Guaranteed probability of correct acceptance/rejection

Contents

[1 Notation and short description 2](#_Toc162979819)

[2 Corrector estimation model 2](#_Toc162979820)

[2.1 General estimation 5](#_Toc162979821)

[2.2 Detailed estimation 5](#_Toc162979822)

[3 Technical description 5](#_Toc162979823)

[3.1 Function modelCreator 5](#_Toc162979824)

[3.2 Function estimate 7](#_Toc162979825)

[3.3 Class rejectAcceptModel 9](#_Toc162979826)

[3.3.1 Constructor 9](#_Toc162979827)

[3.3.2 Method generalEstimate 10](#_Toc162979828)

[3.3.3 Method detailEstimate 10](#_Toc162979829)

[3.4 Function specDir 11](#_Toc162979830)

[3.5 Function fisher 12](#_Toc162979831)

[3.6 Function oneDClass 13](#_Toc162979832)

[References 14](#_Toc162979833)

# Notation and short description

This documentation follows paper [1] in ideas and most part of notation.

We consider problem of classification in space where is input space following to some unknown distribution and is set of possible labels. For convenience we add one more class 0 which is not class at all and corresponds to unknown answer or to absence of decision.

System can be used for arbitrary classifier . Prediction of classifier we will denote .

Our corrector can be based on original features () or on some other vector, calculated from original attributes by some map .

Input for our corrector estimation system is set of points where is set of attributes calculated from original attributes, is pure label (used to form corrector estimation model only) and is predicted class. Output can depend on request. It can be estimation of probabilities of correct acceptance and correct rejection for specified thresholds of acceptance and rejection. It also can be estimation of probabilities of correct acceptance and correct rejection for thresholds defined by query point.

Function defined as

Test dataset we denote

For each class we defined direction and score (dot product).

For each class we defined two subsets – subset of test set correctly recognised as class – and – subset of test set wrongly recognised as class .

For each class we defined two cumulative distribution functions:

For each class we define which is used to define threshold of rejection:

where is pseudo inversion:

Rejection is used if score of case is less than threshold .

For all tests and illustration, the QSAR database is used [2, 3].

# Corrector estimation model

Let us consider one class . For specified threshold of rejection , sets of points and we can estimate probabilities

1. Of correct accepting of classifier prediction as
2. Of correct rejecting of classifier prediction as

where

and

Let us simplify function definitions:

is the same as

In this function supremum is different from maximum only if tends to zero or .

Let us denote

Let us consider these two points and internal extremum.

If then , and we have

If then , and we have .

Technical comment to use it further is

To check internal points, we should calculate derivative and equal it to zero:

Unfortunately, there is no analytic solution of this equation. To use quick numerical search, it is good idea to check concavity of function . Let us calculate the second derivative of :

The first multiplier is positive: This means that sign of the second derivative can be arbitrary from positive when to negative when  
 There is no regular method to calculate extreme value of . From the other side, direct calculation of this function shows shapes, presented in Figure 1.

