

LSHBOX

0.1

Generated by Doxygen 1.8.7

Tue Jul 1 2014 21:07:16

Contents

1	Namespace Index	1
1.1	Namespace List	1
2	Class Index	3
2.1	Class List	3
3	File Index	5
3.1	File List	5
4	Namespace Documentation	7
4.1	Ishbox Namespace Reference	7
4.1.1	Function Documentation	7
4.1.1.1	ascend	7
4.1.1.2	sqr	8
5	Class Documentation	9
5.1	Matrix< T >::Accessor Class Reference	9
5.1.1	Detailed Description	11
5.1.2	Member Typedef Documentation	11
5.1.2.1	DATATYPE	11
5.1.2.2	Key	11
5.1.2.3	Value	11
5.1.3	Constructor & Destructor Documentation	11
5.1.3.1	Accessor	11
5.1.4	Member Function Documentation	11
5.1.4.1	mark	11
5.1.4.2	operator()	11
5.1.4.3	reset	12
5.1.5	Member Data Documentation	12
5.1.5.1	flags_	12
5.1.5.2	matrix_	12
5.2	Benchmark Class Reference	12
5.2.1	Detailed Description	14

5.2.2	Constructor & Destructor Documentation	14
5.2.2.1	Benchmark	14
5.2.2.2	~Benchmark	14
5.2.3	Member Function Documentation	14
5.2.3.1	getAnswer	14
5.2.3.2	getAnswer	14
5.2.3.3	getK	14
5.2.3.4	getQ	14
5.2.3.5	getQuery	14
5.2.3.6	init	15
5.2.3.7	load	15
5.2.3.8	load	16
5.2.3.9	resize	16
5.2.3.10	save	16
5.2.3.11	save	17
5.2.4	Member Data Documentation	17
5.2.4.1	K_	17
5.2.4.2	Q_	17
5.2.4.3	queries_	17
5.2.4.4	topks_	17
5.3	itqLsh< DATATYPE > Class Template Reference	18
5.3.1	Detailed Description	19
5.3.2	Constructor & Destructor Documentation	19
5.3.2.1	itqLsh	19
5.3.2.2	itqLsh	19
5.3.2.3	~itqLsh	20
5.3.3	Member Function Documentation	20
5.3.3.1	insert	20
5.3.3.2	load	20
5.3.3.3	query	20
5.3.3.4	reset	20
5.3.3.5	save	21
5.3.3.6	train	21
5.3.4	Member Data Documentation	21
5.3.4.1	omegasAll	22
5.3.4.2	param	22
5.3.4.3	pcsAll	22
5.3.4.4	rndArray	22
5.3.4.5	tables	22
5.4	Matrix< T > Class Template Reference	22

5.4.1	Detailed Description	24
5.4.2	Constructor & Destructor Documentation	24
5.4.2.1	Matrix	24
5.4.2.2	Matrix	24
5.4.2.3	~Matrix	24
5.4.2.4	Matrix	24
5.4.3	Member Function Documentation	25
5.4.3.1	free	25
5.4.3.2	getDim	25
5.4.3.3	getSize	25
5.4.3.4	load	26
5.4.3.5	load	26
5.4.3.6	load	26
5.4.3.7	operator[]	27
5.4.3.8	operator[]	27
5.4.3.9	reset	27
5.4.3.10	save	28
5.4.4	Member Data Documentation	28
5.4.4.1	dim	28
5.4.4.2	dims	28
5.4.4.3	N	28
5.5	Metric< DATATYPE > Class Template Reference	28
5.5.1	Detailed Description	29
5.5.2	Constructor & Destructor Documentation	29
5.5.2.1	Metric	29
5.5.2.2	~Metric	29
5.5.3	Member Function Documentation	29
5.5.3.1	dim	29
5.5.3.2	dist	29
5.5.4	Member Data Documentation	30
5.5.4.1	dim_	30
5.5.4.2	type_	30
5.6	shLsh< DATATYPE >::Parameter Struct Reference	30
5.6.1	Detailed Description	31
5.6.2	Member Data Documentation	31
5.6.2.1	D	31
5.6.2.2	L	31
5.6.2.3	M	31
5.6.2.4	N	31
5.6.2.5	S	31

5.7	thLsh< DATATYPE >::Parameter Struct Reference	32
5.7.1	Detailed Description	32
5.7.2	Member Data Documentation	32
5.7.2.1	D	32
5.7.2.2	L	33
5.7.2.3	M	33
5.7.2.4	Max	33
5.7.2.5	Min	33
5.7.2.6	N	33
5.8	rhplsh< DATATYPE >::Parameter Struct Reference	33
5.8.1	Detailed Description	34
5.8.2	Member Data Documentation	34
5.8.2.1	D	34
5.8.2.2	L	34
5.8.2.3	M	34
5.8.2.4	N	34
5.9	psdLsh< DATATYPE >::Parameter Struct Reference	34
5.9.1	Detailed Description	35
5.9.2	Member Data Documentation	35
5.9.2.1	D	35
5.9.2.2	L	35
5.9.2.3	M	36
5.9.2.4	T	36
5.9.2.5	W	36
5.10	rbslsh::Parameter Struct Reference	36
5.10.1	Detailed Description	37
5.10.2	Member Data Documentation	37
5.10.2.1	C	37
5.10.2.2	D	37
5.10.2.3	L	37
5.10.2.4	M	37
5.10.2.5	N	37
5.11	itqLsh< DATATYPE >::Parameter Struct Reference	37
5.11.1	Detailed Description	38
5.11.2	Member Data Documentation	38
5.11.2.1	D	38
5.11.2.2	I	38
5.11.2.3	L	39
5.11.2.4	M	39
5.11.2.5	N	39

5.11.2.6	S	39
5.12	progress_display Class Reference	39
5.12.1	Detailed Description	41
5.12.2	Constructor & Destructor Documentation	41
5.12.2.1	progress_display	41
5.12.3	Member Function Documentation	41
5.12.3.1	count	41
5.12.3.2	display_tic	41
5.12.3.3	expected_count	41
5.12.3.4	operator++	42
5.12.3.5	operator+=	42
5.12.3.6	restart	43
5.12.4	Member Data Documentation	43
5.12.4.1	_count	43
5.12.4.2	_expected_count	43
5.12.4.3	_next_tic_count	43
5.12.4.4	_tic	43
5.12.4.5	m_os	43
5.12.4.6	m_s1	43
5.12.4.7	m_s2	44
5.12.4.8	m_s3	44
5.13	psdLsh< DATATYPE > Class Template Reference	44
5.13.1	Detailed Description	45
5.13.2	Constructor & Destructor Documentation	45
5.13.2.1	psdLsh	45
5.13.2.2	psdLsh	45
5.13.2.3	~psdLsh	46
5.13.3	Member Function Documentation	46
5.13.3.1	insert	46
5.13.3.2	load	46
5.13.3.3	query	46
5.13.3.4	reset	46
5.13.3.5	save	47
5.13.4	Member Data Documentation	47
5.13.4.1	param	47
5.13.4.2	rndBs	47
5.13.4.3	stableArray	47
5.13.4.4	tables	47
5.14	rbsLsh Class Reference	47
5.14.1	Detailed Description	49

5.14.2	Constructor & Destructor Documentation	49
5.14.2.1	rbsLsh	49
5.14.2.2	rbsLsh	49
5.14.2.3	~rbsLsh	49
5.14.3	Member Function Documentation	49
5.14.3.1	insert	49
5.14.3.2	load	50
5.14.3.3	query	50
5.14.3.4	reset	50
5.14.3.5	save	50
5.14.4	Member Data Documentation	51
5.14.4.1	param	51
5.14.4.2	rndArray	51
5.14.4.3	rndBits	51
5.14.4.4	tables	51
5.15	rhplsh< DATATYPE > Class Template Reference	51
5.15.1	Detailed Description	53
5.15.2	Constructor & Destructor Documentation	53
5.15.2.1	rhplsh	53
5.15.2.2	rhplsh	53
5.15.2.3	~rhplsh	53
5.15.3	Member Function Documentation	53
5.15.3.1	insert	53
5.15.3.2	load	54
5.15.3.3	query	54
5.15.3.4	reset	54
5.15.3.5	save	54
5.15.4	Member Data Documentation	55
5.15.4.1	param	55
5.15.4.2	rndArray	55
5.15.4.3	tables	55
5.15.4.4	uosArray	55
5.16	Scanner< ACCESSOR > Class Template Reference	55
5.16.1	Detailed Description	57
5.16.2	Member Typedef Documentation	57
5.16.2.1	DATATYPE	57
5.16.2.2	Value	57
5.16.3	Constructor & Destructor Documentation	57
5.16.3.1	Scanner	57
5.16.4	Member Function Documentation	57

5.16.4.1	cnt	57
5.16.4.2	operator()	58
5.16.4.3	reset	58
5.16.4.4	topk	58
5.16.4.5	topk	58
5.16.5	Member Data Documentation	58
5.16.5.1	accessor_	58
5.16.5.2	cnt_	59
5.16.5.3	K_	59
5.16.5.4	metric_	59
5.16.5.5	query_	59
5.16.5.6	R_	59
5.16.5.7	topk_	59
5.17	shLsh< DATATYPE > Class Template Reference	59
5.17.1	Detailed Description	61
5.17.2	Constructor & Destructor Documentation	61
5.17.2.1	shLsh	61
5.17.2.2	shLsh	61
5.17.2.3	~shLsh	61
5.17.3	Member Function Documentation	62
5.17.3.1	insert	62
5.17.3.2	load	63
5.17.3.3	query	63
5.17.3.4	reset	63
5.17.3.5	save	64
5.17.3.6	train	64
5.17.4	Member Data Documentation	64
5.17.4.1	minsAll	65
5.17.4.2	omegasAll	65
5.17.4.3	param	65
5.17.4.4	pcsAll	65
5.17.4.5	rndArray	65
5.17.4.6	tables	65
5.18	Stat Class Reference	65
5.18.1	Detailed Description	67
5.18.2	Constructor & Destructor Documentation	67
5.18.2.1	Stat	67
5.18.2.2	~Stat	67
5.18.3	Member Function Documentation	67
5.18.3.1	append	67

5.18.3.2	getAvg	67
5.18.3.3	getCount	67
5.18.3.4	getMax	68
5.18.3.5	getMin	68
5.18.3.6	getStd	68
5.18.3.7	getSum	68
5.18.3.8	merge	68
5.18.3.9	operator<<	68
5.18.3.10	reset	68
5.18.4	Member Data Documentation	68
5.18.4.1	count	68
5.18.4.2	max	68
5.18.4.3	min	69
5.18.4.4	sum	69
5.18.4.5	sum2	69
5.19	thLsh< DATATYPE > Class Template Reference	69
5.19.1	Detailed Description	71
5.19.2	Constructor & Destructor Documentation	71
5.19.2.1	thLsh	71
5.19.2.2	thLsh	71
5.19.2.3	~thLsh	71
5.19.3	Member Function Documentation	72
5.19.3.1	insert	72
5.19.3.2	load	72
5.19.3.3	query	72
5.19.3.4	reset	72
5.19.3.5	save	72
5.19.4	Member Data Documentation	73
5.19.4.1	param	73
5.19.4.2	rndArray	73
5.19.4.3	rndBits	73
5.19.4.4	tables	73
5.19.4.5	thresholds	73
5.20	timer Class Reference	73
5.20.1	Detailed Description	74
5.20.2	Constructor & Destructor Documentation	74
5.20.2.1	timer	74
5.20.2.2	~timer	74
5.20.3	Member Function Documentation	74
5.20.3.1	elapsed	74

5.20.3.2	restart	74
5.20.4	Member Data Documentation	74
5.20.4.1	time	74
5.21	Topk Class Reference	75
5.21.1	Detailed Description	75
5.21.2	Constructor & Destructor Documentation	76
5.21.2.1	Topk	76
5.21.2.2	~Topk	76
5.21.3	Member Function Documentation	76
5.21.3.1	getMin	76
5.21.3.2	getTopk	76
5.21.3.3	getTopk	76
5.21.3.4	push	76
5.21.3.5	recall	77
5.21.3.6	reset	77
5.21.4	Member Data Documentation	78
5.21.4.1	heapv	78
5.21.4.2	K	78
5.21.4.3	R	78
6	File Documentation	79
6.1	include/lshbox.h File Reference	79
6.1.1	Detailed Description	79
6.2	include/lshbox/basis.h File Reference	80
6.2.1	Detailed Description	81
6.2.2	Macro Definition Documentation	81
6.2.2.1	M_PI	81
6.3	include/lshbox/eval.h File Reference	81
6.3.1	Detailed Description	82
6.4	include/lshbox/itqlsh.h File Reference	83
6.4.1	Detailed Description	84
6.5	include/lshbox/matrix.h File Reference	84
6.5.1	Detailed Description	85
6.6	include/lshbox/metric.h File Reference	86
6.6.1	Detailed Description	86
6.6.2	Macro Definition Documentation	87
6.6.2.1	L1_DIST	87
6.6.2.2	L2_DIST	87
6.7	include/lshbox/psdlsh.h File Reference	87
6.7.1	Detailed Description	88

6.7.2	Macro Definition Documentation	89
6.7.2.1	CAUCHY	89
6.7.2.2	GAUSSIAN	89
6.8	include/lshbox/rbsh.h File Reference	89
6.8.1	Detailed Description	90
6.9	include/lshbox/rhplsh.h File Reference	90
6.9.1	Detailed Description	91
6.10	include/lshbox/shlsh.h File Reference	92
6.10.1	Detailed Description	93
6.11	include/lshbox/thlsh.h File Reference	93
6.11.1	Detailed Description	94
6.12	include/lshbox/topk.h File Reference	95
6.12.1	Detailed Description	95
Index		97

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

Ishbox	7
---------------	-------	---

Chapter 2

Class Index

2.1 Class List

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

Matrix< T >::Accessor	9
Benchmark	12
itqLsh< DATATYPE >	18
Matrix< T >	22
Metric< DATATYPE >	28
shLsh< DATATYPE >::Parameter	30
thLsh< DATATYPE >::Parameter	32
rhplsh< DATATYPE >::Parameter	33
psdLsh< DATATYPE >::Parameter	34
rbsLsh::Parameter	36
itqLsh< DATATYPE >::Parameter	37
progress_display	39
psdLsh< DATATYPE >	44
rbsLsh	47
rhplsh< DATATYPE >	51
Scanner< ACCESSOR >	55
shLsh< DATATYPE >	59
Stat	65
thLsh< DATATYPE >	69
timer	73
Topk	75

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

include/ lshbox.h	
The LSHBOX master header file	79
include/lshbox/ basis.h	
A set of basic tools	80
include/lshbox/ eval.h	
A set of classes for evaluation	81
include/lshbox/ itqlsh.h	
Locality-Sensitive Hashing Scheme Based on Iterative Quantization	83
include/lshbox/ matrix.h	
Dataset management class	84
include/lshbox/ metric.h	
Common distance measures	86
include/lshbox/ psdlsh.h	
Locality-Sensitive Hashing Scheme Based on p-Stable Distributions	87
include/lshbox/ rbslsh.h	
Locality-Sensitive Hashing Scheme Based on Random Bits Sampling	89
include/lshbox/ rhplsh.h	
Locality-Sensitive Hashing Scheme Based on Random Hyperplane	90
include/lshbox/ shlsh.h	
Locality-Sensitive Hashing Scheme Based on Spectral Hashing	92
include/lshbox/ thlsh.h	
Locality-Sensitive Hashing Scheme Based on Thresholding	93
include/lshbox/ topk.h	
Top-K data structures	95

Chapter 4

Namespace Documentation

4.1 Ishbox Namespace Reference

Classes

- class **Benchmark**
- class **itqLsh**
- class **Matrix**
- class **Metric**
- class **progress_display**
- class **psdLsh**
- class **rbsLsh**
- class **rhpLsh**
- class **Scanner**
- class **shLsh**
- class **Stat**
- class **thLsh**
- class **timer**
- class **Topk**

Functions

- bool **ascend** (const std::pair< unsigned, float > &lhs, const std::pair< unsigned, float > &rhs)
- template<typename DATATYPE >
DATATYPE **sqr** (const DATATYPE &x)

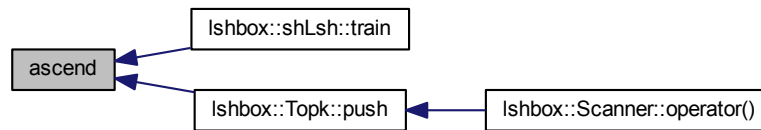
4.1.1 Function Documentation

4.1.1.1 bool Ishbox::ascend (const std::pair< unsigned, float > &lhs, const std::pair< unsigned, float > &rhs)

Sort std::vector<std::pair<unsigned, float> > by the second value.

