() > createMetric, std::function< Norm< Point > *() > createNorm)
- template<typename ForwardIterator , typename OutputIterator >
  void computeCenterSet (ForwardIterator begin, ForwardIterator end, OutputIterator output, size_t k, size_t maxIterations, size_t n=0)

4.14.1 Detailed Description

Adaption of Lloyd's algorithm for probabilistic k-median.

Uses k-means++-like sampling

4.14.2 Member Function Documentation

4.14.2.1 template<typename ForwardIterator , typename OutputIterator > void LloydProbMedian::computeCenterSet (ForwardIterator *begin,* ForwardIterator *end,* OutputIterator *output,* size_t *k,* size_t *maxIterations,* size_t *n* = 0)

Computes a center set

Parameters

begin	Input point set: begin
end	Input point set: end
output	Output iterator
k	Number of centers
maxIterations	Maximum number of iterations
n	Size of input set (optional)

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

- LloydProbMedian.hpp
- LloydProbMedian.cpp

4.15 MergeReduce< T > Class Template Reference

Merge & Reduce framework template.

`#include <MergeReduce.hpp>`

Public Member Functions

- MergeReduce (int firstBucketSize)
- MergeReduce & operator<< (T const &element)
- virtual std::unique_ptr
  < std::vector< T > > **assemble** ()

4.15.1 Detailed Description

template<typename T>class MergeReduce< T >

Merge & Reduce framework template.

Can be used to form an offline algorithm into a streaming algorithm

4.15.2 Constructor & Destructor Documentation

4.15.2.1 template<typename T> MergeReduce< T >::MergeReduce (int *firstBucketSize*) [inline]

Parameters

firstBucketSize	Size of initial Merge & Reduce bucket

4.15.3 Member Function Documentation

4.15.3.1 template<typename T> MergeReduce< T > & MergeReduce< T >::operator<< (T const & *element*)

Streaming operator for reading points

Parameters

element	Point

Returns

This object

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

- MergeReduce.hpp

4.16 Metric< T > Class Template Reference

Metric interface.

`#include <Metric.hpp>`

Public Member Functions

- virtual double **distance** (T const &x, T const &y) const =0
- virtual double **distance** (T const *x, T const *y) const

4.16.1 Detailed Description

template<typename T>class Metric< T >

Metric interface.

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

- Metric.hpp

4.17 Norm< T > Class Template Reference

Norm interface.

`#include <Norm.hpp>`

Public Member Functions

- virtual double **length** (T const &x) const =0
- virtual double **length** (T const *x) const

4.17.1 Detailed Description

template<typename T>class Norm< T >

Norm interface.

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

- Norm.hpp

4.18 PKMedian Class Reference

Probabilistic k-median evaluator.

`#include <PKMedian.hpp>`

Public Member Functions

- **PKMedian** (std::function< Metric< Point > *() > createMetric)
- template<typename ForwardIteratorCenter >
  double weightedCost (ProbabilisticPoint const &pp, ForwardIteratorCenter beginC, ForwardIteratorCenter endC)

  Probabilistic k-median.
- template<typename ForwardIteratorPoint , typename ForwardIteratorCenter >
  double weightedCost (ForwardIteratorPoint beginP, ForwardIteratorPoint endP, ForwardIteratorCenter beginC, ForwardIteratorCenter endC)

  Probabilistic k-median.

4.18.1 Detailed Description

Probabilistic k-median evaluator.

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

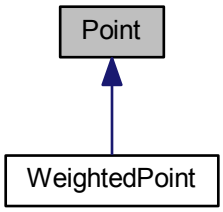
- PKMedian.hpp
- PKMedian.cpp

4.19 Point Class Reference

Point class.

`#include <Point.hpp>`

Inheritance diagram for Point:



Public Member Functions

- **Point** (Point const &point)=default
- **Point** (int dimension)
- **Point** (Point const *point)
- **Point** (std::vector< double > const &coordinates)
- double & **operator[]** (int index)
- double const & **operator[]** (int index) const
- Point & **operator+=** (const Point &point)
- Point & **operator-=** (const Point &point)
- Point const **operator+** (Point const &x) const
- Point const **operator-** (Point const &x) const
- bool **operator==** (Point const &x) const
- bool **operator!=** (Point const &x) const
- std::vector< double >
  ::const_iterator **cbegin** () const
- std::vector< double >
  ::const_iterator **cend** () const
- int **getDimension** () const

4.19.1 Detailed Description

Point class.