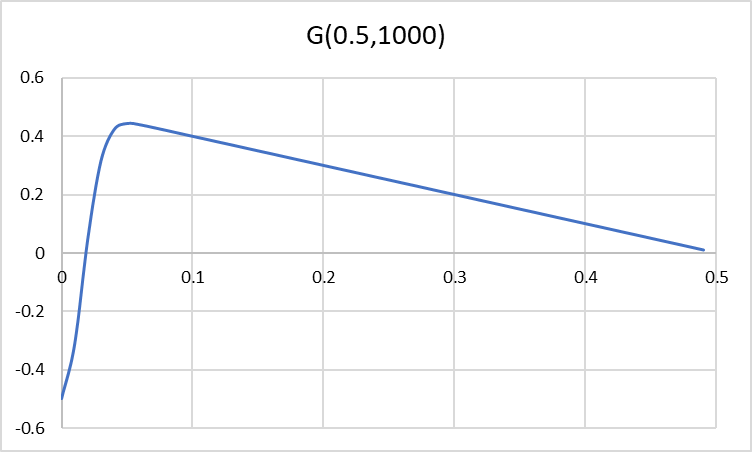
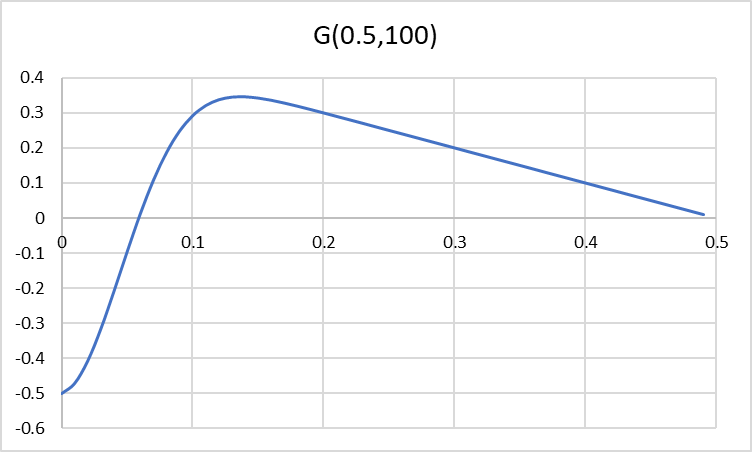
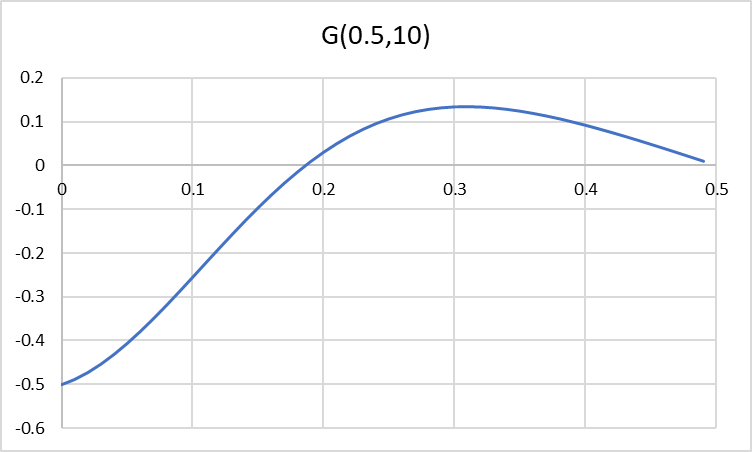
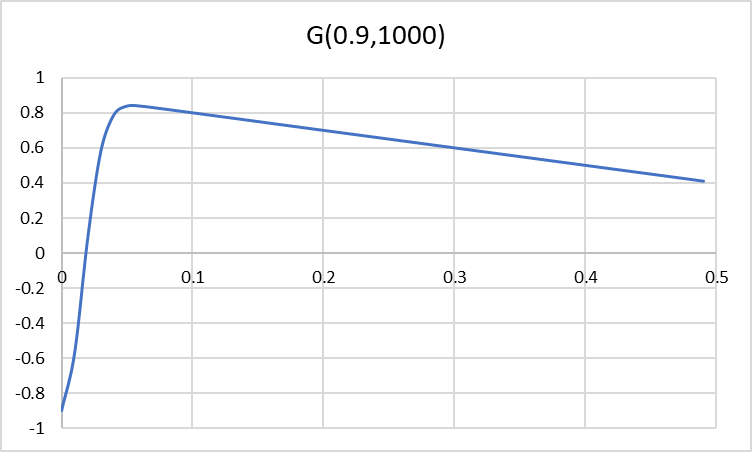
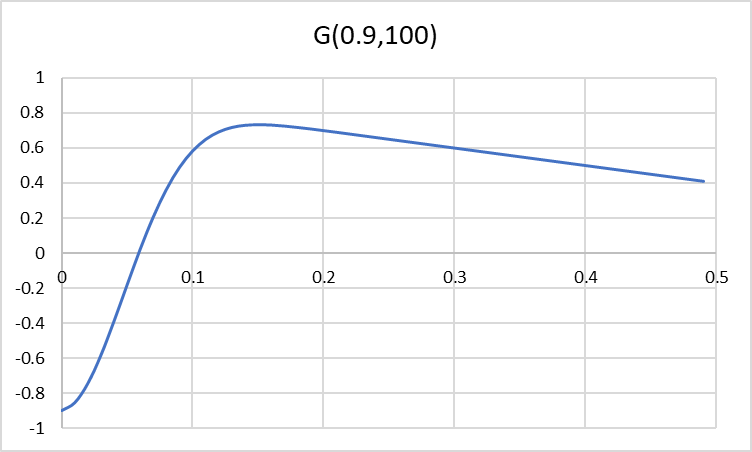
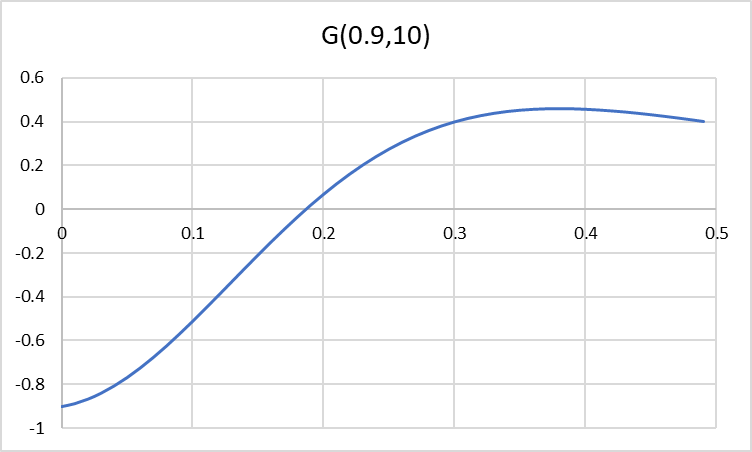


Figure . Shapes of for different values of attributes

As we can see, function has shape, which is appropriate for many search procedures. The only question is effectiveness of such search. We compared two types of searches to identify the most appropriate algorithm:

1. Grid search – calculation of function in 1000 with selection of maximal value.
2. Standard Matlab function fminbnd.

Time was estimated for six set of parameters used in Figure 1. Results are presented in Table 1. As we can see, grid search is 2.5 time faster. Further in Matlab grid search is used.

Table . Time efficiency for 100,000 searches of maximum of function

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 0.9,10 | 0.9,100 | 0.9,1000 | 0.5,10 | 0.5,100 | 0.5,1000 |
| Grid | 0.85827 | 0.85690 | 0.82997 | 0.82797 | 0.86025 | 0.85898 |
| fminbnd | 2.14020 | 2.13291 | 2.15389 | 2.10431 | 2.11753 | 2.13500 |
| Fraction | 2.49 | 2.49 | 2.60 | 2.54 | 2.46 | 2.49 |

For function

we can conclude that for we have and value of function is greater than 1. This is meaningless because we search probability. This means that we can search infimum in interval . Infimum become minimum if border points are not extremum ones.