Definition at line 39 of file basis.h.

Here is the caller graph for this function:

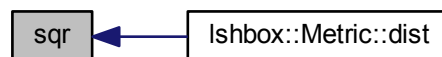


4.1.1.2 DATATYPE Ishbox::sqr (const DATATYPE & x)

The calculation of square.

Definition at line 38 of file metric.h.

Here is the caller graph for this function:



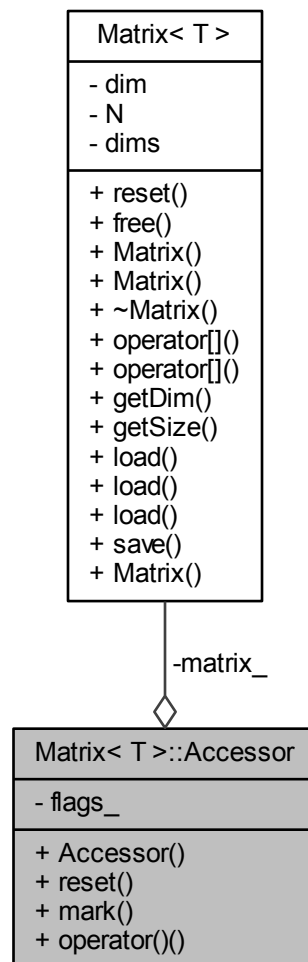
Chapter 5

Class Documentation

5.1 Matrix< T >::Accessor Class Reference

```
#include <matrix.h>
```

Collaboration diagram for Matrix< T >::Accessor:



Public Types

- typedef unsigned **Key**
- typedef const T * **Value**
- typedef T **DATATYPE**

Public Member Functions

- **Accessor** (const **Matrix** &matrix)
- void **reset** ()
- bool **mark** (unsigned key)
- const T * **operator**() (unsigned key)

Private Attributes

- const **Matrix** & **matrix_**

- `std::vector< bool > flags_`

5.1.1 Detailed Description

`template<class T>class lshbox::Matrix< T >::Accessor`

An accessor class to be used with LSH index.

Definition at line 179 of file `matrix.h`.

5.1.2 Member Typedef Documentation

5.1.2.1 `typedef T DATATYPE`

Definition at line 186 of file `matrix.h`.

5.1.2.2 `typedef unsigned Key`

Definition at line 184 of file `matrix.h`.

5.1.2.3 `typedef const T* Value`

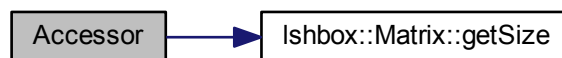
Definition at line 185 of file `matrix.h`.

5.1.3 Constructor & Destructor Documentation

5.1.3.1 `Accessor (const Matrix & matrix) [inline]`

Definition at line 187 of file `matrix.h`.

Here is the call graph for this function:



5.1.4 Member Function Documentation

5.1.4.1 `bool mark (unsigned key) [inline]`

Definition at line 196 of file `matrix.h`.

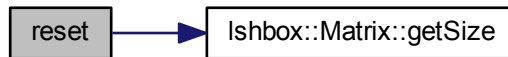
5.1.4.2 `const T* operator() (unsigned key) [inline]`

Definition at line 205 of file `matrix.h`.

5.1.4.3 void reset () [inline]

Definition at line 191 of file matrix.h.

Here is the call graph for this function:



5.1.5 Member Data Documentation

5.1.5.1 std::vector<bool> flags_ [private]

Definition at line 182 of file matrix.h.

5.1.5.2 const Matrix& matrix_ [private]

Definition at line 181 of file matrix.h.

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

- include/lshbox/**matrix.h**

5.2 Benchmark Class Reference

```
#include <eval.h>
```


Collaboration diagram for Benchmark:

Benchmark
- Q_ - K_ - queries_ - topks_
+ Benchmark() + resize() + init() + ~Benchmark() + save() + save() + load() + load() + getQ() + getK() + getQuery() + getAnswer() + getAnswer()

Public Member Functions

- **Benchmark** ()
- void **resize** (unsigned Q, unsigned K=0)
- void **init** (unsigned Q, unsigned K, unsigned maxID, unsigned seed=0)
- **~Benchmark** ()
- void **save** (std::ostream &os) const
- void **save** (const std::string &path) const
- void **load** (std::istream &is)
- void **load** (const std::string &path)
- unsigned **getQ** () const
- unsigned **getK** () const
- unsigned **getQuery** (unsigned n) const
- const **Topk** & **getAnswer** (unsigned n) const
- **Topk** & **getAnswer** (unsigned n)

Private Attributes

- unsigned **Q_**
- unsigned **K_**
- std::vector< unsigned > **queries_**
- std::vector< **Topk** > **topks_**

5.2.1 Detailed Description

Use for access a benchmark file.

We assume the feature vectors in the benchmark database are numbered from 0 to N. We sample Q queries as test examples and run K-NN search against the database with linear scan. The results are saved in a benchmark file for evaluation purpose. A benchmark file is made up of Q lines, each line represents a test query and is of the following format:

```
[query ID] [K] [1st NN's ID] [distance] [2nd NN's ID] [distance] ... [Kth NN's ID] [distance]
```

For all queries in the benchmark file, the K value should be the same.

Because the query points are also sampled from the database, they should be excluded from scanning when running this particular query.

Definition at line 56 of file eval.h.

5.2.2 Constructor & Destructor Documentation

5.2.2.1 `Benchmark()` [inline]

Definition at line 59 of file eval.h.

5.2.2.2 `~Benchmark()` [inline]

Definition at line 114 of file eval.h.

5.2.3 Member Function Documentation

5.2.3.1 `const Topk& getAnswer(unsigned n) const` [inline]

Get the nearest neighbors of the nth query.

Definition at line 195 of file eval.h.

5.2.3.2 `Topk& getAnswer(unsigned n)` [inline]

Get the KNNs for modification.

Definition at line 202 of file eval.h.

5.2.3.3 `unsigned getK() const` [inline]

Get the result number for each query.

Definition at line 181 of file eval.h.

5.2.3.4 `unsigned getQ() const` [inline]

Get the query number for benchmark.

Definition at line 174 of file eval.h.

5.2.3.5 `unsigned getQuery(unsigned n) const` [inline]

Get the ID of the nth query.

Definition at line 188 of file eval.h.

5.2.3.6 `void init (unsigned Q, unsigned K, unsigned maxID, unsigned seed = 0) [inline]`

Random initialization.

Parameters

<i>Q</i>	The query number for brencmark
<i>K</i>	The result number for each query
<i>maxID</i>	The number of vectors in the search library
<i>seed</i>	Seed some value for random to generate different query samples

Definition at line 85 of file eval.h.

Here is the call graph for this function:



5.2.3.7 `void load (std::istream & is) [inline]`

Load the benchmark from byte stream.

Definition at line 144 of file eval.h.

Here is the call graph for this function:



Here is the caller graph for this function:

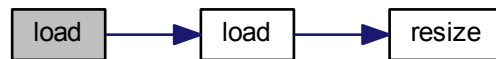


5.2.3.8 void load (const std::string & path) [inline]

Load the benchmark from a text file.

Definition at line 165 of file eval.h.

Here is the call graph for this function:



5.2.3.9 void resize (unsigned Q, unsigned K = 0) [inline]

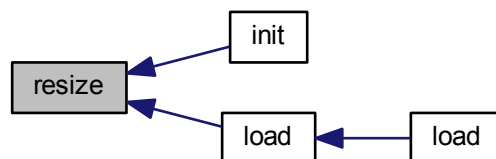
Change the query number for benchmark and the result number for each query.

Parameters

Q	The new query number for benchmark
K	The new result number for each query

Definition at line 66 of file eval.h.

Here is the caller graph for this function:



5.2.3.10 void save (std::ostream & os) const [inline]

Save the benchmark to byte stream.

Definition at line 118 of file eval.h.

Here is the caller graph for this function:



5.2.3.11 `void save (const std::string & path) const` `[inline]`

Save the benchmark as a text file.

Definition at line 135 of file eval.h.

Here is the call graph for this function:



5.2.4 Member Data Documentation

5.2.4.1 `unsigned K_` `[private]`

Definition at line 208 of file eval.h.

5.2.4.2 `unsigned Q_` `[private]`

Definition at line 207 of file eval.h.

5.2.4.3 `std::vector<unsigned> queries_` `[private]`

Definition at line 209 of file eval.h.

5.2.4.4 `std::vector<Topk> topks_` `[private]`

Definition at line 210 of file eval.h.

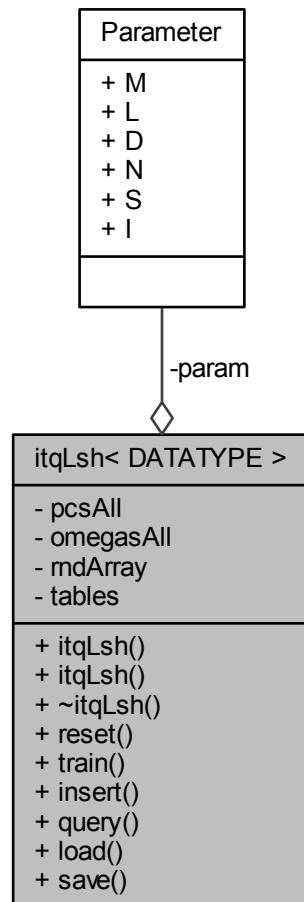
The documentation for this class was generated from the following file:

- include/lshbox/**eval.h**

5.3 itqLsh< DATATYPE > Class Template Reference

```
#include <itqlsh.h>
```

Collaboration diagram for itqLsh< DATATYPE >:



Classes

- struct **Parameter**

Public Member Functions

- **itqLsh** ()
- **itqLsh** (const **Parameter** ¶m_)
- **~itqLsh** ()
- void **reset** (const **Parameter** ¶m_)
- void **train** (**Matrix**< DATATYPE > &data)
- void **insert** (unsigned key, DATATYPE *domin)
- template<typename SCANNER >
void **query** (DATATYPE *domin, SCANNER &scanner)

- void **load** (const std::string &file)
- void **save** (const std::string &file)

Private Attributes

- **Parameter param**
- std::vector< std::vector
< std::vector< float > > > **pcsAll**
- std::vector< std::vector
< std::vector< float > > > **omegasAll**
- std::vector< std::vector
< unsigned > > **rndArray**
- std::vector< std::map
< unsigned, std::vector
< unsigned > > > **tables**

5.3.1 Detailed Description

template<typename DATATYPE = float>class Ishbox::itqLsh< DATATYPE >

Locality-Sensitive Hashing Scheme Based on Iterative Quantization.

For more information on iterative quantization based LSH, see the following reference.

Gong Y, Lazebnik S, Gordo A, et al. Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval[J]. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 2013, 35(12): 2916-2929.

Definition at line 50 of file itqlsh.h.

5.3.2 Constructor & Destructor Documentation

5.3.2.1 itqLsh () [inline]

Definition at line 68 of file itqlsh.h.

5.3.2.2 itqLsh (const Parameter & param_) [inline]

Definition at line 69 of file itqlsh.h.

Here is the call graph for this function:



5.3.2.3 `~itqLsh()` `[inline]`

Definition at line 73 of file `itqlsh.h`.

5.3.3 Member Function Documentation

5.3.3.1 `void insert(unsigned key, DATATYPE * domin)` `[inline]`

Insert a vector to the index.

Parameters

<i>key</i>	The sequence number of vector
<i>domin</i>	The pointer to the vector

Definition at line 195 of file `itqlsh.h`.

Here is the caller graph for this function:



5.3.3.2 `void load(const std::string & file)` `[inline]`

Load the index from binary file.

Parameters

<i>file</i>	The path of binary file.
-------------	--------------------------

Definition at line 271 of file `itqlsh.h`.

5.3.3.3 `void query(DATATYPE * domin, SCANNER & scanner)` `[inline]`

Query the approximate nearest neighborhoods.

Parameters

<i>domin</i>	The pointer to the vector
<i>scanner</i>	Top-K scanner, use for scan the approximate nearest neighborhoods

Definition at line 231 of file `itqlsh.h`.

5.3.3.4 `void reset(const Parameter & param_)` `[inline]`

Reset the parameter setting

Parameters

<i>param_</i>	A instance of itqLsh<DATATYPE>::Parametor, which contains the necessary parameters
---------------	--

Definition at line 80 of file itqlsh.h.

Here is the caller graph for this function:



5.3.3.5 void save (const std::string & file) [inline]

Save the index as binary file.

Parameters

<i>file</i>	The path of binary file.
-------------	--------------------------

Definition at line 315 of file itqlsh.h.

5.3.3.6 void train (Matrix< DATATYPE > & data) [inline]

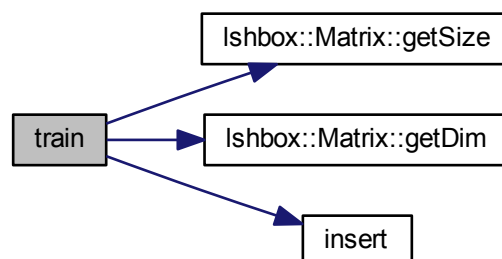
Train the data to get several groups of suitable vector for index.

Parameters

<i>data</i>	A instance of Matrix<DATATYPE>, most of the time, is the search library.
-------------	--

Definition at line 102 of file itqlsh.h.

Here is the call graph for this function:



5.3.4 Member Data Documentation

5.3.4.1 `std::vector<std::vector<std::vector<float>>> omegasAll` [private]

Definition at line 347 of file itqlsh.h.

5.3.4.2 `Parameter param` [private]

Definition at line 345 of file itqlsh.h.

5.3.4.3 `std::vector<std::vector<std::vector<float>>> pcsAll` [private]

Definition at line 346 of file itqlsh.h.

5.3.4.4 `std::vector<std::vector<unsigned>> rndArray` [private]

Definition at line 348 of file itqlsh.h.

5.3.4.5 `std::vector<std::map<unsigned, std::vector<unsigned>>> tables` [private]

Definition at line 349 of file itqlsh.h.

