My Project

Generated by Doxygen 1.8.7

Mon Jul 7 2014 03:49:57

Contents

1 Class Index 1

1.1 Class List 1

2 Class Documentation 3

2.1 KOIL Class Reference 3

2.1.1 Detailed Description 4

2.1.2 Member Function Documentation 4

2.1.2.1 evaluate_AUC 4

2.1.2.2 fifo_plus 4

2.1.2.3 fifo_update_budget 4

2.1.2.4 rs_plus 5

2.1.2.5 rs_update_budget 5

2.1.2.6 update_b 5

2.1.2.7 update_kernel 6

2.1.2.8 update_kernel_l2 6

2.2 koil_result Struct Reference 7

2.2.1 Detailed Description 7

2.3 svm_model Struct Reference 7

2.3.1 Detailed Description 8

2.4 svm_node Struct Reference 8

2.4.1 Detailed Description 8

2.5 svm_parameter Struct Reference 8

2.5.1 Detailed Description 9

2.6 svm_problem Struct Reference 9

2.6.1 Detailed Description 9

2.6.2 Member Function Documentation 9

2.6.2.1 load_cross_validation 9

2.6.2.2 load_problem 10

Chapter 1

Class Index

1.1 Class List

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

KOIL
Kernelized Online Imbalanced Learning 3
koil_result
Struct: store the AUC, Accuracy, time, error number in n runs 7
svm_model
SVM decision function 7
svm_node
SVM node: <index, value> pair for each feature 8
svm_parameter
All the SVM parameters 8
svm_problem
Libsvm type dataset 9

Chapter 2

Class Documentation

2.1 KOIL Class Reference

Kernelized Online Imbalanced Learning.

`#include <KOIL.h>`

Public Member Functions

- void rs_plus (int ∗id_train, int cnt_train, int ∗id_test, int cnt_test, string losstype, svm_model &model, double &AUC, double &Accuracy, double &time, int &err_count)

  KOIL_RS++.
- void fifo_plus (int ∗id_train, int cnt_train, int ∗id_test, int cnt_test, string losstype, svm_model &model, double &AUC, double &Accuracy, double &time, int &err_count)

  KOIL_FIFO++.
- void rs_update_budget (svm_node ∗xt, double at, int max_n, int t, svm_model &model, double ∗&alpha, svm_node ∗∗&SV, int &cur_n, bool &flag, int &ridx)

  KOIL_RS++: update budget.
- void fifo_update_budget (svm_node ∗xt, double at, int max_n, svm_model &model, int &fidx, double ∗&alpha, svm_node ∗∗&SV, int &cur_n, bool &flag, int &ridx)

  KOIL_FIFO++: update budget.
- void update_b (svm_model &model)

  KOIL: update threshold of decision function.
- void update_kernel (svm_node ∗xt, double yt, svm_model &model, double &at)

  update the weight for SV
- void update_kernel_l2 (svm_node ∗xt, double yt, svm_model &model, double &at)

  update the weight for SV
- void evaluate_AUC (double ∗f, double ∗y, int n, double &AUC, double &Accuracy)

  the calculate the AUC and Accuracy between f and y

Public Attributes

- svm_problem **prob**
- svm_model **rs_model**
- svm_model **fifo_model**
- koil_result **rs_result**
- koil_result **fifo_result**
- string **save_result_path**

- string **load_data_path**
- string **dataset_file**
- string **idx_asso_file**
- string **idx_cv_file**
- string **rs_model_file**
- string **fifo_model_file**
- string **rs_result_file**
- string **fifo_result_file**
- string **log_file**

2.1.1 Detailed Description

Kernelized Online Imbalanced Learning.

2.1.2 Member Function Documentation

2.1.2.1 void KOIL::evaluate_AUC (double ∗ *f,* double ∗ *y,* int *n,* double & *AUC,* double & *Accuracy*)

the calculate the AUC and Accuracy between f and y

Parameters

f	1xn vector, the predicted label by the model
y	1xn vector, the true label
n	the number of the label

Returns

AUC AUC value
Accuracy Accuracy for the correct prediction

2.1.2.2 void KOIL::fifo_plus (int ∗ *id_train,* int *cnt_train,* int ∗ *id_test,* int *cnt_test,* string *losstype,* svm_model & *model,* double & *AUC,* double & *Accuracy,* double & *time,* int & *err_count*)

KOIL_FIFO++.

Parameters

id_train	the index of the training samples
cnt_train	the number of the training samples
id_test	the index of the testing samples
cnt_test	the number of the testing samples

Returns

model the learned decision function
AUC the AUC value on the testing samples
Accuracy the Accuracy on the testing samples
time the time used for training
err_count the number of the misclassified samples online

2.1.2.3 void KOIL::fifo_update_budget (svm_node ∗ *xt,* double *at,* int *max_n,* svm_model & *model,* int & *fidx,* double ∗& *alpha,* svm_node ∗∗& *SV,* int & *cur_n,* bool & *flag,* int & *ridx*)

KOIL_FIFO++: update budget.

Parameters

xt	xt the t-th sample xt
at	the weight of xt
max_n	the maximun number for the buffer
model	the current decision function f
fidx	the index of the first SV in the buffer (FIFO)
alpha	the weights of SVs, which have the same label with xt
SV	the SV, which have the same label with xt
cur_n	current number of SVs in the buffer
flag	indicate whether xt is put in the buffer or not
ridx	the replaced index for xt if xt is put in the buffer

2.1.2.4 void KOIL::rs_plus (int ∗ *id_train,* int *cnt_train,* int ∗ *id_test,* int *cnt_test,* string *losstype,* svm_model & *model,* double & *AUC,* double & *Accuracy,* double & *time,* int & *err_count*)

KOIL_RS++.