Let us denote

If then , and we have

If then , and we have .

Graphs of are presented in Figure 2. We can see that to have probability boundary less than 100% it is necessary to have or very small border of false positive rejection or big enough number of cases.

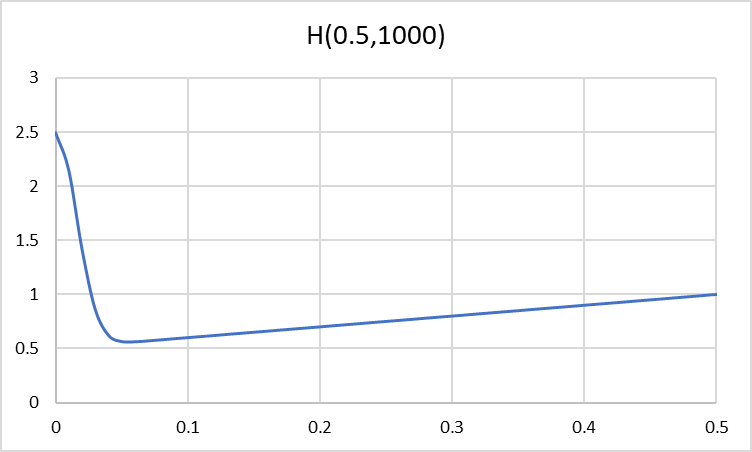
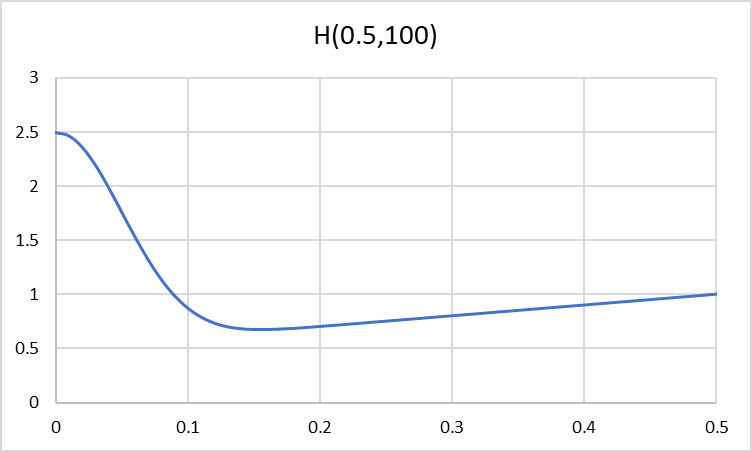
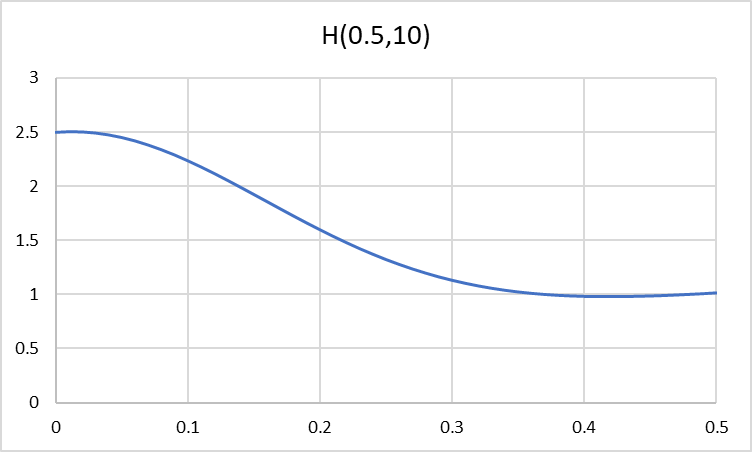
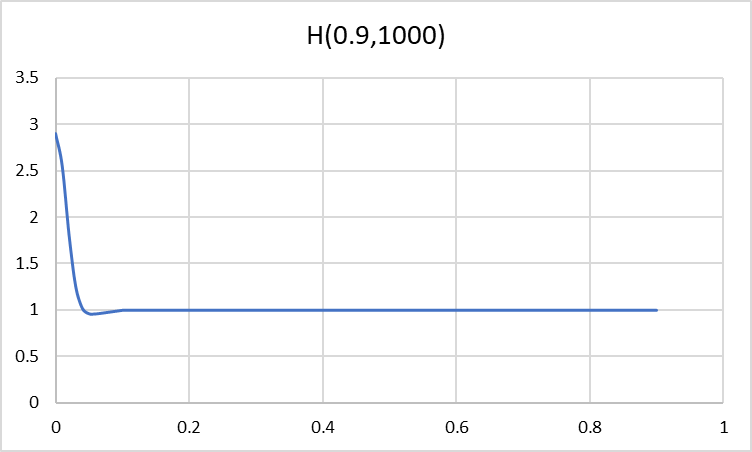
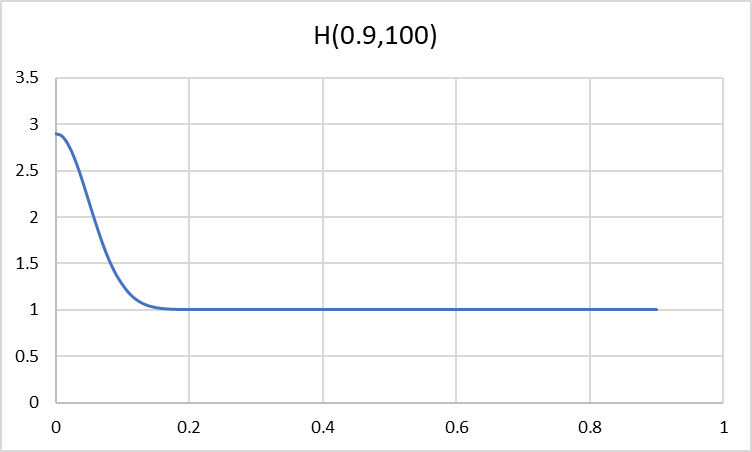
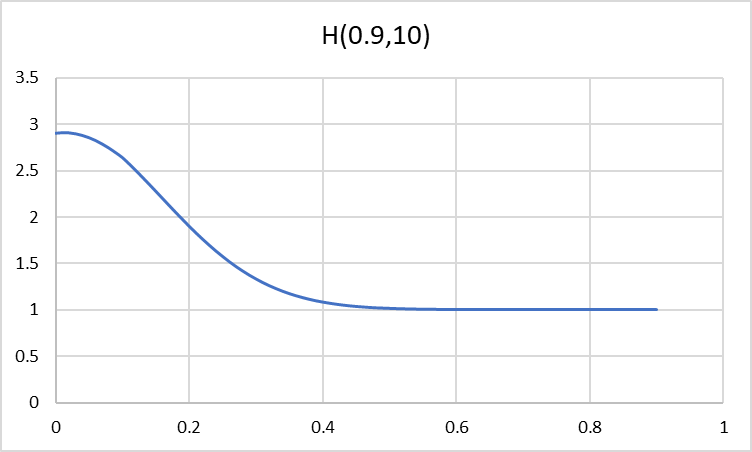