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

- `include/lshbox/itqlsh.h`

5.4 `Matrix< T >` Class Template Reference

```
#include <matrix.h>
```

Collaboration diagram for Matrix< T >:

Matrix< T >
- dim - N - dims
+ reset() + free() + Matrix() + Matrix() + ~Matrix() + operator[]() + operator[]() + getDim() + getSize() + load() + load() + load() + save() + Matrix()

Classes

- class **Accessor**

Public Member Functions

- void **reset** (int _dim, int _N)
- void **free** (void)
- **Matrix** ()
- **Matrix** (int _dim, int _N)
- **~Matrix** ()
- const T * **operator[]** (int i) const
- T * **operator[]** (int i)
- int **getDim** () const
- int **getSize** () const
- void **load** (const std::string &path)
- void **load** (std::vector< T > &vec, int _N, int _dim)
- void **load** (T *source, int _N, int _dim)
- void **save** (const std::string &path)
- **Matrix** (const std::string &path)

Private Attributes

- int **dim**
- int **N**
- T * **dims**

5.4.1 Detailed Description

```
template<class T>class Lshbox::Matrix< T >
```

Dataset management class. A dataset is maintained as a matrix in memory.

The file contains N D-dimensional vectors of single precision floating point numbers.

Such binary files can be accessed using `Lshbox::Matrix<double>`.

Definition at line 43 of file `matrix.h`.

5.4.2 Constructor & Destructor Documentation

5.4.2.1 `Matrix ()` `[inline]`

Definition at line 74 of file `matrix.h`.

5.4.2.2 `Matrix (int_dim, int_N)` `[inline]`

Definition at line 75 of file `matrix.h`.

Here is the call graph for this function:



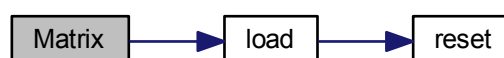
5.4.2.3 `~Matrix ()` `[inline]`

Definition at line 79 of file `matrix.h`.

5.4.2.4 `Matrix (const std::string & path)` `[inline]`

Definition at line 172 of file `matrix.h`.

Here is the call graph for this function:



5.4.3 Member Function Documentation

5.4.3.1 void free (void) [inline]

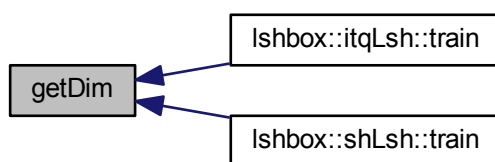
Definition at line 65 of file matrix.h.

5.4.3.2 int getDim () const [inline]

Get the dimension.

Definition at line 103 of file matrix.h.

Here is the caller graph for this function:

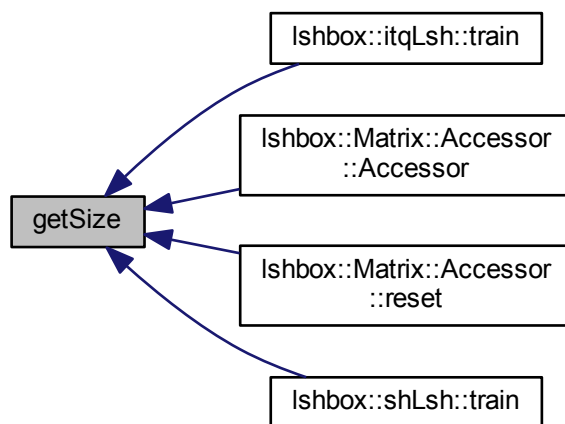


5.4.3.3 int getSize () const [inline]

Get the size.

Definition at line 110 of file matrix.h.

Here is the caller graph for this function:



5.4.3.4 void load (const std::string & path) [inline]

Load the **Matrix** (p. 22) from a binary file.

Definition at line 117 of file matrix.h.

Here is the call graph for this function:



Here is the caller graph for this function:



5.4.3.5 void load (std::vector< T > & vec, int _N, int _dim) [inline]

Load the **Matrix** (p. 22) from std::vector<T>.

Parameters

<i>vec</i>	The reference of std::vector<T>.
<i>_N</i>	Number of vectors
<i>_dim</i>	Dimension of each vector

Definition at line 134 of file matrix.h.

Here is the call graph for this function:



5.4.3.6 void load (T * source, int _N, int _dim) [inline]

Load the **Matrix** (p. 22) from T*.

Parameters

<i>source</i>	The pointer to T*.
<i>_N</i>	Number of vectors
<i>_dim</i>	Dimension of each vector

Definition at line 149 of file matrix.h.

Here is the call graph for this function:



5.4.3.7 `const T* operator[] (int i) const` `[inline]`

Access the ith vector.

Definition at line 89 of file matrix.h.

5.4.3.8 `T* operator[] (int i)` `[inline]`

Access the ith vector.

Definition at line 96 of file matrix.h.

5.4.3.9 `void reset (int _dim, int _N)` `[inline]`

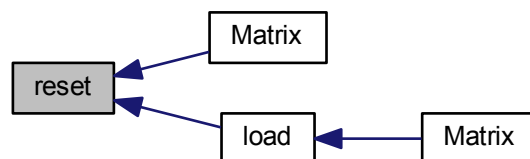
Reset the size.

Parameters

<i>_dim</i>	Dimension of each vector
<i>_N</i>	Number of vectors

Definition at line 55 of file matrix.h.

Here is the caller graph for this function:



5.4.3.10 void save (const std::string & path) [inline]

Save the **Matrix** (p. 22) as a binary file.

Definition at line 160 of file matrix.h.

5.4.4 Member Data Documentation

5.4.4.1 int dim [private]

Definition at line 45 of file matrix.h.

5.4.4.2 T* dims [private]

Definition at line 47 of file matrix.h.

5.4.4.3 int N [private]

Definition at line 46 of file matrix.h.

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

- include/lshbox/**matrix.h**

5.5 Metric< DATATYPE > Class Template Reference

```
#include <metric.h>
```

Collaboration diagram for Metric< DATATYPE >:

Metric< DATATYPE >
- type_ - dim_
+ Metric() + ~Metric() + dim() + dist()

Public Member Functions

- **Metric** (unsigned **dim**, unsigned type)
- **~Metric** ()
- unsigned **dim** () const
- float **dist** (const DATATYPE *vec1, const DATATYPE *vec2) const

Private Attributes

- unsigned **type_**
- unsigned **dim_**

5.5.1 Detailed Description

template<typename DATATYPE>class Lshbox::Metric< DATATYPE >

Use for common distance functions.

Definition at line 46 of file metric.h.

5.5.2 Constructor & Destructor Documentation

5.5.2.1 Metric (unsigned *dim*, unsigned *type*) [inline]

Constructor for this class.

Parameters

<i>dim</i>	Dimension of each vector
<i>type</i>	The way to measure the distance, you can choose 1(L1_DIST) or 2(L2_DIST)

Definition at line 57 of file metric.h.

5.5.2.2 ~Metric () [inline]

Definition at line 58 of file metric.h.

5.5.3 Member Function Documentation

5.5.3.1 unsigned dim () const [inline]

Get the dimension of the vectors

Definition at line 62 of file metric.h.

5.5.3.2 float dist (const DATATYPE * *vec1*, const DATATYPE * *vec2*) const [inline]

measure the distance.

Parameters

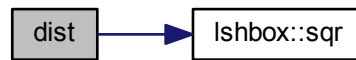
<i>vec1</i>	The first vector
<i>vec2</i>	The second vector

Returns

The distance

Definition at line 73 of file metric.h.

Here is the call graph for this function:

**5.5.4 Member Data Documentation****5.5.4.1 unsigned dim_ [private]**

Definition at line 49 of file metric.h.

5.5.4.2 unsigned type_ [private]

Definition at line 48 of file metric.h.

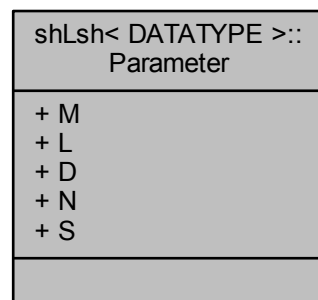
The documentation for this class was generated from the following file:

- include/lshbox/**metric.h**

5.6 shLsh< DATATYPE >::Parameter Struct Reference

```
#include <shlsh.h>
```

Collaboration diagram for shLsh< DATATYPE >::Parameter:



Public Attributes

- unsigned **M**
Hash table size.
- unsigned **L**
Number of hash tables.
- unsigned **D**
*Dimension of the vector, it can be obtained from the instance of **Matrix** (p. 22).*
- unsigned **N**
Binary code bytes.
- unsigned **S**
Size of vectors in train.

5.6.1 Detailed Description

template<typename DATATYPE = float>struct lshbox::shLsh< DATATYPE >::Parameter

Definition at line 51 of file shlsh.h.

5.6.2 Member Data Documentation

5.6.2.1 unsigned D

Dimension of the vector, it can be obtained from the instance of **Matrix** (p. 22).

Definition at line 58 of file shlsh.h.

5.6.2.2 unsigned L

Number of hash tables.

Definition at line 56 of file shlsh.h.

5.6.2.3 unsigned M

Hash table size.

Definition at line 54 of file shlsh.h.

5.6.2.4 unsigned N

Binary code bytes.

Definition at line 60 of file shlsh.h.

5.6.2.5 unsigned S

Size of vectors in train.

Definition at line 62 of file shlsh.h.

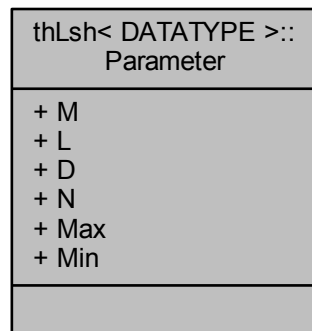
The documentation for this struct was generated from the following file:

- include/lshbox/**shlsh.h**

5.7 thLsh< DATATYPE >::Parameter Struct Reference

```
#include <thlsh.h>
```

Collaboration diagram for thLsh< DATATYPE >::Parameter:



Public Attributes

- unsigned **M**
Hash table size.
- unsigned **L**
Number of hash tables.
- unsigned **D**
*Dimension of the vector, it can be obtained from the instance of **Matrix** (p. 22).*
- unsigned **N**
Binary code bytes.
- float **Max**
Upper bound of each dimension.
- float **Min**
Lower bound of each dimension.

5.7.1 Detailed Description

```
template<typename DATATYPE = float>struct lshbox::thLsh< DATATYPE >::Parameter
```

Definition at line 57 of file thlsh.h.

5.7.2 Member Data Documentation

5.7.2.1 unsigned D

Dimension of the vector, it can be obtained from the instance of **Matrix** (p. 22).

Definition at line 64 of file thlsh.h.

5.7.2.2 unsigned L

Number of hash tables.

Definition at line 62 of file thlsh.h.

5.7.2.3 unsigned M

Hash table size.

Definition at line 60 of file thlsh.h.

5.7.2.4 float Max

Upper bound of each dimension.

Definition at line 68 of file thlsh.h.

5.7.2.5 float Min

Lower bound of each dimension.

Definition at line 70 of file thlsh.h.

5.7.2.6 unsigned N

Binary code bytes.

Definition at line 66 of file thlsh.h.

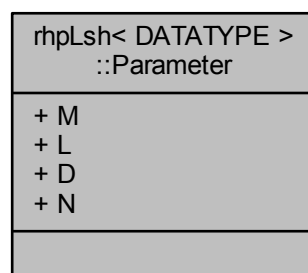
The documentation for this struct was generated from the following file:

- include/lshbox/thlsh.h

5.8 rhpLsh< DATATYPE >::Parameter Struct Reference

```
#include <rhplsh.h>
```

Collaboration diagram for rhpLsh< DATATYPE >::Parameter:



Public Attributes

- unsigned **M**
Hash table size.
- unsigned **L**
Number of hash tables.
- unsigned **D**
*Dimension of the vector, it can be obtained from the instance of **Matrix** (p. 22).*
- unsigned **N**
Binary code bytes.

5.8.1 Detailed Description

template<typename DATATYPE = float>struct Lshbox::rhpLsh< DATATYPE >::Parameter

Definition at line 52 of file rhplsh.h.

5.8.2 Member Data Documentation

5.8.2.1 unsigned D

Dimension of the vector, it can be obtained from the instance of **Matrix** (p. 22).

Definition at line 59 of file rhplsh.h.

5.8.2.2 unsigned L

Number of hash tables.

Definition at line 57 of file rhplsh.h.

5.8.2.3 unsigned M

Hash table size.

Definition at line 55 of file rhplsh.h.

5.8.2.4 unsigned N

Binary code bytes.

Definition at line 61 of file rhplsh.h.

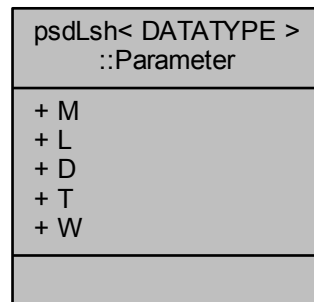
The documentation for this struct was generated from the following file:

- include/lshbox/rhplsh.h

5.9 psdLsh< DATATYPE >::Parameter Struct Reference

```
#include <psdlsh.h>
```

Collaboration diagram for psdLsh< DATATYPE >::Parameter:



Public Attributes

- unsigned **M**
Hash table size.
- unsigned **L**
Number of hash tables.
- unsigned **D**
*Dimension of the vector, it can be obtained from the instance of **Matrix** (p. 22).*
- unsigned **T**
Index mode, you can choose 1(CAUCHY) or 2(GAUSSIAN)
- float **W**
Window size.

5.9.1 Detailed Description

```
template<typename DATATYPE = float>struct Ishbox::psdLsh< DATATYPE >::Parameter
```

Definition at line 54 of file psdLsh.h.

5.9.2 Member Data Documentation

5.9.2.1 unsigned D

Dimension of the vector, it can be obtained from the instance of **Matrix** (p. 22).

Definition at line 61 of file psdLsh.h.

5.9.2.2 unsigned L

Number of hash tables.

Definition at line 59 of file psdLsh.h.

5.9.2.3 unsigned M

Hash table size.

Definition at line 57 of file psdlsh.h.

5.9.2.4 unsigned T

Index mode, you can choose 1(CAUCHY) or 2(GAUSSIAN)

Definition at line 63 of file psdlsh.h.

5.9.2.5 float W

Window size.

Definition at line 65 of file psdlsh.h.

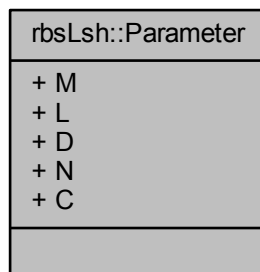
The documentation for this struct was generated from the following file:

- include/lshbox/**psdlsh.h**

5.10 rbsLsh::Parameter Struct Reference

```
#include <rbslsh.h>
```

Collaboration diagram for rbsLsh::Parameter:



Public Attributes

- unsigned **M**
Hash table size.
- unsigned **L**
Number of hash tables.
- unsigned **D**
*Dimension of the vector, it can be obtained from the instance of **Matrix** (p. 22).*
- unsigned **N**
Binary code bytes.

- unsigned C

The Difference between upper and lower bound of each dimension.

5.10.1 Detailed Description

Definition at line 54 of file rbslsh.h.

5.10.2 Member Data Documentation

5.10.2.1 unsigned C

The Difference between upper and lower bound of each dimension.

Definition at line 65 of file rbslsh.h.

5.10.2.2 unsigned D

Dimension of the vector, it can be obtained from the instance of **Matrix** (p. 22).

Definition at line 61 of file rbslsh.h.

5.10.2.3 unsigned L

Number of hash tables.

Definition at line 59 of file rbslsh.h.

5.10.2.4 unsigned M

Hash table size.

Definition at line 57 of file rbslsh.h.

5.10.2.5 unsigned N

Binary code bytes.