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

- Point.hpp
- Point.cpp

4.20 ProbabilisticPoint Class Reference

Probabilistic point.

`#include <ProbabilisticPoint.hpp>`

Public Types

- typedef std::vector < WeightedPoint > ::const_iterator **citerator**

Public Member Functions

- **ProbabilisticPoint** (std::vector< WeightedPoint > distribution)
- **ProbabilisticPoint** (std::vector< WeightedPoint > distribution, double weight)
- WeightedPoint & **operator[]** (int index)
- WeightedPoint const & **operator[]** (int index) const
- std::vector< WeightedPoint > ::const_iterator **cbegin** () const
- std::vector< WeightedPoint > ::const_iterator **cend** () const
- void **setWeight** (double weight)
- unsigned int **getSizeOfDistribution** () const
- double **getWeight** () const
- double **getRealizationProbability** () const

4.20.1 Detailed Description

Probabilistic point.

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

- ProbabilisticPoint.hpp
- ProbabilisticPoint.cpp

4.21 Randomness Class Reference

Encapsulates an STL random generator.

`#include <Randomness.hpp>`

Static Public Member Functions

- static std::mt19937 * **getMT19937** ()

4.21.1 Detailed Description

Encapsulates an STL random generator.

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

- Randomness.hpp
- Randomness.cpp

4.22 Sampling Class Reference

Sampling operations.

`#include <Sampling.hpp>`

Public Member Functions

- template<typename InputIterator , typename ResultType >
  std::unique_ptr< std::vector
  < ResultType > > **sampleWithoutReplacement** (InputIterator first, InputIterator last, size_t sizeOfSample)
- template<typename RandomAccessIterator , typename ResultType >
  std::unique_ptr< std::vector
  < ResultType > > **sampleWithoutReplacementFast** (RandomAccessIterator first, RandomAccessIterator last, size_t sizeOfSample)
- template<typename InputIterator , typename ResultType >
  std::unique_ptr< std::vector
  < ResultType > > **sampleWithReplacement** (InputIterator first, InputIterator last, std::vector< double > &weights, size_t sizeOfSample)
- template<typename InputIterator , typename ResultType >
  std::unique_ptr< std::vector
  < ResultType > > **sampleWithReplacement** (InputIterator first, InputIterator last, size_t sizeOfSet, size_t sizeOfSample)

Static Public Member Functions

- template<typename InputIterator , typename ResultType >
  static std::unique_ptr
  < std::vector< ResultType > > sampleWithoutReplacement (InputIterator first, InputIterator last, size_t sizeOfSample)

  Sample without replacement.
- template<typename RandomAccessIterator , typename ResultType >
  static std::unique_ptr
  < std::vector< ResultType > > sampleWithoutReplacementFast (RandomAccessIterator first, RandomAccessIterator last, size_t sizeOfSample)

  Sample without replacement.
- template<typename InputIterator , typename ResultType >
  static std::unique_ptr
  < std::vector< ResultType > > sampleWithReplacement (InputIterator first, InputIterator last, std::vector< double > &weights, size_t sizeOfSample)

  Sample with replacement.
- template<typename InputIterator , typename ResultType >
  static std::unique_ptr
  < std::vector< ResultType > > sampleWithReplacement (InputIterator first, InputIterator last, size_t sizeOfSet, size_t sizeOfSample)

  Sample with replacement.

4.22.1 Detailed Description

Sampling operations.

4.22.2 Member Function Documentation

4.22.2.1 template<typename InputIterator , typename ResultType > static std::unique_ptr<std::vector<ResultType> > Sampling::sampleWithoutReplacement (InputIterator *first,* InputIterator *last,* size_t *sizeOfSample*) [static]

Sample without replacement.

Requires InputIterator

4.22.2.2 template<typename RandomAccessIterator , typename ResultType > static std::unique_ptr<std::vector<ResultType> > Sampling::sampleWithoutReplacementFast (RandomAccessIterator *first,* RandomAccessIterator *last,* size_t *sizeOfSample*) [static]

Sample without replacement.

Requires RandomAccessIterator

4.22.2.3 template<typename InputIterator , typename ResultType > static std::unique_ptr<std::vector<ResultType> > Sampling::sampleWithReplacement (InputIterator *first,* InputIterator *last,* std::vector< double > & *weights,* size_t *sizeOfSample*) [static]

Sample with replacement.

Output order is not random!