Figure . Shapes of for different values of attributes

For python I compared time consumption for rho function and found that for 10,000 calculations of time of grid search is 0.17806s and for scipy.optimize.fminbound time is 0.88591. This means that grid search in this case is 4.975 times faster. I preserve usage of grid search.

## General estimation

According to [1] there are two possible tolerance measures:

* is rejection tolerance – fraction of rejected false positives.
* is acceptance tolerance – fraction of rejected true positives.

In fact, and is used to calculate threshold for corrector rejection rule: if score of tested case is less then then decision of classifier should be rejected. Formally, both tolerance measures can be defined independently, but it seems very reasonable to use equal thresholds for acceptance and rejection. According to [1] formal rule is

|  |  |
| --- | --- |
|  | (1) |

This means that corrector will accept classifier decision for score , calculated to input . The only question is how to calculate threshold . There are two formulae:

|  |  |
| --- | --- |
|  | (2) |
|  | (3) |

It seems very reasonable to use the same value of threshold

|  |  |
| --- | --- |
|  | (4) |

Presented version of software used vector as input, calculated threshold by (2) and then calculated .

## Detailed estimation

We should decide reject or accept for each input . It will be very useful to know the probability of correctness of our decision for each input individually. It is absolutely clear, that for some cases general estimation will be more or less accurate but for cases with score far from threshold this is inacceptable. From the other side, we know score of case under consideration and we can interpret this that we used rule (1) for threshold equal to this value. In this case we can calculate and estimate probability of correct rejection/acceptance for this value. In this case we can have different probabilities for each individual point.

# Technical description

Proposed software includes several functions to process data. The main functions are “Function modelCreator” and “Function estimate”. Detailed description is presented in the next section.

Function fisher produced modelling of Fisher discriminant for each class and estimated parameters of corrector. Detailed description is presented in “Function fisher”.

Function oneDClass calculated optimal threshold for presented data and can be used separately from all other functions for estimation of any one dimensional classifiers. Detailed description is presented in “Function oneDClass”.

Python implementation is object-oriented. Class rejectAcceptModel implemented Matlab functionality described in “Function modelCreator” and “Function estimate”. This class is described in additional subsection “”

## Function modelCreator

I think that the next version must be an object implementation but this version based on usual functions. Model creation used dataset to form sets and for all classes . Model also based on 1D projections for each class. Detailed description of all features is presented below.

Inputs

x is N-by-M matrix which contains test set points. One row contains one

observation.

This matrix corresponds to dataset in paper notation.

labels is N-by-1 vector with true labels of cases in x. Classes must be

integer numbers 1, 2, ..., C, where C is the number of classes. All

classes must be presented and for each class must be at least one

case correctly predicted and at least one case wrongly predicted.

This vector contains predefined (observed, true, pure) labels.

prediction is N-by-1 vector with labels of cases in x, predicted by

corrected classifier. Classes must be integer numbers 1, 2, ..., C,

where C is the number of classes. All classes must be presented and

for each class must be at least one case correctly predicted and at

least one case wrongly predicted.

This vector contains results of prediction of classifier under correction.

Name, Value pairs can include following elements:

'name' means list of labels for figures. Default value "None" and

assumed absence of figures. Value can be single string:

'None' for omitting figures.

'Auto' for automatic generation of names.

Value also can be array of three strings or cell array of three

strings. In this case meaning of strings is

name(1) is name of attribute (title of histogram). It will be

used in title of the x-axis and in title of figure. In

title of figure to name(1) will be added fragment "for

class N" where N will be number of class under

consideration.

name(2) is name of the first class (the first element of

legend)

name(3) is name of the second class (the second element of

legend)

Attribute name serves for two purposed. If it is empty (or ‘None’) then graphs will not be formed. If name is specified the elements of this array are used for figure decoration as described in “Function oneDClass”.