Definition at line 63 of file rbslsh.h.

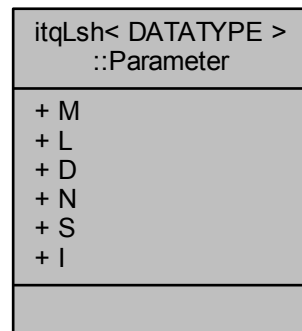
The documentation for this struct was generated from the following file:

- include/lshbox/**rbslsh.h**

5.11 itqLsh< DATATYPE >::Parameter Struct Reference

```
#include <itqlsh.h>
```

Collaboration diagram for itqLsh< DATATYPE >::Parameter:



Public Attributes

- unsigned **M**
Hash table size.
- unsigned **L**
Number of hash tables.
- unsigned **D**
*Dimension of the vector, it can be obtained from the instance of **Matrix** (p. 22).*
- unsigned **N**
Binary code bytes.
- unsigned **S**
Size of vectors in train.
- unsigned **I**
Training iterations.

5.11.1 Detailed Description

```
template<typename DATATYPE = float>struct lshbox::itqLsh< DATATYPE >::Parameter
```

Definition at line 53 of file itqlsh.h.

5.11.2 Member Data Documentation

5.11.2.1 unsigned D

Dimension of the vector, it can be obtained from the instance of **Matrix** (p. 22).

Definition at line 60 of file itqlsh.h.

5.11.2.2 unsigned I

Training iterations.

Definition at line 66 of file itqlsh.h.

5.11.2.3 unsigned L

Number of hash tables.

Definition at line 58 of file itqlsh.h.

5.11.2.4 unsigned M

Hash table size.

Definition at line 56 of file itqlsh.h.

5.11.2.5 unsigned N

Binary code bytes.

Definition at line 62 of file itqlsh.h.

5.11.2.6 unsigned S

Size of vectors in train.

Definition at line 64 of file itqlsh.h.

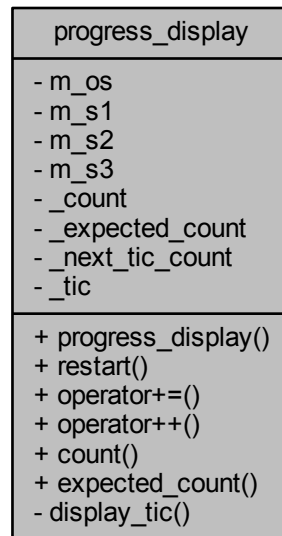
The documentation for this struct was generated from the following file:

- include/lshbox/**itqlsh.h**

5.12 progress_display Class Reference

```
#include <basis.h>
```

Collaboration diagram for progress_display:



Public Member Functions

- **progress_display** (unsigned long **expected_count**, std::ostream &os=std::cout, const std::string &s1="\n", const std::string &s2="", const std::string &s3="")
- void **restart** (unsigned long **expected_count**)
- unsigned long **operator+=** (unsigned long increment)
- unsigned long **operator++** ()
- unsigned long **count** () const
- unsigned long **expected_count** () const

Private Member Functions

- void **display_tic** ()

Private Attributes

- std::ostream & **m_os**
- const std::string **m_s1**
- const std::string **m_s2**
- const std::string **m_s3**
- unsigned long **_count**
- unsigned long **_expected_count**
- unsigned long **_next_tic_count**
- unsigned **_tic**

5.12.1 Detailed Description

Displays an appropriate indication of progress at an appropriate place in an appropriate form.

If you are familiar with the Boost library, you should be very familiar with this class.

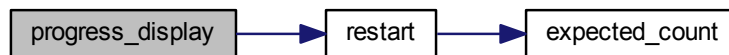
Definition at line 51 of file basis.h.

5.12.2 Constructor & Destructor Documentation

5.12.2.1 `progress_display (unsigned long expected_count, std::ostream & os = std::cout, const std::string & s1 = "\n", const std::string & s2 = "", const std::string & s3 = "")` `[inline],[explicit]`

Definition at line 54 of file basis.h.

Here is the call graph for this function:



5.12.3 Member Function Documentation

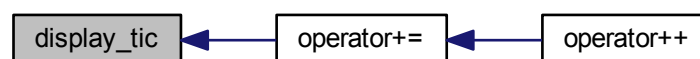
5.12.3.1 `unsigned long count () const` `[inline]`

Definition at line 89 of file basis.h.

5.12.3.2 `void display_tic ()` `[inline],[private]`

Definition at line 104 of file basis.h.

Here is the caller graph for this function:



5.12.3.3 `unsigned long expected_count () const` `[inline]`

Definition at line 93 of file basis.h.

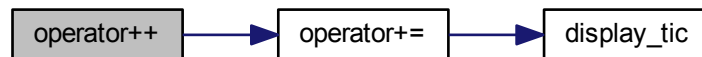
Here is the caller graph for this function:



5.12.3.4 unsigned long operator++() [inline]

Definition at line 85 of file basis.h.

Here is the call graph for this function:



5.12.3.5 unsigned long operator+=(unsigned long *increment*) [inline]

Definition at line 77 of file basis.h.

Here is the call graph for this function:



Here is the caller graph for this function:



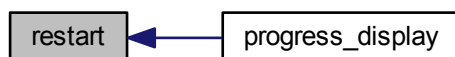
5.12.3.6 void restart (unsigned long *expected_count*) [inline]

Definition at line 64 of file basis.h.

Here is the call graph for this function:



Here is the caller graph for this function:



5.12.4 Member Data Documentation

5.12.4.1 unsigned long _count [private]

Definition at line 102 of file basis.h.

5.12.4.2 unsigned long _expected_count [private]

Definition at line 102 of file basis.h.

5.12.4.3 unsigned long _next_tic_count [private]

Definition at line 102 of file basis.h.

5.12.4.4 unsigned _tic [private]

Definition at line 103 of file basis.h.

5.12.4.5 std::ostream& m_os [private]

Definition at line 98 of file basis.h.

5.12.4.6 const std::string m_s1 [private]

Definition at line 99 of file basis.h.

5.12.4.7 `const std::string m_s2` `[private]`

Definition at line 100 of file basis.h.

5.12.4.8 `const std::string m_s3` `[private]`

Definition at line 101 of file basis.h.

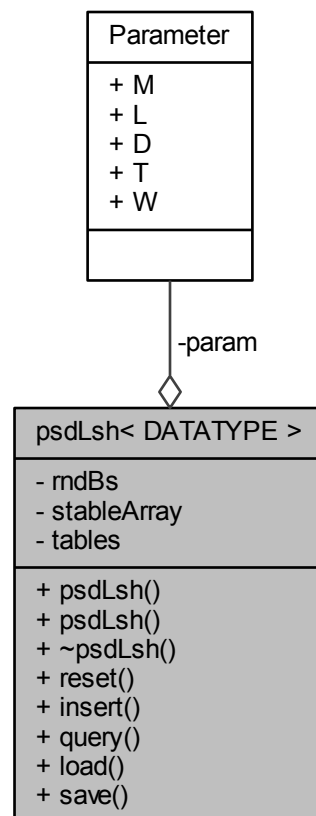
The documentation for this class was generated from the following file:

- include/lshbox/**basis.h**

5.13 `psdLsh< DATATYPE >` Class Template Reference

```
#include <psdlsh.h>
```

Collaboration diagram for `psdLsh< DATATYPE >`:



Classes

- struct **Parameter**

Public Member Functions

- **psdLsh** ()
- **psdLsh** (const **Parameter** ¶m_)
- **~psdLsh** ()
- void **reset** (const **Parameter** ¶m_)
- void **insert** (unsigned key, DATATYPE *domin)
- template<typename SCANNER >
void **query** (DATATYPE *domin, SCANNER &scanner)
- void **load** (const std::string &file)
- void **save** (const std::string &file)

Private Attributes

- **Parameter** param
- std::vector< float > **rndBs**
- std::vector< std::vector< float > > **stableArray**
- std::vector< std::map
< unsigned, std::vector
< unsigned > > > **tables**

5.13.1 Detailed Description

template<typename DATATYPE = float>class Ishbox::psdLsh< DATATYPE >

Locality-Sensitive Hashing Scheme Based on p-Stable Distributions.

For more information on p-stable distribution based LSH, see the following reference.

Mayur Datar , Nicole Immorlica , Piotr Indyk , Vahab S. Mirrokni,
Locality-sensitive hashing scheme based on p-stable distributions,
Proceedings of the twentieth annual symposium on Computational geometry, June
08-11, 2004, Brooklyn, New York, USA.

Definition at line 49 of file psdsh.h.

5.13.2 Constructor & Destructor Documentation

5.13.2.1 **psdLsh** () [inline]

Definition at line 67 of file psdsh.h.

5.13.2.2 **psdLsh** (const **Parameter** & param_) [inline]

Definition at line 68 of file psdsh.h.

Here is the call graph for this function:



5.13.2.3 ~psdLsh () [inline]

Definition at line 72 of file psdLsh.h.

5.13.3 Member Function Documentation

5.13.3.1 void insert (unsigned *key*, DATATYPE * *domin*) [inline]

Insert a vector to the index.

Parameters

<i>key</i>	The sequence number of vector
<i>domin</i>	The pointer to the vector

Definition at line 126 of file psdLsh.h.

5.13.3.2 void load (const std::string & *file*) [inline]

Load the index from binary file.

Parameters

<i>file</i>	The path of binary file.
-------------	--------------------------

Definition at line 170 of file psdLsh.h.

5.13.3.3 void query (DATATYPE * *domin*, SCANNER & *scanner*) [inline]

Query the approximate nearest neighborholds.

Parameters

<i>domin</i>	The pointer to the vector
<i>scanner</i>	Top-K scanner, use for scan the approximate nearest neighborholds

Definition at line 146 of file psdLsh.h.

5.13.3.4 void reset (const Parameter & *param_*) [inline]

Reset the parameter setting

Parameters

<i>param_</i>	A instance of psdLsh<DATATYPE>::Parameter, which contains the necessary parameters
---------------	--

Definition at line 79 of file psdLsh.h.

Here is the caller graph for this function:



5.13.3.5 `void save (const std::string & file) [inline]`

Save the index as binary file.

Parameters

<i>file</i>	The path of binary file.
-------------	--------------------------

Definition at line 204 of file psdLsh.h.

5.13.4 Member Data Documentation

5.13.4.1 `Parameter param [private]`

Definition at line 229 of file psdLsh.h.

5.13.4.2 `std::vector<float> rndBs [private]`

Definition at line 230 of file psdLsh.h.

5.13.4.3 `std::vector<std::vector<float> > stableArray [private]`

Definition at line 231 of file psdLsh.h.

5.13.4.4 `std::vector<std::map<unsigned, std::vector<unsigned> > > tables [private]`

Definition at line 232 of file psdLsh.h.

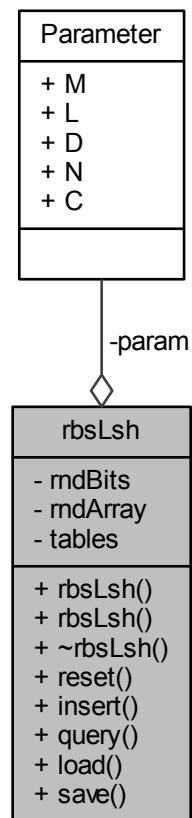
The documentation for this class was generated from the following file:

- include/lshbox/**psdLsh.h**

5.14 rbsLsh Class Reference

```
#include <rbslsh.h>
```

Collaboration diagram for rbsLsh:



Classes

- struct **Parameter**

Public Member Functions

- **rbsLsh** ()
- **rbsLsh** (const **Parameter** ¶m_)
- **~rbsLsh** ()
- void **reset** (const **Parameter** ¶m_)
- void **insert** (unsigned key, unsigned *domin)
- template<typename SCANNER >
void **query** (unsigned *domin, SCANNER &scanner)
- void **load** (const std::string &file)
- void **save** (const std::string &file)

Private Attributes

- **Parameter** param

- `std::vector< std::vector< unsigned > > rndBits`
- `std::vector< std::vector< unsigned > > rndArray`
- `std::vector< std::map< unsigned, std::vector< unsigned > > > tables`

5.14.1 Detailed Description

Locality-Sensitive Hashing Scheme Based on Random Bits Sampling.

For more information on random bits sampling based LSH, see the following reference.

P. Indyk and R. Motwani. Approximate Nearest Neighbor - Towards Removing the Curse of Dimensionality. In Proceedings of the 30th Symposium on Theory of Computing, 1998, pp. 604-613.

A. Gionis, P. Indyk, and R. Motwani. Similarity search in high dimensions via hashing. Proceedings of the 25th International Conference on Very Large Data Bases (VLDB), 1999.

Definition at line 51 of file `rbslsh.h`.

5.14.2 Constructor & Destructor Documentation

5.14.2.1 `rbsLsh ()` `[inline]`

Definition at line 67 of file `rbslsh.h`.

5.14.2.2 `rbsLsh (const Parameter & param_)` `[inline]`

Definition at line 68 of file `rbslsh.h`.

Here is the call graph for this function:



5.14.2.3 `~rbsLsh ()` `[inline]`

Definition at line 72 of file `rbslsh.h`.

5.14.3 Member Function Documentation

5.14.3.1 `void insert (unsigned key, unsigned * domin)` `[inline]`

Insert a vector to the index.

Parameters

<i>key</i>	The sequence number of vector
<i>domin</i>	The pointer to the vector

Definition at line 114 of file rbslsh.h.

5.14.3.2 void load (const std::string & *file*) [inline]

Load the index from binary file.

Parameters

<i>file</i>	The path of binary file.
-------------	--------------------------

Definition at line 162 of file rbslsh.h.

5.14.3.3 void query (unsigned * *domin*, SCANNER & *scanner*) [inline]

Query the approximate nearest neighborholds.

Parameters

<i>domin</i>	The pointer to the vector
<i>scanner</i>	Top-K scanner, use for scan the approximate nearest neighborholds

Definition at line 137 of file rbslsh.h.

5.14.3.4 void reset (const Parameter & *param_*) [inline]

Reset the parameter setting

Parameters

<i>param_</i>	A instance of rbsLsh::Parametor, which contains the necessary parameters
---------------	--

Definition at line 79 of file rbslsh.h.

Here is the caller graph for this function:



5.14.3.5 void save (const std::string & *file*) [inline]

Save the index as binary file.

Parameters

<i>file</i>	The path of binary file.
-------------	--------------------------

Definition at line 198 of file rbslsh.h.

5.14.4 Member Data Documentation

5.14.4.1 Parameter param [private]

Definition at line 224 of file rbslsh.h.

5.14.4.2 std::vector<std::vector<unsigned>> rndArray [private]

Definition at line 226 of file rbslsh.h.

5.14.4.3 std::vector<std::vector<unsigned>> rndBits [private]

Definition at line 225 of file rbslsh.h.

5.14.4.4 std::vector<std::map<unsigned, std::vector<unsigned>>> tables [private]

Definition at line 227 of file rbslsh.h.