4.22.2.4 template<typename InputIterator , typename ResultType > static std::unique_ptr<std::vector<ResultType> > Sampling::sampleWithReplacement (InputIterator *first,* InputIterator *last,* size_t *sizeOfSet,* size_t *sizeOfSample*) [static]

Sample with replacement.

Output order is not random!

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

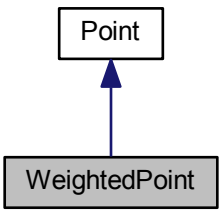
- Sampling.hpp

4.23 WeightedPoint Class Reference

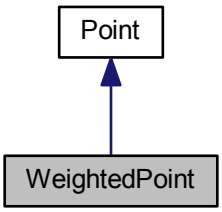
Weighted point class.

`#include <WeightedPoint.hpp>`

Inheritance diagram for WeightedPoint:



Collaboration diagram for WeightedPoint:



Public Member Functions

- **WeightedPoint** (WeightedPoint const &point)=default
- **WeightedPoint** (Point const &point)
- **WeightedPoint** (std::vector< double > const &coordinates)
- **WeightedPoint** (std::vector< double > const &coordinates, double weight)
- void **setWeight** (double weight)
- double **getWeight** () const

4.23.1 Detailed Description

Weighted point class.

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

- WeightedPoint.hpp
- WeightedPoint.cpp

4.24 Weiszfeld Class Reference

1-median approximation

`#include <Weiszfeld.hpp>`

Classes

- class IterationFailed

Public Member Functions

- **Weiszfeld** (std::function< Metric< Point > *() > createMetric)
- template<typename ForwardIterator >
  Point approximateOneMedian (ForwardIterator begin, ForwardIterator end, int max_iteration=15)

  Approximate 1-median.

4.24.1 Detailed Description

1-median approximation

Weiszfeld

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

- Weiszfeld.hpp
- Weiszfeld.cpp

Index

AdaptiveSampling, 7
    computeCenterSet, 7
approximateOneMedian
    KumarMedian, 18
approximateOneMedianRounds
    KumarMedian, 18

CenterOfGravity, 8
    cog, 8
cog
    CenterOfGravity, 8
computeCenterSet
    AdaptiveSampling, 7
    LloydMedian, 19
    LloydProbMedian, 20
computeCoreset
    FastCoreset, 15

DatastreamCoreset, 8
    DatastreamCoreset, 9
    DatastreamCoreset, 9

EuclideanMetric, 10
EuclideanNorm, 11
EuclideanSpace, 12
EuclideanSquaredMetric, 12
EuclideanSquaredNorm, 13

FastCoreset, 14
    computeCoreset, 15
    getAllSamplesSize, 15
    getK, 15
    getKumarMedianIterations, 15
    getMaxLloydClusteringIterations, 16
    getWeiszfeldMedianIterations, 16
    setAllSamplesSize, 16
    setK, 16
    setKumarMedianIterations, 16
    setMaxLloydClusteringIterations, 16
    setWeiszfeldMedianIterations, 17

getAllSamplesSize
    FastCoreset, 15
getK
    FastCoreset, 15
getKumarMedianIterations
    FastCoreset, 15
getMaxLloydClusteringIterations
    FastCoreset, 16
getWeiszfeldMedianIterations
    FastCoreset, 16

KMedian, 17
KumarMedian, 18
    approximateOneMedian, 18
    approximateOneMedianRounds, 18

LloydMedian, 19
    computeCenterSet, 19
LloydProbMedian, 20
    computeCenterSet, 20

MergeReduce
    MergeReduce, 21
    MergeReduce, 21
    operator<<, 21
MergeReduce< T >, 20
Metric< T >, 21

Norm< T >, 22

operator<<
    MergeReduce, 21

PKMedian, 22
Point, 22
ProbabilisticPoint, 23

Randomness, 24

sampleWithReplacement
    Sampling, 26
sampleWithoutReplacement
    Sampling, 25
sampleWithoutReplacementFast
    Sampling, 25
Sampling, 24
    sampleWithReplacement, 26
    sampleWithoutReplacement, 25
    sampleWithoutReplacementFast, 25
setAllSamplesSize
    FastCoreset, 16
setK
    FastCoreset, 16
setKumarMedianIterations
    FastCoreset, 16
setMaxLloydClusteringIterations
    FastCoreset, 16
setWeiszfeldMedianIterations
    FastCoreset, 17

WeightedPoint, 26
Weiszfeld, 27
Weiszfeld::IterationFailed, 17