'Acc' or 'Accuracy' has Value equal to string with name of one of

proposed measure of quality to select optimal threshold:

'BA' means balanced accuracy:

(TP / (TP + FN) + NT / (TN + FP)) / 2

'accuracy' means (TP + TN) / (TP + FN + TN + FP)

'f1' means F1 score 2 \* TP / (2 \* TP + FN + FP)

'NPV' means Negative predictive value TN / (TN + FN)

'PPV' means Positive predictive value TP / (TP + FP)

'TPR', 'recall', 'sens', 'power' means true positive rate

(recall, probability of detection, hit rate, power)

TP / (TP + FN)

'TNR', 'spec', 'sel', means true negative rate (specificity,

selectivity) TN / (TN + FP)

Alternatively, Value can be function handle. Function must have

following syntaxis:

function acc = funcName(TP, FP, TN, FN)

where TP means true positive, FP means false positive,

TN means true negative, FN means false negative.

Attribute “accuracy” serves to specify classifier quality measure to find optimal threshold in “Function oneDClass”. Six widely used quality measures is implemented and can be specified by name. User defined function can be specified if necessary.

'Dir' or 'Directions has value of string or matrix with directions

defined for classes. For string only value 'Fisher' is

acceptable. If set of directions is defined than it must be

matrix M-by-C (number of rows corresponds to dimension of data

space and number of columns corresponds to number of classes).

Each column in this matrix corresponds to vector to project sets

S-k and S+k to calculate scores, CDF and probabilities.

This attribute specified directions for 1D projections for each class. Fisher’s discriminant direction can be requested by string “Fisher”. User specified directions can be specified in form of numeric matrix.

The output of modelCreator function is structure (will be class in the next version) with following fields

sp is cell array with array of scores for set sp for each class

Each set also ordered. These scores can be used to estimate CDF for probabilities estimation.

sm is cell array with array of scores for set sm for each class

Each set also ordered. These scores can be used to estimate CDF for probabilities estimation.

thresholds is array with thresholds optimal for each class

Estimation of these thresholds performed by quality measure defined by argument “Accuracy”. These thresholds can be used to calculate “Natural” value of delta.

direct is M-by-C matrix with used direction for each class in

corresponding column. If vectors were specified by user, they

will be rescaled to unit length and can be inverted to provide

low values for rejection.

This matrix contains in columns direction vectors to project data (one column for one class). These vectors can be calculated through Fisher’s discriminant or from user defined directions. In any case each vector has unit length and projection of data on this vector produces smaller value (score) for false positive points (points which should be rejected).

error is array with error estimated for each class for optimal

threshold.

This array contains errors (type of measure is specified in argument “Accuracy”) which corresponds to optimal threshold.

## Function estimate

This function is using the previously created model to estimate probabilities to correctly reject classifier’s false positive and probability to accept classifier’s true positive. Both probabilities are estimated from both sides.

This estimation can be produced for arbitrary specified rejection threshold , for rejection threshold corresponding to optimal thresholds, defined during model creation, or individually for each point. The last can be used for estimation of probability that prediction is correct.

This function also can define acceptance/rejection for each element of sent dataset.

Argument mdl contains model created by Function modelCreator.

mdl is model created by modelCreator function. This model contains

following fields

sp is cell array with array of scores for set sp for each class

sm is cell array with array of scores for set sm for each class

thresholds is array with thresholds optimal for each class

direct is M-by-C matrix with used direction for each class in

corresponding column. If vectors were specified by user, they

will be rescaled to unit length and can be inverted to provide

low values for rejection.

error is array with error estimated for each class for optimal

threshold.

To specify what to calculate for general outcome the arguments delta is used.

delta specifies rejection threshold for all classes and can has

following values

'Auto' for optimal threshold defined during model creation

Real number between 0 and 1 to use the same threshold for all

classes. In this case for each class threshold defined by

finding of threshold which corresponds to cumulative function

equal to delta.

C-by-1 vector contains specified values of delta for each class.

To request detailed prediction for additional dataset with known prediction the arguments data and predicted are used.

data (optional) is N-by-M matrix with data to test. M must be equal to

number of rows in matrix direct in model mdl.

predicted (optional, must be presented together with data) is N-by-1

vector of predicted by classifier labels for records in data.

Number of elements in predicted must be the same as number of rows

in matrix data.

The output of function depends on set of specified inputs. Table general will be calculated anyway. Table detailed will be calculated if specified matrix data and vector predicted. Structures on tables general and detailed are the same but meaning of columns is slightly different.

general is table which calculated for anytime and presented general

information for specified delta and mdl. general is table with

following columns:

Class is column with class number

Threshold is used threshold for rejection (if score is less than

threshold, then prediction of classifier is rejected)

Delta is fraction of false positives to reject (corresponds to

Threshold).

Delta\_A is fraction of true positives to accept (corresponds to

Threshold).

LowBoundReject is lower boundary of probability of correct rejection

(equation (8) in journal paper).

UpBoundReject is upper boundary of probability of correct rejection

(equation (8) in journal paper).

LowBoundAccept is lower boundary of probability of correct

acceptance (equation (7) in journal paper).

UpBoundAccept is upper boundary of probability of correct

acceptance (equation (7) in journal paper).

detailed is table with individual output for each point in data. Table

contains one row for each row in data. Structure of table is the

same as for general but with slightly different meanings of columns:

Class is column with class number predicted by classifier (from

vector predicted)

Threshold is used threshold for rejection (if score is less than

threshold, then prediction of classifier is rejected).