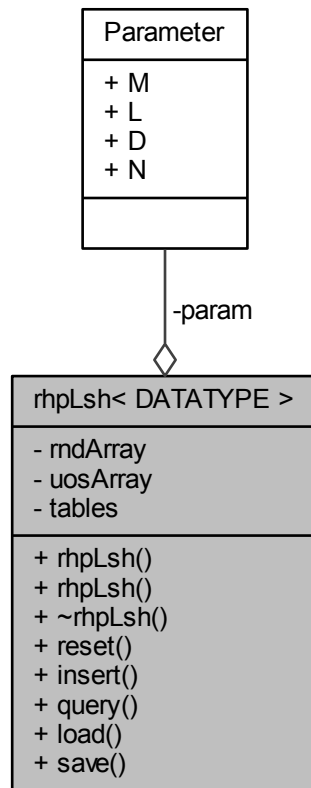
The documentation for this class was generated from the following file:

- include/lshbox/rbslsh.h

5.15 rhpLsh< DATATYPE > Class Template Reference

```
#include <rhplsh.h>
```

Collaboration diagram for `rhplsh< DATATYPE >`:



Classes

- struct **Parameter**

Public Member Functions

- **rhplsh** ()
- **rhplsh** (const **Parameter** ¶m_)
- **~rhplsh** ()
- void **reset** (const **Parameter** ¶m_)
- void **insert** (unsigned key, DATATYPE *domin)
- template<typename SCANNER >
void **query** (DATATYPE *domin, SCANNER &scanner)
- void **load** (const std::string &file)
- void **save** (const std::string &file)

Private Attributes

- **Parameter** param

- `std::vector< std::vector< unsigned > > rndArray`
- `std::vector< std::vector< std::vector< float > > > uosArray`
- `std::vector< std::map< unsigned, std::vector< unsigned > > > tables`

5.15.1 Detailed Description

`template<typename DATATYPE = float>class Ishbox::rhpLsh< DATATYPE >`

Locality-Sensitive Hashing Scheme Based on Random Hyperplane.

For more information on random hyperplane based LSH, see the following reference.

Charikar, M. S. 2002. Similarity estimation techniques from rounding algorithms. In Proceedings of the Thiry-Fourth Annual ACM Symposium on theory of Computing (Montreal, Quebec, Canada, May 19 - 21, 2002). STOC '02. ACM, New York, NY, 380-388. DOI= <http://doi.acm.org/10.1145/509907.509965>

Definition at line 49 of file `rhplsh.h`.

5.15.2 Constructor & Destructor Documentation

5.15.2.1 `rhpLsh()` [inline]

Definition at line 63 of file `rhplsh.h`.

5.15.2.2 `rhpLsh(const Parameter & param_)` [inline]

Definition at line 64 of file `rhplsh.h`.

Here is the call graph for this function:



5.15.2.3 `~rhpLsh()` [inline]

Definition at line 68 of file `rhplsh.h`.

5.15.3 Member Function Documentation

5.15.3.1 `void insert(unsigned key, DATATYPE * domin)` [inline]

Insert a vector to the index.

Parameters

<i>key</i>	The sequence number of vector
<i>domin</i>	The pointer to the vector

Definition at line 109 of file rhplsh.h.

5.15.3.2 void load (const std::string & *file*) [inline]

Load the index from binary file.

Parameters

<i>file</i>	The path of binary file.
-------------	--------------------------

Definition at line 169 of file rhplsh.h.

5.15.3.3 void query (DATATYPE * *domin*, SCANNER & *scanner*) [inline]

Query the approximate nearest neighborhoods.

Parameters

<i>domin</i>	The pointer to the vector
<i>scanner</i>	Top-K scanner, use for scan the approximate nearest neighborhoods

Definition at line 137 of file rhplsh.h.

5.15.3.4 void reset (const Parameter & *param_*) [inline]

Reset the parameter setting

Parameters

<i>param_</i>	A instance of rhpLsh<DATATYPE>::Parametor, which contains the necessary parameters
---------------	--

Definition at line 75 of file rhplsh.h.

Here is the caller graph for this function:



5.15.3.5 void save (const std::string & *file*) [inline]

Save the index as binary file.

Parameters

<i>file</i>	The path of binary file.
-------------	--------------------------

Definition at line 208 of file rhplsh.h.

5.15.4 Member Data Documentation

5.15.4.1 Parameter param [private]

Definition at line 236 of file rhplsh.h.

5.15.4.2 std::vector<std::vector<unsigned>> rndArray [private]

Definition at line 237 of file rhplsh.h.

5.15.4.3 std::vector<std::map<unsigned, std::vector<unsigned>>> tables [private]

Definition at line 239 of file rhplsh.h.

5.15.4.4 std::vector<std::vector<std::vector<float>>> uosArray [private]

Definition at line 238 of file rhplsh.h.

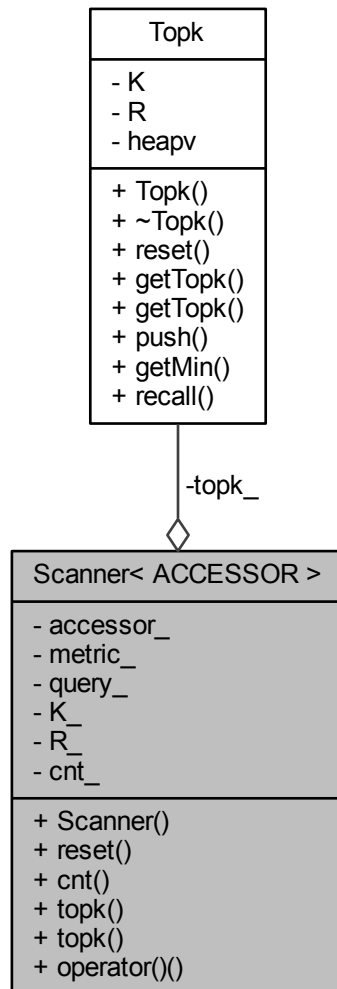
The documentation for this class was generated from the following file:

- include/lshbox/rhplsh.h

5.16 Scanner< ACCESSOR > Class Template Reference

```
#include <topk.h>
```

Collaboration diagram for Scanner< ACCESSOR >:



Public Types

- typedef ACCESSOR::Value **Value**
- typedef ACCESSOR::DATATYPE **DATATYPE**

Public Member Functions

- **Scanner** (const ACCESSOR &accessor, const **Metric**< **DATATYPE** > &metric, unsigned K, float R=std::numeric_limits< float >::max())
- void **reset** (**Value** query)
- unsigned **cnt** () const
- const **Topk** & **topk** () const
- **Topk** & **topk** ()
- void **operator**() (unsigned key)

Private Attributes

- ACCESSOR **accessor_**
- Metric< DATATYPE > **metric_**
- Topk **topk_**
- Value **query_**
- unsigned **K_**
- float **R_**
- unsigned **cnt_**

5.16.1 Detailed Description

template<typename ACCESSOR>class lshbox::Scanner< ACCESSOR >

Top-K scanner.

Scans keys for top-K query, this is the object passed into the LSH query interface.

Definition at line 130 of file topk.h.

5.16.2 Member Typedef Documentation

5.16.2.1 typedef ACCESSOR::DATATYPE DATATYPE

Definition at line 134 of file topk.h.

5.16.2.2 typedef ACCESSOR::Value Value

Definition at line 133 of file topk.h.

5.16.3 Constructor & Destructor Documentation

5.16.3.1 Scanner (const ACCESSOR & *accessor*, const Metric< DATATYPE > & *metric*, unsigned *K*, float *R* = std::numeric_limits<float>::max()) [inline]

Constructor for this class.

Parameters

<i>accessor</i>	The scanner use accessor to retrieve values from keys.
<i>metric</i>	The distance metric.
<i>K</i>	Value used to reset internal Topk (p. 75) class.
<i>R</i>	Value used to reset internal Topk (p. 75) class.

Definition at line 143 of file topk.h.

5.16.4 Member Function Documentation

5.16.4.1 unsigned cnt () const [inline]

Number of points scanned for the current query.

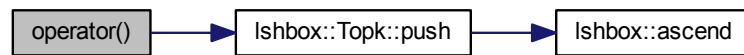
Definition at line 162 of file topk.h.

5.16.4.2 `void operator() (unsigned key) [inline]`

Update the current query by scanning key, this is normally invoked by the LSH index structure.

Definition at line 184 of file `topk.h`.

Here is the call graph for this function:

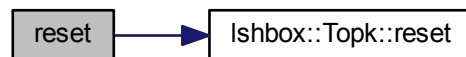


5.16.4.3 `void reset (Value query) [inline]`

Reset the query, this function should be invoked before each query.

Definition at line 152 of file `topk.h`.

Here is the call graph for this function:



5.16.4.4 `const Topk& topk () const [inline]`

TopK results.

Definition at line 169 of file `topk.h`.

5.16.4.5 `Topk& topk () [inline]`

TopK results.

Definition at line 176 of file `topk.h`.

5.16.5 Member Data Documentation

5.16.5.1 `ACCESSOR accessor_ [private]`

Definition at line 193 of file `topk.h`.

5.16.5.2 unsigned cnt_ [private]

Definition at line 199 of file topk.h.

5.16.5.3 unsigned K_ [private]

Definition at line 197 of file topk.h.

5.16.5.4 Metric<DATATYPE> metric_ [private]

Definition at line 194 of file topk.h.

5.16.5.5 Value query_ [private]

Definition at line 196 of file topk.h.

5.16.5.6 float R_ [private]

Definition at line 198 of file topk.h.

5.16.5.7 Topk topk_ [private]

Definition at line 195 of file topk.h.

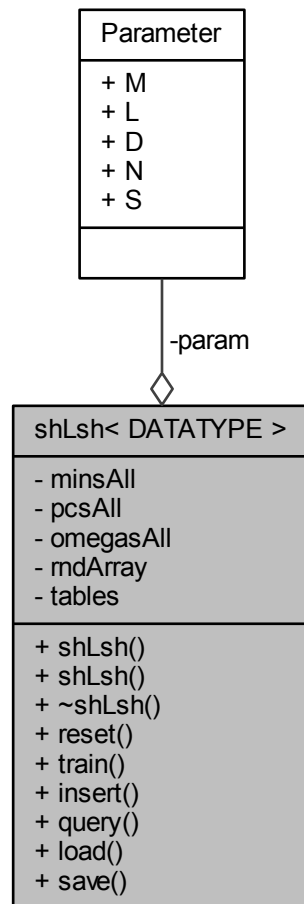
The documentation for this class was generated from the following file:

- include/lshbox/topk.h

5.17 shLsh< DATATYPE > Class Template Reference

```
#include <shlsh.h>
```

Collaboration diagram for shLsh< DATATYPE >:



Classes

- struct **Parameter**

Public Member Functions

- **shLsh** ()
- **shLsh** (const **Parameter** ¶m_)
- **~shLsh** ()
- void **reset** (const **Parameter** ¶m_)
- void **train** (**Matrix**< DATATYPE > &data)
- void **insert** (unsigned key, DATATYPE *domin)
- template<typename SCANNER >
void **query** (const DATATYPE *domin, SCANNER &scanner)
- void **load** (const std::string &file)
- void **save** (const std::string &file)

Private Attributes

- **Parameter param**
- `std::vector< std::vector< double > > minsAll`
- `std::vector< std::vector< std::vector< float > > > pcsAll`
- `std::vector< std::vector< std::vector< float > > > omegasAll`
- `std::vector< std::vector< unsigned > > rndArray`
- `std::vector< std::map< unsigned, std::vector< unsigned > > > tables`

5.17.1 Detailed Description

`template<typename DATATYPE = float>class Ishbox::shLsh< DATATYPE >`

Locality-Sensitive Hashing Scheme Based on Spectral Hashing.

For more information on spectral hashing based LSH, see the following reference.

Y. Weiss, A. Torralba, R. Fergus. Spectral Hashing.
Advances in Neural Information Processing Systems, 2008.

Definition at line 48 of file shlsh.h.

5.17.2 Constructor & Destructor Documentation

5.17.2.1 shLsh() [inline]

Definition at line 64 of file shlsh.h.

5.17.2.2 shLsh(const Parameter & param_) [inline]

Definition at line 65 of file shlsh.h.

Here is the call graph for this function:



5.17.2.3 ~shLsh() [inline]

Definition at line 69 of file shlsh.h.

5.17.3 Member Function Documentation

5.17.3.1 `void insert (unsigned key, DATATYPE * domin)` `[inline]`

Insert a vector to the index.

Parameters

<i>key</i>	The sequence number of vector
<i>domin</i>	The pointer to the vector

Definition at line 213 of file shlsh.h.

Here is the caller graph for this function:



5.17.3.2 void load (const std::string & file) [inline]

Load the index from binary file.

Parameters

<i>file</i>	The path of binary file.
-------------	--------------------------

Definition at line 291 of file shlsh.h.

5.17.3.3 void query (const DATATYPE * domin, SCANNER & scanner) [inline]

Query the approximate nearest neighborholds.

Parameters

<i>domin</i>	The pointer to the vector
<i>scanner</i>	Top-K scanner, use for scan the approximate nearest neighborholds

Definition at line 250 of file shlsh.h.

5.17.3.4 void reset (const Parameter & param_) [inline]

Reset the parameter setting

Parameters

<i>param_</i>	A instance of shLsh<DATATYPE>::Parametor, which contains the necessary parameters
---------------	---

Definition at line 76 of file shlsh.h.

Here is the caller graph for this function:



5.17.3.5 void save (const std::string & file) [inline]

Save the index as binary file.

Parameters

<i>file</i>	The path of binary file.
-------------	--------------------------

Definition at line 338 of file shlsh.h.

5.17.3.6 void train (Matrix< DATATYPE > & data) [inline]

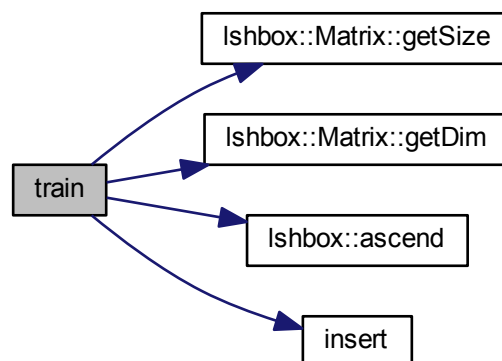
Train the data to get several groups of suitable vector for index.

Parameters

<i>data</i>	A instance of Matrix<DATATYPE>, most of the time, is the search library.
-------------	--

Definition at line 99 of file shlsh.h.

Here is the call graph for this function:



5.17.4 Member Data Documentation

5.17.4.1 `std::vector<std::vector<double>>> minsAll` [private]

Definition at line 370 of file shlsh.h.

5.17.4.2 `std::vector<std::vector<std::vector<float>>>> omegasAll` [private]

Definition at line 372 of file shlsh.h.

5.17.4.3 **Parameter** `param` [private]

Definition at line 369 of file shlsh.h.

5.17.4.4 `std::vector<std::vector<std::vector<float>>>> pcsAll` [private]

Definition at line 371 of file shlsh.h.

5.17.4.5 `std::vector<std::vector<unsigned>>> rndArray` [private]

Definition at line 373 of file shlsh.h.

5.17.4.6 `std::vector<std::map<unsigned, std::vector<unsigned>>>> tables` [private]

Definition at line 374 of file shlsh.h.

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

- include/lshbox/**shlsh.h**

5.18 Stat Class Reference

```
#include <eval.h>
```

Collaboration diagram for Stat:

Stat
<ul style="list-style-type: none">- count- sum- sum2- min- max
<ul style="list-style-type: none">+ Stat()+ ~Stat()+ reset()+ append()+ operator<<()+ getCount()+ getSum()+ getAvg()+ getMax()+ getMin()+ getStd()+ merge()

Public Member Functions

- **Stat** ()
- **~Stat** ()
- void **reset** ()
- void **append** (float r)
- **Stat** & **operator<<** (float r)
- int **getCount** () const
- float **getSum** () const
- float **getAvg** () const
- float **getMax** () const
- float **getMin** () const
- float **getStd** () const
- void **merge** (const **Stat** &stat)

Private Attributes

- int **count**
- float **sum**
- float **sum2**
- float **min**
- float **max**

5.18.1 Detailed Description

Use for basic statistics, and the interface is self-evident.