Parameters

id_train	the index of the training samples
cnt_train	the number of the training samples
id_test	the index of the testing samples
cnt_test	the number of the testing samples

Returns

model the learned decision function
AUC the AUC value on the testing samples
Accuracy the Accuracy on the testing samples
time the time used for training
err_count the number of the misclassified samples online

2.1.2.5 void KOIL::rs_update_budget (svm_node ∗ *xt,* double *at,* int *max_n,* int *t,* svm_model & *model,* double ∗& *alpha,* svm_node ∗∗& *SV,* int & *cur_n,* bool & *flag,* int & *ridx*)

KOIL_RS++: update budget.

Parameters

xt	xt the t-th sample xt
at	the weight of xt
max_n	the maximun number for the buffer
t	the current iteration
model	the current decision function f
alpha	the weights of SVs, which have the same label with xt
SV	the SV, which have the same label with xt
cur_n	current number of SVs in the buffer
flag	indicate whether xt is put in the buffer or not
ridx	the replaced index for xt if xt is put in the buffer

2.1.2.6 void KOIL::update_b (svm_model & *model*)

KOIL: update threshold of decision function.

Parameters

model	the current decision function

2.1.2.7 void KOIL::update_kernel (svm_node $*$ *xt,* double *yt,* svm_model & *model,* double & *at*)

update the weight for SV

Parameters

xt	the t-th sample xt
yt	the label of xt
model	the current decision function f

Returns

at return the weight of xt

Parameters

losstype	indicate l1 or l2 loss, default = "l1"
xt	the t-th sample xt
yt	the label of xt
model	the current decision function f

Returns

at return the weight of xt

2.1.2.8 void KOIL::update_kernel_l2 (svm_node $*$ *xt,* double *yt,* svm_model & *model,* double & *at*)

update the weight for SV

update the weight for SV based on smooth pairwise hinge loss

Parameters

xt	the t-th sample xt
yt	the label of xt
model	the current decision function f

Returns

at return the weight of xt

Parameters

losstype	indicate l1 or l2 loss, default = "l1"
xt	the t-th sample xt
yt	the label of xt
model	the current decision function f

Returns

at return the weight of xt

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

- KOIL.h
- KOIL.cpp

2.2 koil_result Struct Reference

struct: store the AUC, Accuracy, time, error number in n runs

```
#include <KOIL.h>
```

Public Member Functions

- void **initial_result** (int n)
- void **free_result** ()
- void **save_result** (string path, string method)
- void **load_result** (string path)

Public Attributes

- int **runs**
- double ∗ **auc**
- double ∗ **accuracy**
- double ∗ **time**
- int ∗ **err_cnt**

2.2.1 Detailed Description

struct: store the AUC, Accuracy, time, error number in n runs

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

- KOIL.h
- KOIL.cpp

2.3 svm_model Struct Reference

SVM decision function.

```
#include <svm.h>
```

Public Member Functions

- void **initialize** (int budget_size)
- void **free_model** ()
- int **load_model** (string model_file_name)
- int **save_model** (string model_file_name)
- double **predict** (svm_node ∗xt)
- double ∗ **predict_list** (svm_node ∗∗xt, int n)
- double **kernel_func** (svm_node ∗x1, svm_node ∗x2)

Public Attributes

- struct svm_parameter **param**
- int **l**
- double **b**
- int **k_num**

- int **fpidx**
- int **fnidx**
- struct svm_node ∗∗ **pos_SV**
- double ∗ **pos_alpha**
- int **pos_n**
- int **max_pos_n**
- struct svm_node ∗∗ **neg_SV**
- double ∗ **neg_alpha**
- int **neg_n**
- int **max_neg_n**

2.3.1 Detailed Description

SVM decision function.

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

- svm.h
- svm.cpp

2.4 svm_node Struct Reference

SVM node: <index, value> pair for each feature.

```
#include <svm.h>
```

Public Attributes

- int **index**
- double **value**

2.4.1 Detailed Description

SVM node: <index, value> pair for each feature.

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

- svm.h

2.5 svm_parameter Struct Reference

contains all the SVM parameters

```
#include <svm.h>
```

Public Attributes

- double **C**
- int **kernel_type**
- int **degree**
- double **gamma**
- double **eta**

2.5.1 Detailed Description

contains all the SVM parameters

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

- svm.h

2.6 svm_problem Struct Reference

libsvm type dataset

```
#include <svm.h>
```

Public Member Functions

- void load_problem (string filename)

  load samples x and label y
- void load_cross_validation (string assofile, string cvfile)

  load the cross validation information

Public Attributes

- unsigned int **n**
- unsigned int **pos**
- unsigned int **neg**
- unsigned int **d**
- unsigned int **elements**
- struct svm_node ∗ **x_space**
- struct svm_node ∗∗ **x**
- double ∗ **y**
- int ∗∗ **idx_cv**
- int **n_cv**
- int **d_cv**
- int ∗∗ **idx_Asso**
- int **n_Asso**
- int **d_Asso**

2.6.1 Detailed Description

libsvm type dataset

2.6.2 Member Function Documentation

2.6.2.1 void svm_problem::load_cross_validation (string *assofile*, string *cvfile*)

load the cross validation information

Parameters

assofile	load the associated file
cvfile	load the cross validation file

2.6.2.2 void svm_problem::load_problem (string *filename*)

load samples x and label y

Parameters

filename	file name

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

- svm.h
- svm.cpp