Threshold is equal to score calculated for current point.

Delta is fraction of false positives with smaller score

Delta\_A is fraction of true positives with smaller score

LowBoundReject is lower boundary of probability of correct rejection

(equation (8) in journal paper).

UpBoundReject is upper boundary of probability of correct rejection

(equation (8) in journal paper).

LowBoundAccept is lower boundary of probability of correct

acceptance (equation (7) in journal paper).

UpBoundAccept is upper boundary of probability of correct

acceptance (equation (7) in journal paper).

In QSAR dataset used for model testing we have sizes of sets presented in Table 2. As we can see sizes of sets are very small and we cannot expect thin interval of probabilities. Really, results of general estimation, presented in Table 3, demonstrated very wide intervals, or, really, absence of certainty in decisions. This means that for small datasets there is no any methods to guaranty correctness of correctors and, really, classifiers. For individual points intervals can be essentially better if these points locate far from optimal border.

Table . Sizes of sets for QSAR dataset

|  |  |  |
| --- | --- | --- |
| Class (k) |  |  |
| 1 | 9 | 62 |
| 2 | 9 | 27 |

Table . Estimated probabilities for QSAR dataset

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Class | Threshold | Delta | Delta\_a | LowBoundReject | UpBoundReject | LowBoundAccept | UpBoundAccept |
| 1 | 6.45562 | 1.00000 | 0.11290 | 0.53287 | 1.00000 | 0.67441 | 0.99372 |
| 2 | 0.99129 | 1.00000 | 0.03704 | 0.53287 | 1.00000 | 0.65427 | 1.00000 |

## Class rejectAcceptModel

This class create model, fit parameters and estimate probabilities. Really this object implemented all functionality, described in “Function modelCreator” and “Function estimate”.

Attributes of class and their initial values:

sp = None # is list with array of scores for set sp for each class

sm = None # is list with array of scores for set sm for each class

thresholds = None # is array with thresholds optimal for each class

direct = None # is M-by-C matrix with used direction for each class in

# corresponding column. If vectors were specified by user, they

# will be rescaled to unit length and can be inverted to provide

# low values for rejection.

error = None # is array with error estimated for each class for optimal

# threshold.

### Constructor

Constructor of class have following attributes (the same as Function modelCreator in Matlab version):

x : 2D ndarray

DESCRIPTION. N-by-M matrix which contains test set points. One row

contains one observation

labels : 1D ndarray, list or tuple

DESCRIPTION. N-by-1 vector with true labels of cases in x. Classes

must be integer numbers 1, 2, ..., C, where C is the number of

classes. All classes must be presented and for each class must

be at least one case correctly predicted and at least one case

wrongly predicted.

prediction : 1D ndarray, list or tuple

DESCRIPTION. is N-by-1 vector with labels of cases in x, predicted

by corrected classifier. Classes must be integer numbers

1, 2, ..., C, where C is the number of classes. All classes must

be presented and for each class must be at least one case correctly

predicted and at least one case wrongly predicted.

name : None, string or array of strings, optional

DESCRIPTION. Possible values and their meaning:

None for omitting figures.

'Auto' for automatic generation of names.

Name also can be array of three strings. In this case meaning of

strings are

name(1) is name of attribute (title of histogram). It will be

used in title of the x-axis and in title of figure. In

title of figure to name(1) will be added fragment "for

class N" where N will be number of class under

consideration.

name(2) is name of the first class (the first element of legend)

name(3) is name of the second class (the second element of legend)

The default is None and assumed absence of figures.

acc : defined in function oneDClass, optional

DESCRIPTION. defined in function oneDClass. The default is 'ba'.

direct : string or M-by-C matrix (2D ndarray), optional

DESCRIPTION. For string only value 'Fisher' is acceptable.

If set of directions is defined than it must be matrix M-by-C

(number of rows corresponds to dimension of data space and number

of columns corresponds to number of classes). Each column in this

matrix corresponds to vector to project sets S-k and S+k to

calculate scores, CDF and probabilities

The default is 'Fisher'.

The only output of constructor is created object of class rejectAcceptModel.

### Method generalEstimate

This method produced the same calculation as the first part of Function estimate.

Argument self contains object of class rejectAcceptModel.

self is object of class rejectAcceptModel

To specify what to calculate for general outcome the arguments delta is used.

delta specifies rejection threshold for all classes and can has

following values

'Auto' for optimal threshold defined during model creation

Real number between 0 and 1 to use the same threshold for all

classes. In this case for each class threshold defined by

finding of threshold which corresponds to cumulative function

equal to delta.

C-by-1 vector contains specified values of delta for each class.

This method returns one 2D array with one row for each class and following meaning of columns:

0. Class is column with class number from 1 to C

1. Threshold is used threshold for rejection (if score is less than

threshold, then prediction of classifier is rejected)

2. Delta is fraction of false positives to reject (corresponds to

Threshold).

3. DeltaA is fraction of true positives to accept (corresponds to

Threshold).

4. LowBoundReject is lower boundary of probability of correct

rejection (equation (8) in journal paper).

5. UpBoundReject is upper boundary of probability of correct

rejection (equation (8) in journal paper).

6. LowBoundAccept is lower boundary of probability of correct

acceptance (equation (7) in journal paper).

7. UpBoundAccept is upper boundary of probability of correct

acceptance (equation (7) in journal paper).