Usage:

```
Stat stat;  
  
stat << 1.0 << 2.0 << 3.0;  
  
Stat stat2;  
stat2 << 3.0 << 5.0 << 6.0;  
  
stat.merge(stat2);  
  
stat.getCount();  
stat.getSum();  
stat.getMax();  
stat.getMin();  
stat.getStd();
```

Definition at line 235 of file eval.h.

5.18.2 Constructor & Destructor Documentation

5.18.2.1 Stat () [inline]

Definition at line 243 of file eval.h.

5.18.2.2 ~Stat () [inline]

Definition at line 244 of file eval.h.

5.18.3 Member Function Documentation

5.18.3.1 void append (float r) [inline]

Definition at line 252 of file eval.h.

Here is the caller graph for this function:



5.18.3.2 float getAvg () const [inline]

Definition at line 273 of file eval.h.

5.18.3.3 int getCount () const [inline]

Definition at line 265 of file eval.h.

5.18.3.4 `float getMax () const` `[inline]`

Definition at line 277 of file eval.h.

5.18.3.5 `float getMin () const` `[inline]`

Definition at line 281 of file eval.h.

5.18.3.6 `float getStd () const` `[inline]`

Definition at line 285 of file eval.h.

5.18.3.7 `float getSum () const` `[inline]`

Definition at line 269 of file eval.h.

5.18.3.8 `void merge (const Stat & stat)` `[inline]`

Definition at line 296 of file eval.h.

5.18.3.9 `Stat& operator<< (float r)` `[inline]`

Definition at line 260 of file eval.h.

Here is the call graph for this function:



5.18.3.10 `void reset ()` `[inline]`

Definition at line 245 of file eval.h.

5.18.4 Member Data Documentation

5.18.4.1 `int count` `[private]`

Definition at line 237 of file eval.h.

5.18.4.2 `float max` `[private]`

Definition at line 241 of file eval.h.

5.18.4.3 float min [private]

Definition at line 240 of file eval.h.

5.18.4.4 float sum [private]

Definition at line 238 of file eval.h.

5.18.4.5 float sum2 [private]

Definition at line 239 of file eval.h.

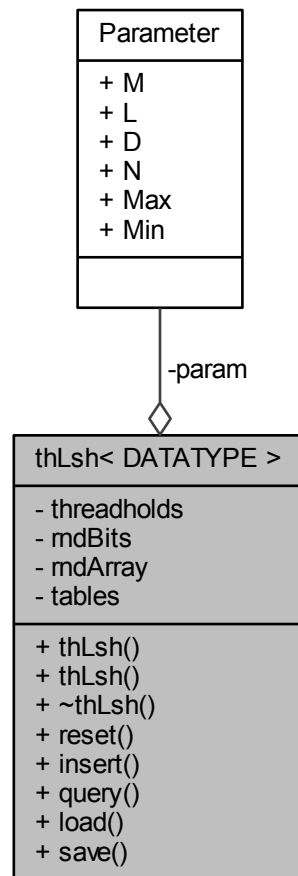
The documentation for this class was generated from the following file:

- include/lshbox/**eval.h**

5.19 thLsh< DATATYPE > Class Template Reference

```
#include <thlsh.h>
```

Collaboration diagram for thLsh< DATATYPE >:



Classes

- struct **Parameter**

Public Member Functions

- **thLsh** ()
- **thLsh** (const **Parameter** ¶m_)
- **~thLsh** ()
- void **reset** (const **Parameter** ¶m_)
- void **insert** (unsigned key, DATATYPE *domin)
- template<typename SCANNER >
void **query** (DATATYPE *domin, SCANNER &scanner)
- void **load** (const std::string &file)
- void **save** (const std::string &file)

Private Attributes

- **Parameter param**
- `std::vector< float >` **thresholds**
- `std::vector< std::vector< unsigned > >` **rndBits**
- `std::vector< std::vector< unsigned > >` **rndArray**
- `std::vector< std::map< unsigned, std::vector< unsigned > > >` **tables**

5.19.1 Detailed Description

`template<typename DATATYPE = float>class lshbox::thLsh< DATATYPE >`

Locality-Sensitive Hashing Scheme Based on Thresholding.

For more information on thresholding based LSH, see the following reference.

Zhe Wang, Wei Dong, William Josephson, Qin Lv, Moses Charikar, Kai Li. Sizing Sketches: A Rank-Based Analysis for Similarity Search. In Proceedings of the 2007 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems . San Diego, CA, USA. June 2007.

Qin Lv, Moses Charikar, Kai Li. Image Similarity Search with Compact Data Structures. In Proceedings of ACM 13th Conference on Information and Knowledge Management (CIKM), Washington D.C., USA. November 2004.

Definition at line 54 of file thlsh.h.

5.19.2 Constructor & Destructor Documentation

5.19.2.1 `thLsh()` `[inline]`

Definition at line 72 of file thlsh.h.

5.19.2.2 `thLsh(const Parameter & param_)` `[inline]`

Definition at line 73 of file thlsh.h.

Here is the call graph for this function:



5.19.2.3 `~thLsh()` `[inline]`

Definition at line 77 of file thlsh.h.

5.19.3 Member Function Documentation

5.19.3.1 `void insert (unsigned key, DATATYPE * domin) [inline]`

Insert a vector to the index.

Parameters

<i>key</i>	The sequence number of vector
<i>domin</i>	The pointer to the vector

Definition at line 121 of file thlsh.h.

5.19.3.2 `void load (const std::string & file) [inline]`

Load the index from binary file.

Parameters

<i>file</i>	The path of binary file.
-------------	--------------------------

Definition at line 173 of file thlsh.h.

5.19.3.3 `void query (DATATYPE * domin, SCANNER & scanner) [inline]`

Query the approximate nearest neighborholds.

Parameters

<i>domin</i>	The pointer to the vector
<i>scanner</i>	Top-K scanner, use for scan the approximate nearest neighborholds

Definition at line 145 of file thlsh.h.

5.19.3.4 `void reset (const Parameter & param_) [inline]`

Reset the parameter setting

Parameters

<i>param_</i>	A instance of thLsh<DATATYPE>::Parametor, which contains the necessary parameters
---------------	---

Definition at line 84 of file thlsh.h.

Here is the caller graph for this function:



5.19.3.5 `void save (const std::string & file) [inline]`

Save the index as binary file.

Parameters

<i>file</i>	The path of binary file.
-------------	--------------------------

Definition at line 212 of file thlsh.h.

5.19.4 Member Data Documentation

5.19.4.1 Parameter param [private]

Definition at line 240 of file thlsh.h.

5.19.4.2 std::vector<std::vector<unsigned>> rndArray [private]

Definition at line 243 of file thlsh.h.

5.19.4.3 std::vector<std::vector<unsigned>> rndBits [private]

Definition at line 242 of file thlsh.h.

5.19.4.4 std::vector<std::map<unsigned, std::vector<unsigned>>> tables [private]

Definition at line 244 of file thlsh.h.

5.19.4.5 std::vector<float> thresholds [private]

Definition at line 241 of file thlsh.h.

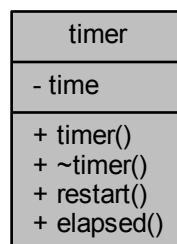
The documentation for this class was generated from the following file:

- include/lshbox/thlsh.h

5.20 timer Class Reference

```
#include <basis.h>
```

Collaboration diagram for timer:



Public Member Functions

- **timer** ()
- **~timer** ()
- void **restart** ()
- double **elapsed** ()

Private Attributes

- double **time**

5.20.1 Detailed Description

A timer object measures elapsed time, and it is very similar to `boost::timer`.

Definition at line 123 of file `basis.h`.

5.20.2 Constructor & Destructor Documentation

5.20.2.1 `timer ()` `[inline]`

Definition at line 126 of file `basis.h`.

5.20.2.2 `~timer ()` `[inline]`

Definition at line 127 of file `basis.h`.

5.20.3 Member Function Documentation

5.20.3.1 `double elapsed ()` `[inline]`

Measures elapsed time.

Returns

The elapsed time

Definition at line 140 of file `basis.h`.

5.20.3.2 `void restart ()` `[inline]`

Restart the timer.

Definition at line 131 of file `basis.h`.

5.20.4 Member Data Documentation

5.20.4.1 `double time` `[private]`

Definition at line 145 of file `basis.h`.

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

- `include/lshbox/basis.h`

5.21 Topk Class Reference

```
#include <topk.h>
```

Collaboration diagram for Topk:

Topk
<ul style="list-style-type: none"> - K - R - heapv
<ul style="list-style-type: none"> + Topk() + ~Topk() + reset() + getTopk() + getTopk() + push() + getMin() + recall()

Public Member Functions

- **Topk** ()
- **~Topk** ()
- void **reset** (unsigned k, float r=std::numeric_limits< float >::max())
- const std::vector< std::pair
 < unsigned, float > > & **getTopk** () const
- std::vector< std::pair
 < unsigned, float > > & **getTopk** ()
- void **push** (unsigned key, float dist)
- float **getMin** () const
- float **recall** (const **Topk** &topk) const

Private Attributes

- unsigned **K**
- float **R**
- std::vector< std::pair
 < unsigned, float > > **heapv**

5.21.1 Detailed Description

Top-K heap.

At this point topk should contain the nearest k query keys and distances.

Definition at line 40 of file topk.h.

5.21.2 Constructor & Destructor Documentation

5.21.2.1 `Topk ()` `[inline]`

Definition at line 47 of file topk.h.

5.21.2.2 `~Topk ()` `[inline]`

Definition at line 48 of file topk.h.

5.21.3 Member Function Documentation

5.21.3.1 `float getMin () const` `[inline]`

Get the most nearest distance in the heap.

Definition at line 100 of file topk.h.

5.21.3.2 `const std::vector<std::pair<unsigned, float> > & getTopk () const` `[inline]`

Get the `std::vector<std::pair<unsigned, float> >` instance which contains the nearest keys and distances.

Definition at line 70 of file topk.h.

Here is the caller graph for this function:



5.21.3.3 `std::vector<std::pair<unsigned, float> > & getTopk ()` `[inline]`

Get the `std::vector<std::pair<unsigned, float> >` instance which contains the nearest keys and distances.

Definition at line 78 of file topk.h.

5.21.3.4 `void push (unsigned key, float dist)` `[inline]`

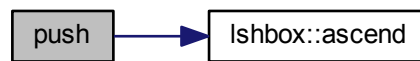
Add an element to the heap and then update it.

Parameters

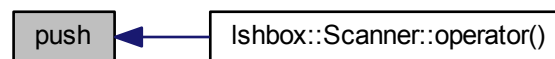
<i>key</i>	The key of the element
<i>dist</i>	The distance of the element

Definition at line 88 of file topk.h.

Here is the call graph for this function:



Here is the caller graph for this function:



5.21.3.5 `float recall (const Topk & topk) const` `[inline]`

Calculate the recall vale with another heap.

Definition at line 107 of file `topk.h`.

Here is the call graph for this function:



5.21.3.6 `void reset (unsigned k, float r = std::numeric_limits<float>::max())` `[inline]`

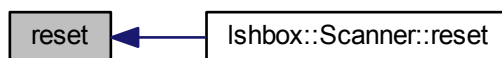
Reset the heap

Parameters

<i>k</i>	The number of nearest query keys
<i>r</i>	The initialization distance

Definition at line 55 of file `topk.h`.

Here is the caller graph for this function:



5.21.4 Member Data Documentation

5.21.4.1 `std::vector<std::pair<unsigned, float>> heapv` [private]

Definition at line 45 of file `topk.h`.

5.21.4.2 `unsigned K` [private]

Definition at line 43 of file `topk.h`.

5.21.4.3 `float R` [private]

Definition at line 44 of file `topk.h`.

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

- `include/lshbox/topk.h`

Chapter 6

File Documentation

6.1 include/lshbox.h File Reference

The LSHBOX master header file.

```
#include <lshbox/basis.h>
#include <lshbox/matrix.h>
#include <lshbox/metric.h>
#include <lshbox/topk.h>
#include <lshbox/eval.h>
#include <lshbox/rbslsh.h>
#include <lshbox/rhplsh.h>
#include <lshbox/thlsh.h>
#include <lshbox/psdlsh.h>
#include <lshbox/shlsh.h>
#include <lshbox/itqlsh.h>
```

Include dependency graph for lshbox.h:



6.1.1 Detailed Description

The LSHBOX master header file.

Copyright (C) 2014 Gefu Tang tanggefufu@gmail.com. All Rights Reserved.

This file is part of LSHBOX.

LSHBOX is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

LSHBOX is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with LSHBOX. If not, see <http://www.gnu.org/licenses/>.

Version

0.1

Author

Gefu Tang & Zhifeng Xiao

Date

2014.6.30

You only need to include this file to use the most functionalities of LSHBOX.

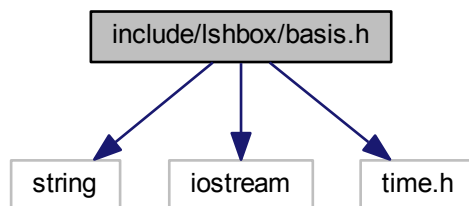
Definition in file **lshbox.h**.

6.2 include/lshbox/basis.h File Reference

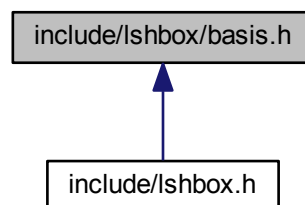
A set of basic tools.

```
#include <string>
#include <iostream>
#include <time.h>
```

Include dependency graph for basis.h:



This graph shows which files directly or indirectly include this file:

**Classes**

- class **progress_display**
- class **timer**

Namespaces

- **lshbox**

Macros

- `#define M_PI 3.14159265358979323846`

Functions

- `bool ascend (const std::pair< unsigned, float > &lhs, const std::pair< unsigned, float > &rhs)`

6.2.1 Detailed Description

A set of basic tools.

Copyright (C) 2014 Gefu Tang tanggefu@gmail.com. All Rights Reserved.

This file is part of LSHBOX.

LSHBOX is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

LSHBOX is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with LSHBOX. If not, see <http://www.gnu.org/licenses/>.

Version

0.1

Author

Gefu Tang & Zhifeng Xiao

Date

2014.6.30

Definition in file **basis.h**.

6.2.2 Macro Definition Documentation

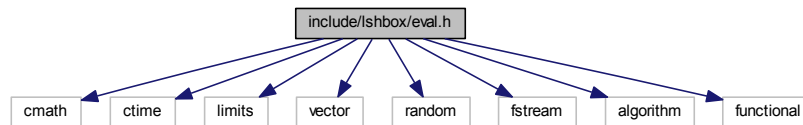
6.2.2.1 `#define M_PI 3.14159265358979323846`

Definition at line 35 of file basis.h.

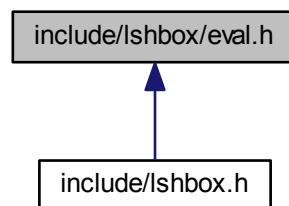
6.3 include/lshbox/eval.h File Reference

A set of classes for evaluation.

```
#include <cmath>
#include <ctime>
#include <limits>
#include <vector>
#include <random>
#include <fstream>
#include <algorithm>
#include <functional>
Include dependency graph for eval.h:
```



This graph shows which files directly or indirectly include this file:



Classes

- class **Benchmark**
- class **Stat**

Namespaces

- **lshbox**

6.3.1 Detailed Description

A set of classes for evaluation.