### Method detailEstimate

This method produced the same calculation as the second part of Function estimate – estimation of probabilities for each point also described in “Detailed estimation”.

Argument self contains object of class rejectAcceptModel.

self is object of class rejectAcceptModel

Method estimates probabilities for arbitrary set of points. To specify data to test two arguments should be specified: data and predicted:

data : 2D ndarray of float

DESCRIPTION. data is N-by-M matrix with data to test. M must be

equal to number of rows in matrix direct in object self of class

rejectAcceptModel.

predicted : 1D array or array like type

DESCRIPTION. predicted is N element vector of predicted by

classifier labels for records in data. Number of elements in

predicted must be the same as number of rows in matrix data.

This method returns one 2D array with number of rows equal to number of rows in data and following meaning of columns:

Class is column with class number predicted by classifier (from

vector predicted)

Threshold is used threshold for rejection (if score is less than

threshold, then prediction of classifier is rejected).

Threshold is equal to score calculated for current point.

Delta is fraction of false positives with smaller score

Delta\_A is fraction of true positives with smaller score

LowBoundReject is lower boundary of probability of correct

rejection (equation (8) in journal paper).

UpBoundReject is upper boundary of probability of correct rejection

(equation (8) in journal paper).

LowBoundAccept is lower boundary of probability of correct

acceptance (equation (7) in journal paper).

UpBoundAccept is upper boundary of probability of correct

acceptance (equation (7) in journal paper).

## Function specDir

This function used vector, specified by user, to project all data points onto this direction, select the optimal threshold. Specified direction will be normalised to unit length and can be inversed to provide mean projection of class 1 is less than mean projection of class 2.

Formal description of arguments is presented below. Matlab documentation is presented but python description is the same exclude several minor language specific details.

Inputs

x is matrix which contains points for Class 1

y is matrix which contains points for Class 2

dir is direction to project data

name (optional) is array of strings or cell array of strings:

name(1) is name of attribute (title of histogram)

name(2) is name of the first class (the first element of legend)

name(3) is name of the second class (the second element of legend)

if name is omitted or specified as [] then graphs will not be formed.

acc (optional) is string or handle of accuracy measure function. For

string there are several appropriate values:

'BA' means balanced accuracy: (TP / (TP + FN) + NT / (TN + FP)) / 2

'accuracy' means (TP + TN) / (TP + FN + TN + FP)

'f1' means F1 score 2 \* TP / (2 \* TP + FN + FP)

'NPV' means Negative predictive value TN / (TN + FN)

'PPV' means Positive predictive value TP / (TP + FP)

'TPR', 'recall', 'sens', 'power' means true positive rate (recall,

probability of detection, hit rate, power) TP / (TP + FN)

'TNR', 'spec', 'sel', means true negative rate (specificity,

selectivity) TN / (TN + FP)

For function handle default value is @balancedAccuracy. function

must have following syntaxis:

function acc = funcName(TP, FP, TN, FN)

where TP means true positive, FP means false positive, TN means

true negative, FN means false negative.

Outputs

bestT is optimal threshold

bestErr is minimal error which corresponds to threshold bestT. Error is

one minus accuracy defined by acc

dir is vector with used direction

Results of specDir usage for 5 randomly generated vectors are presented in Figure 2.

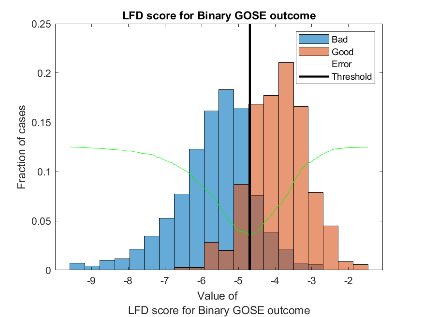
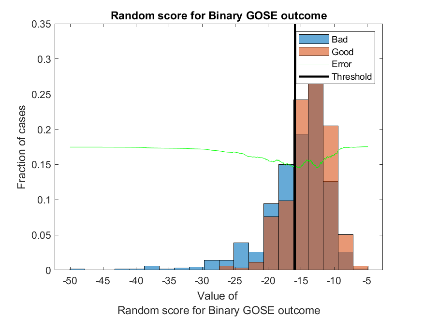
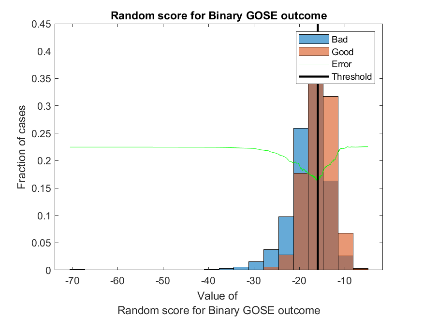
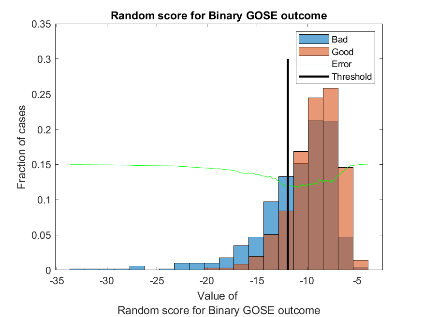
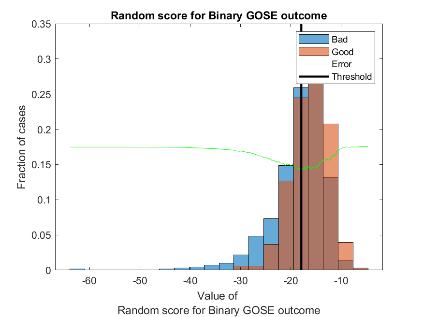
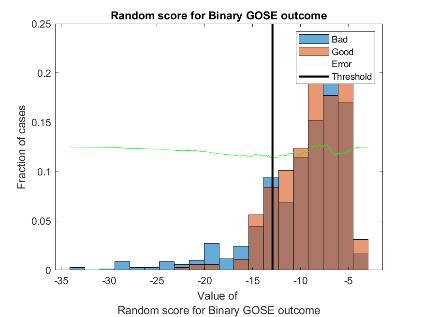
abc  
def