Copyright (C) 2014 Gefu Tang tanggifu@gmail.com. All Rights Reserved.

This file is part of LSHBOX.

LSHBOX is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

LSHBOX is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with LSHBOX. If not, see <http://www.gnu.org/licenses/>.

Version

0.1

Author

Gefu Tang & Zhifeng Xiao

Date

2014.6.30

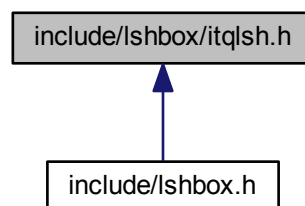
Definition in file **eval.h**.

6.4 include/lshbox/itqlsh.h File Reference

Locality-Sensitive Hashing Scheme Based on Iterative Quantization.

```
#include <map>
#include <vector>
#include <random>
#include <iostream>
#include <functional>
#include <eigen/Eigen/Dense>
```

This graph shows which files directly or indirectly include this file:



Classes

- class **itqLsh**< **DATATYPE** >
- struct **itqLsh**< **DATATYPE** >::Parameter

Namespaces

- **lshbox**

6.4.1 Detailed Description

Locality-Sensitive Hashing Scheme Based on Iterative Quantization.

Copyright (C) 2014 Gefu Tang tanggefu@gmail.com. All Rights Reserved.

This file is part of LSHBOX.

LSHBOX is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

LSHBOX is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with LSHBOX. If not, see <http://www.gnu.org/licenses/>.

Version

0.1

Author

Gefu Tang & Zhifeng Xiao

Date

2014.6.30

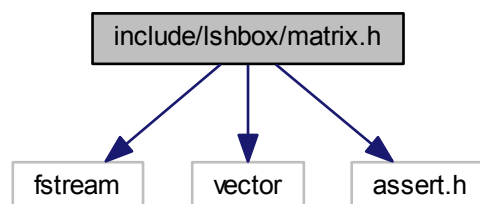
Definition in file **itqlsh.h**.

6.5 include/lshbox/matrix.h File Reference

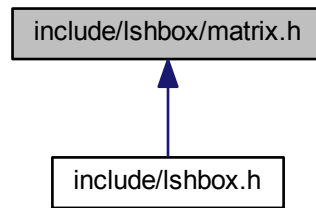
Dataset management class.

```
#include <fstream>
#include <vector>
#include <assert.h>
```

Include dependency graph for matrix.h:



This graph shows which files directly or indirectly include this file:



Classes

- class **Matrix**< T >
- class **Matrix**< T >::Accessor

Namespaces

- **lshbox**

6.5.1 Detailed Description

Dataset management class.

Copyright (C) 2014 Gefu Tang tanggefu@gmail.com. All Rights Reserved.

This file is part of LSHBOX.

LSHBOX is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or(at your option) any later version.

LSHBOX is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with LSHBOX. If not, see <http://www.gnu.org/licenses/>.

Version

0.1

Author

Gefu Tang & Zhifeng Xiao

Date

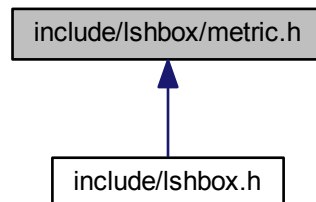
2014.6.30

Definition in file **matrix.h**.

6.6 include/lshbox/metric.h File Reference

Common distance measures.

This graph shows which files directly or indirectly include this file:



Classes

- class **Metric**< **DATATYPE** >

Namespaces

- **lshbox**

Macros

- #define **L1_DIST** 1
- #define **L2_DIST** 2

Functions

- template<typename **DATATYPE** >
DATATYPE **sqr** (const **DATATYPE** &x)

6.6.1 Detailed Description

Common distance measures.

Copyright (C) 2014 Gefu Tang tanggefu@gmail.com. All Rights Reserved.

This file is part of LSHBOX.

LSHBOX is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

LSHBOX is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with LSHBOX. If not, see <http://www.gnu.org/licenses/>.

Version

0.1

Author

Gefu Tang & Zhifeng Xiao

Date

2014.6.30

Definition in file **metric.h**.

6.6.2 Macro Definition Documentation

6.6.2.1 #define L1_DIST 1

Definition at line 32 of file metric.h.

6.6.2.2 #define L2_DIST 2

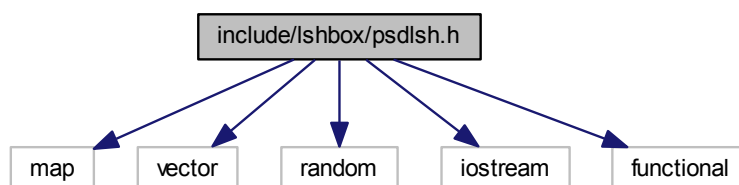
Definition at line 33 of file metric.h.

6.7 include/lshbox/psdlsh.h File Reference

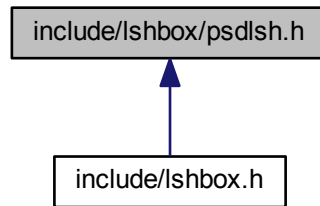
Locality-Sensitive Hashing Scheme Based on p-Stable Distributions.

```
#include <map>
#include <vector>
#include <random>
#include <iostream>
#include <functional>
```

Include dependency graph for psdlsh.h:



This graph shows which files directly or indirectly include this file:



Classes

- class **psdLsh**< **DATATYPE** >
- struct **psdLsh**< **DATATYPE** >::**Parameter**

Namespaces

- **lshbox**

Macros

- #define **CAUCHY** 1
- #define **GAUSSIAN** 2

6.7.1 Detailed Description

Locality-Sensitive Hashing Scheme Based on p-Stable Distributions.

Copyright (C) 2014 Gefu Tang tanggefu@gmail.com. All Rights Reserved.

This file is part of LSHBOX.

LSHBOX is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or(at your option) any later version.

LSHBOX is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with LSHBOX. If not, see <http://www.gnu.org/licenses/>.

Version

0.1

Author

Gefu Tang & Zhifeng Xiao

Date

2014.6.30

Definition in file **psdlsh.h**.

6.7.2 Macro Definition Documentation

6.7.2.1 #define CAUCHY 1

Definition at line 51 of file psdlsh.h.

6.7.2.2 #define GAUSSIAN 2

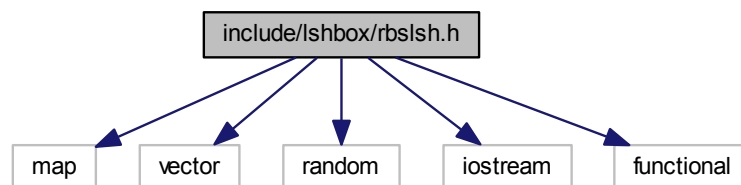
Definition at line 52 of file psdlsh.h.

6.8 include/lshbox/rbslsh.h File Reference

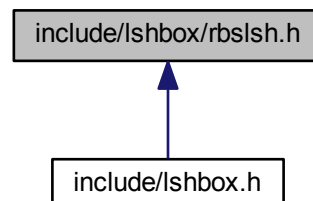
Locality-Sensitive Hashing Scheme Based on Random Bits Sampling.

```
#include <map>
#include <vector>
#include <random>
#include <iostream>
#include <functional>
```

Include dependency graph for rbslsh.h:



This graph shows which files directly or indirectly include this file:



Classes

- class **rbsLsh**
- struct **rbsLsh::Parameter**

Namespaces

- **lshbox**

6.8.1 Detailed Description

Locality-Sensitive Hashing Scheme Based on Random Bits Sampling.

Copyright (C) 2014 Gefu Tang tanggefu@gmail.com. All Rights Reserved.

This file is part of LSHBOX.

LSHBOX is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

LSHBOX is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with LSHBOX. If not, see <http://www.gnu.org/licenses/>.

Version

0.1

Author

Gefu Tang & Zhifeng Xiao

Date

2014.6.30

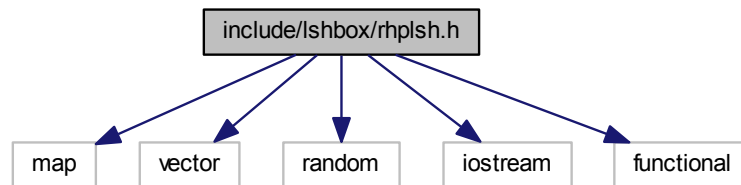
Definition in file **rbslsh.h**.

6.9 include/lshbox/rhplsh.h File Reference

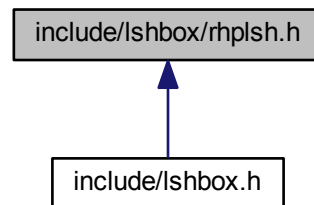
Locality-Sensitive Hashing Scheme Based on Random Hyperplane.

```
#include <map>
#include <vector>
#include <random>
#include <iostream>
#include <functional>
```

Include dependency graph for rhplsh.h:



This graph shows which files directly or indirectly include this file:



Classes

- class **rhplsh**< **DATATYPE** >
- struct **rhplsh**< **DATATYPE** >::**Parameter**

Namespaces

- **lshbox**

6.9.1 Detailed Description

Locality-Sensitive Hashing Scheme Based on Random Hyperplane.

Copyright (C) 2014 Gefu Tang tanggefu@gmail.com. All Rights Reserved.

This file is part of LSHBOX.

LSHBOX is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or(at your option) any later version.

LSHBOX is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with LSHBOX. If not, see <http://www.gnu.org/licenses/>.

Version

0.1

Author

Gefu Tang & Zhifeng Xiao

Date

2014.6.30

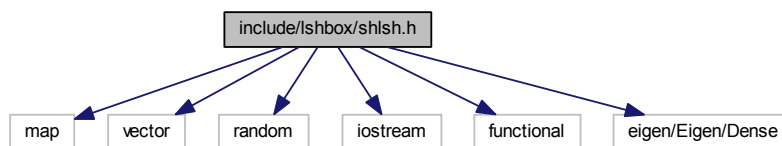
Definition in file **rhplsh.h**.

6.10 include/lshbox/shlsh.h File Reference

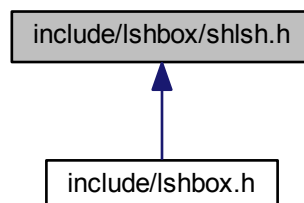
Locality-Sensitive Hashing Scheme Based on Spectral Hashing.

```
#include <map>
#include <vector>
#include <random>
#include <iostream>
#include <functional>
#include <eigen/Eigen/Dense>
```

Include dependency graph for shlsh.h:



This graph shows which files directly or indirectly include this file:

**Classes**

- class **shLsh**< **DATATYPE** >
- struct **shLsh**< **DATATYPE** >::**Parameter**

Namespaces

- **lshbox**

6.10.1 Detailed Description

Locality-Sensitive Hashing Scheme Based on Spectral Hashing.

Copyright (C) 2014 Gefu Tang tanggefu@gmail.com. All Rights Reserved.

This file is part of LSHBOX.

LSHBOX is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

LSHBOX is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with LSHBOX. If not, see <http://www.gnu.org/licenses/>.

Version

0.1

Author

Gefu Tang & Zhifeng Xiao

Date

2014.6.30

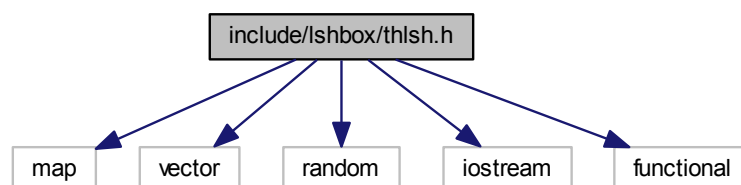
Definition in file **shlsh.h**.

6.11 include/lshbox/thlsh.h File Reference

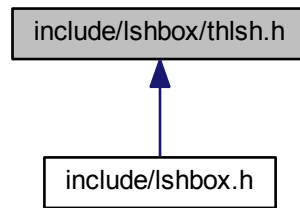
Locality-Sensitive Hashing Scheme Based on Thresholding.

```
#include <map>
#include <vector>
#include <random>
#include <iostream>
#include <functional>
```

Include dependency graph for thlsh.h:



This graph shows which files directly or indirectly include this file:



Classes

- class **thLsh**< **DATATYPE** >
- struct **thLsh**< **DATATYPE** >::**Parameter**

Namespaces

- **lshbox**

6.11.1 Detailed Description

Locality-Sensitive Hashing Scheme Based on Thresholding.

Copyright (C) 2014 Gefu Tang tanggefu@gmail.com. All Rights Reserved.

This file is part of LSHBOX.

LSHBOX is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or(at your option) any later version.

LSHBOX is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with LSHBOX. If not, see <http://www.gnu.org/licenses/>.

Version

0.1

Author

Gefu Tang & Zhifeng Xiao

Date

2014.6.30

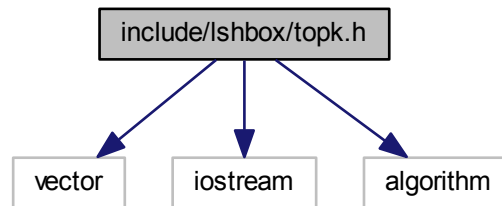
Definition in file **thlsh.h**.

6.12 include/lshbox/topk.h File Reference

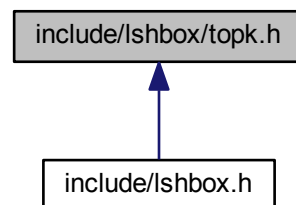
Top-K data structures.

```
#include <vector>
#include <iostream>
#include <algorithm>
```

Include dependency graph for topk.h:



This graph shows which files directly or indirectly include this file:



Classes

- class **Topk**
- class **Scanner**< **ACCESSOR** >

Namespaces

- **lshbox**

6.12.1 Detailed Description

Top-K data structures.

Copyright (C) 2014 Gefu Tang tanggefu@gmail.com. All Rights Reserved.

This file is part of LSHBOX.

LSHBOX is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

LSHBOX is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with LSHBOX. If not, see <http://www.gnu.org/licenses/>.

Version

0.1

Author

Gefu Tang & Zhifeng Xiao

Date

2014.6.30

Definition in file **topk.h**.

Index

- _count
 - lshbox::progress_display, 43
 - _expected_count
 - lshbox::progress_display, 43
 - _next_tic_count
 - lshbox::progress_display, 43
 - _tic
 - lshbox::progress_display, 43
 - ~Benchmark
 - lshbox::Benchmark, 14
 - ~Matrix
 - lshbox::Matrix, 24
 - ~Metric
 - lshbox::Metric, 29
 - ~Stat
 - lshbox::Stat, 67
 - ~Topk
 - lshbox::Topk, 76
 - ~itqLsh
 - lshbox::itqLsh, 19
 - ~psdLsh
 - lshbox::psdLsh, 46
 - ~rbsLsh
 - lshbox::rbsLsh, 49
 - ~rhpLsh
 - lshbox::rhpLsh, 53
 - ~shLsh
 - lshbox::shLsh, 61
 - ~thLsh
 - lshbox::thLsh, 71
 - ~timer
 - lshbox::timer, 74
- Accessor
 - lshbox::Matrix::Accessor, 11
- accessor_
 - lshbox::Scanner, 58
- append
 - lshbox::Stat, 67
- ascend
 - lshbox, 7
- basis.h
 - M_PI, 81
- Benchmark,
 - lshbox::Benchmark, 14
- C
 - lshbox::rbsLsh::Parameter, 37
- CAUCHY

- psdLsh.h, 89
- cnt
 - lshbox::Scanner, 57
- cnt_
 - lshbox::Scanner, 58
- count
 - lshbox::Stat, 68
 - lshbox::progress_display, 41
- D
 - lshbox::itqLsh::Parameter, 38
 - lshbox::psdLsh::Parameter, 35
 - lshbox::rbsLsh::Parameter, 37
 - lshbox::rhpLsh::Parameter, 34
 - lshbox::shLsh::Parameter, 31
 - lshbox::thLsh::Parameter, 32
- DATATYPE
 - lshbox::Matrix::Accessor, 11
 - lshbox::Scanner, 57
- dim
 - lshbox::Matrix, 28
 - lshbox::Metric, 29
- dim_
 - lshbox::Metric, 30
- dims
 - lshbox::Matrix, 28
- display_tic
 - lshbox::progress_display, 41
- dist
 - lshbox::Metric, 29
- elapsed
 - lshbox::timer, 74
- expected_count
 - lshbox::progress_display, 41
- flags_
 - lshbox::Matrix::Accessor, 12
- free
 - lshbox::Matrix, 25
- GAUSSIAN
 - psdLsh.h, 89
- getAnswer
 - lshbox::Benchmark, 14
- getAvg
 - lshbox::Stat, 67
- getCount
 - lshbox::Stat, 67
- getDim

- Ishbox::Matrix, 25
- getK
 - Ishbox::Benchmark, 14
- getMax
 - Ishbox::Stat, 67
- getMin
 - Ishbox::Stat, 68
 - Ishbox::Topk, 76
- getQ
 - Ishbox::Benchmark, 14
- getQuery
 - Ishbox::Benchmark, 14
- getSize
 - Ishbox::Matrix, 25
- getStd
 - Ishbox::Stat, 68
- getSum
 - Ishbox::Stat, 68
- getTopk
 - Ishbox::Topk, 76
- heapv
 - Ishbox::Topk, 78
- I
 - Ishbox::itqLsh::Parameter, 38
- include/Ishbox.h, 79
- include/Ishbox/basis.h, 80
- include/Ishbox/eval.h, 81
- include/Ishbox/itqLsh.h, 83
- include/Ishbox/matrix.h, 84
- include/Ishbox/metric.h, 86
- include/Ishbox/psdLsh.h, 87
- include/Ishbox/rbsLsh.h, 89
- include/Ishbox/rhpLsh.h, 90
- include/Ishbox/shLsh.h, 92
- include/Ishbox/thLsh.h, 93
- include/Ishbox/topk.h, 95
- init
 - Ishbox::Benchmark, 15
- insert
 - Ishbox::itqLsh, 20
 - Ishbox::psdLsh, 46
 - Ishbox::rbsLsh, 49
 - Ishbox::rhpLsh, 53
 - Ishbox::shLsh, 62
 - Ishbox::thLsh, 72
- itqLsh
 - Ishbox::itqLsh, 19
- itqLsh< DATATYPE >, 18
- itqLsh< DATATYPE >::Parameter, 37
- K
 - Ishbox::Topk, 78
- K_
 - Ishbox::Benchmark, 17
 - Ishbox::Scanner, 59
- Key
 - Ishbox::Matrix::Accessor, 11

- L
 - Ishbox::itqLsh::Parameter, 38
 - Ishbox::psdLsh::Parameter, 35
 - Ishbox::rbsLsh::Parameter, 37
 - Ishbox::rhpLsh::Parameter, 34
 - Ishbox::shLsh::Parameter, 31
 - Ishbox::thLsh::Parameter, 32
- L1_DIST
 - metric.h, 87
- L2_DIST
 - metric.h, 87
- load
 - Ishbox::Benchmark, 15
 - Ishbox::Matrix, 25, 26
 - Ishbox::itqLsh, 20
 - Ishbox::psdLsh, 46
 - Ishbox::rbsLsh, 50
 - Ishbox::rhpLsh, 54
 - Ishbox::shLsh, 63
 - Ishbox::thLsh, 72
- Ishbox, 7
 - ascend, 7
 - sqr, 8
- Ishbox::Benchmark
 - ~Benchmark, 14
 - Benchmark, 14
 - getAnswer, 14
 - getK, 14
 - getQ, 14
 - getQuery, 14
 - init, 15
 - K_, 17
 - load, 15
 - Q_, 17
 - queries_, 17
 - resize, 16
 - save, 16, 17
 - topks_, 17
- Ishbox::Matrix
 - ~Matrix, 24
 - dim, 28
 - dims, 28
 - free, 25
 - getDim, 25
 - getSize, 25
 - load, 25, 26
 - Matrix, 24
 - N, 28
 - operator[], 27
 - reset, 27
 - save, 27
- Ishbox::Matrix::Accessor
 - Accessor, 11
 - DATATYPE, 11
 - flags_, 12
 - Key, 11
 - mark, 11
 - matrix_, 12

- operator(), 11
- reset, 11
- Value, 11
- Ishbox::Metric
 - ~Metric, 29
 - dim, 29
 - dim_, 30
 - dist, 29
 - Metric, 29
 - type_, 30
- Ishbox::Scanner
 - accessor_, 58
 - cnt, 57
 - cnt_, 58
 - DATATYPE, 57
 - K_, 59
 - metric_, 59
 - operator(), 57
 - query_, 59
 - R_, 59
 - reset, 58
 - Scanner, 57
 - topk, 58
 - topk_, 59
 - Value, 57
- Ishbox::Stat
 - ~Stat, 67
 - append, 67
 - count, 68
 - getAvg, 67
 - getCount, 67
 - getMax, 67
 - getMin, 68
 - getStd, 68
 - getSum, 68
 - max, 68
 - merge, 68
 - min, 68
 - operator<<, 68
 - reset, 68
 - Stat, 67
 - sum, 69
 - sum2, 69
- Ishbox::Topk
 - ~Topk, 76
 - getMin, 76
 - getTopk, 76
 - heapv, 78
 - K, 78
 - push, 76
 - R, 78
 - recall, 77
 - reset, 77
 - Topk, 76
- Ishbox::itqLsh
 - ~itqLsh, 19
 - insert, 20
 - itqLsh, 19
 - load, 20
 - omegasAll, 21
 - param, 22
 - pcsAll, 22
 - query, 20
 - reset, 20
 - rndArray, 22
 - save, 21
 - tables, 22
 - train, 21
- Ishbox::itqLsh::Parameter
 - D, 38
 - I, 38
 - L, 38
 - M, 39
 - N, 39
 - S, 39
- Ishbox::progress_display
 - _count, 43
 - _expected_count, 43
 - _next_tic_count, 43
 - _tic, 43
 - count, 41
 - display_tic, 41
 - expected_count, 41
 - m_os, 43
 - m_s1, 43
 - m_s2, 43
 - m_s3, 44
 - operator++, 42
 - operator+=", 42
 - progress_display, 41
 - restart, 42
- Ishbox::psdLsh
 - ~psdLsh, 46
 - insert, 46
 - load, 46
 - param, 47
 - psdLsh, 45
 - query, 46
 - reset, 46
 - rndBs, 47
 - save, 46
 - stableArray, 47
 - tables, 47
- Ishbox::psdLsh::Parameter
 - D, 35
 - L, 35
 - M, 35
 - T, 36
 - W, 36
- Ishbox::rbsLsh
 - ~rbsLsh, 49
 - insert, 49
 - load, 50
 - param, 51
 - query, 50
 - rbsLsh, 49

- reset, 50
- rndArray, 51
- rndBits, 51
- save, 50
- tables, 51
- Ishbox::rbsLsh::Parameter
 - C, 37
 - D, 37
 - L, 37
 - M, 37
 - N, 37
- Ishbox::rhpLsh
 - ~rhpLsh, 53
 - insert, 53
 - load, 54
 - param, 55
 - query, 54
 - reset, 54
 - rhpLsh, 53
 - rndArray, 55
 - save, 54
 - tables, 55
 - uosArray, 55
- Ishbox::rhpLsh::Parameter
 - D, 34
 - L, 34
 - M, 34
 - N, 34
- Ishbox::shLsh
 - ~shLsh, 61
 - insert, 62
 - load, 63
 - minsAll, 64
 - omegasAll, 65
 - param, 65
 - pcsAll, 65
 - query, 63
 - reset, 63
 - rndArray, 65
 - save, 64
 - shLsh, 61
 - tables, 65
 - train, 64
- Ishbox::shLsh::Parameter
 - D, 31
 - L, 31
 - M, 31
 - N, 31
 - S, 31
- Ishbox::thLsh
 - ~thLsh, 71
 - insert, 72
 - load, 72
 - param, 73
 - query, 72
 - reset, 72
 - rndArray, 73
 - rndBits, 73
 - save, 72
 - tables, 73
 - thLsh, 71
 - thresholds, 73
- Ishbox::thLsh::Parameter
 - D, 32
 - L, 32
 - M, 33
 - Max, 33
 - Min, 33
 - N, 33
- Ishbox::timer
 - ~timer, 74
 - elapsed, 74
 - restart, 74
 - time, 74
 - timer, 74
- M
 - Ishbox::itqLsh::Parameter, 39
 - Ishbox::psdLsh::Parameter, 35
 - Ishbox::rbsLsh::Parameter, 37
 - Ishbox::rhpLsh::Parameter, 34
 - Ishbox::shLsh::Parameter, 31
 - Ishbox::thLsh::Parameter, 33
- M_PI
 - basis.h, 81
- m_os
 - Ishbox::progress_display, 43
- m_s1
 - Ishbox::progress_display, 43
- m_s2
 - Ishbox::progress_display, 43
- m_s3
 - Ishbox::progress_display, 44
- mark
 - Ishbox::Matrix::Accessor, 11
- Matrix
 - Ishbox::Matrix, 24
- Matrix< T >, 22
- Matrix< T >::Accessor, 9
- matrix_
 - Ishbox::Matrix::Accessor, 12
- Max
 - Ishbox::thLsh::Parameter, 33
- max
 - Ishbox::Stat, 68
- merge
 - Ishbox::Stat, 68
- Metric
 - Ishbox::Metric, 29
- Metric< DATATYPE >, 28
- metric.h
 - L1_DIST, 87
 - L2_DIST, 87
- metric_
 - Ishbox::Scanner, 59
- Min
 - Ishbox::thLsh::Parameter, 33

- min
 - lshbox::Stat, 68
- minsAll
 - lshbox::shLsh, 64
- N
 - lshbox::Matrix, 28
 - lshbox::itqLsh::Parameter, 39
 - lshbox::rbsLsh::Parameter, 37
 - lshbox::rhpLsh::Parameter, 34
 - lshbox::shLsh::Parameter, 31
 - lshbox::thLsh::Parameter, 33
- omegasAll
 - lshbox::itqLsh, 21
 - lshbox::shLsh, 65
- operator<<
 - lshbox::Stat, 68
- operator()
 - lshbox::Matrix::Accessor, 11
 - lshbox::Scanner, 57
- operator++
 - lshbox::progress_display, 42
- operator+=
 - lshbox::progress_display, 42
- operator[]
 - lshbox::Matrix, 27
- param
 - lshbox::itqLsh, 22
 - lshbox::psdLsh, 47
 - lshbox::rbsLsh, 51
 - lshbox::rhpLsh, 55
 - lshbox::shLsh, 65
 - lshbox::thLsh, 73
- pcsAll
 - lshbox::itqLsh, 22
 - lshbox::shLsh, 65
- progress_display, 39
 - lshbox::progress_display, 41
- psdLsh
 - lshbox::psdLsh, 45
- psdLsh< DATATYPE >, 44
- psdLsh< DATATYPE >::Parameter, 34
- psdLsh.h
 - CAUCHY, 89
 - GAUSSIAN, 89
- push
 - lshbox::Topk, 76
- Q_
 - lshbox::Benchmark, 17
- queries_
 - lshbox::Benchmark, 17
- query
 - lshbox::itqLsh, 20
 - lshbox::psdLsh, 46
 - lshbox::rbsLsh, 50
 - lshbox::rhpLsh, 54
- lshbox::shLsh, 63
- lshbox::thLsh, 72
- lshbox::shLsh, 63
- lshbox::thLsh, 72
- query_
 - lshbox::Scanner, 59
- R
 - lshbox::Topk, 78
- R_
 - lshbox::Scanner, 59
- rbsLsh, 47
 - lshbox::rbsLsh, 49
- rbsLsh::Parameter, 36
- recall
 - lshbox::Topk, 77
- reset
 - lshbox::Matrix, 27
 - lshbox::Matrix::Accessor, 11
 - lshbox::Scanner, 58
 - lshbox::Stat, 68
 - lshbox::Topk, 77
 - lshbox::itqLsh, 20
 - lshbox::psdLsh, 46
 - lshbox::rbsLsh, 50
 - lshbox::rhpLsh, 54
 - lshbox::shLsh, 63
 - lshbox::thLsh, 72
- resize
 - lshbox::Benchmark, 16
- restart
 - lshbox::progress_display, 42
 - lshbox::timer, 74
- rhpLsh
 - lshbox::rhpLsh, 53
- rhpLsh< DATATYPE >, 51
- rhpLsh< DATATYPE >::Parameter, 33
- rndArray
 - lshbox::itqLsh, 22
 - lshbox::rbsLsh, 51
 - lshbox::rhpLsh, 55
 - lshbox::shLsh, 65
 - lshbox::thLsh, 73
- rndBits
 - lshbox::rbsLsh, 51
 - lshbox::thLsh, 73
- rndBs
 - lshbox::psdLsh, 47
- S
 - lshbox::itqLsh::Parameter, 39
 - lshbox::shLsh::Parameter, 31
- save
 - lshbox::Benchmark, 16, 17
 - lshbox::Matrix, 27
 - lshbox::itqLsh, 21
 - lshbox::psdLsh, 46
 - lshbox::rbsLsh, 50
 - lshbox::rhpLsh, 54
 - lshbox::shLsh, 64
 - lshbox::thLsh, 72

Scanner

- Ishbox::Scanner, 57

Scanner< ACCESSOR >, 55

shLsh

- Ishbox::shLsh, 61

shLsh< DATATYPE >, 59

shLsh< DATATYPE >::Parameter, 30

sqr

- Ishbox, 8

stableArray

- Ishbox::psdLsh, 47

Stat, 65

- Ishbox::Stat, 67

sum

- Ishbox::Stat, 69

sum2

- Ishbox::Stat, 69

T

- Ishbox::psdLsh::Parameter, 36

tables

- Ishbox::itqLsh, 22

- Ishbox::psdLsh, 47

- Ishbox::rbsLsh, 51

- Ishbox::rhpLsh, 55

- Ishbox::shLsh, 65

- Ishbox::thLsh, 73

thLsh

- Ishbox::thLsh, 71

thLsh< DATATYPE >, 69

thLsh< DATATYPE >::Parameter, 32

thresholds

- Ishbox::thLsh, 73

time

- Ishbox::timer, 74

timer, 73

- Ishbox::timer, 74

Topk, 75

- Ishbox::Topk, 76

topk

- Ishbox::Scanner, 58

topk_

- Ishbox::Scanner, 59

topks_

- Ishbox::Benchmark, 17

train

- Ishbox::itqLsh, 21

- Ishbox::shLsh, 64

type_

- Ishbox::Metric, 30

uosArray

- Ishbox::rhpLsh, 55

Value

- Ishbox::Matrix::Accessor, 11

- Ishbox::Scanner, 57

W

- Ishbox::psdLsh::Parameter, 36