Figure . Results of Fisher’s discriminant (a) and five randomly generated directions, balanced accuracy was used for all figures

## Function fisher

Calculate Fisher's discriminant direction, project all data points onto this direction, select the optimal threshold. It is necessary to take into account that applied version of Fisher’s discriminant does not assume normality of distribution and also does not assume equality of covariance matrices of two classes. Formal description of arguments is presented below. Matlab documentation is presented but python description is the same exclude several minor language specific details.

Inputs

x is matrix which contains points for Class 1

y is matrix which contains points for Class 2

name (optional) is array of strings or cell array of strings:

name(1) is name of attribute (title of histogram)

name(2) is name of the first class (the first element of legend)

name(3) is name of the second class (the second element of legend)

if name is omitted or specified as [] then graphs will not be formed.

acc (optional) is string or handle of accuracy measure function. For

string there are several appropriate values:

'BA' means balanced accuracy: (TP / (TP + FN) + NT / (TN + FP)) / 2

'accuracy' means (TP + TN) / (TP + FN + TN + FP)

'f1' means F1 score 2 \* TP / (2 \* TP + FN + FP)

'NPV' means Negative predictive value TN / (TN + FN)

'PPV' means Positive predictive value TP / (TP + FP)

'TPR', 'recall', 'sens', 'power' means true positive rate (recall,

probability of detection, hit rate, power) TP / (TP + FN)

'TNR', 'spec', 'sel', means true negative rate (specificity,

selectivity) TN / (TN + FP)

For function handle default value is @balancedAccuracy. function

must have following syntaxis:

function acc = funcName(TP, FP, TN, FN)

where TP means true positive, FP means false positive, TN means

true negative, FN means false negative.

Outputs

bestT is optimal threshold

bestErr is minimal error which corresponds to threshold bestT. Error is

one minus accuracy defined by acc

dir is vector with fisher direction

## Function oneDClass

This function is presented in file oneDClass for Matlab version and is part of guarantee library in python version. Formal description of arguments is presented below. Matlab documentation is presented but python description is the same exclude several minor language specific details.

Input parameters:

x contains values for Class 1

y contains values for Class 2

name (optional) is array of strings or cell array of strings:

name(1) is name of attribute (title of histogram)

name(2) is name of the first class (the first element of legend)

name(3) is name of the second class (the second element of legend)

if name is omitted or specified as [] then graphs will not be formed.

acc (optional) is string or handle of accuracy measure function. For

string there are several appropriate values:

'BA' means balanced accuracy: (TP / (TP + FN) + NT / (TN + FP)) / 2

'accuracy' means (TP + TN) / (TP + FN + TN + FP)

'f1' means F1 score 2 \* TP / (2 \* TP + FN + FP)

'NPV' means Negative predictive value TN / (TN + FN)

'PPV' means Positive predictive value TP / (TP + FP)

'TPR', 'recall', 'sens', 'power' means true positive rate (recall,

probability of detection, hit rate, power) TP / (TP + FN)

'TNR', 'spec', 'sel', means true negative rate (specificity,

selectivity) TN / (TN + FP)

For function handle default value is @balancedAccuracy. function

must have following syntaxis:

function acc = funcName(TP, FP, TN, FN)

where TP means true positive, FP means false positive, TN means

true negative, FN means false negative.

Outputs:

bestT is optimal threshold

bestErr is minimal error which corresponds to threshold bestT. Error is

one minus accuracy defined by parameter acc.

If argument name was specified, then function also formed one figure. It is necessary to take into account that some indicators of classifier quality can produce meaningless thresholds. Figure 2 presents figures for different classifier quality measure. As we can see only three of them are more or less reasonable: accuracy, balanced accuracy and F1 score. For all other threshold is defined at one of border because these criteria considered quality of prediction of one class only.

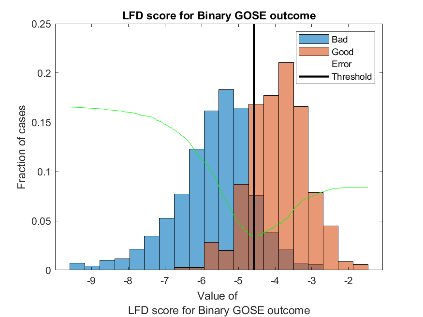
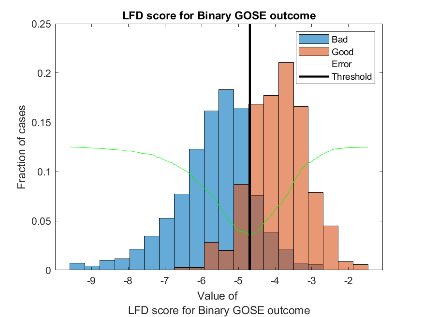
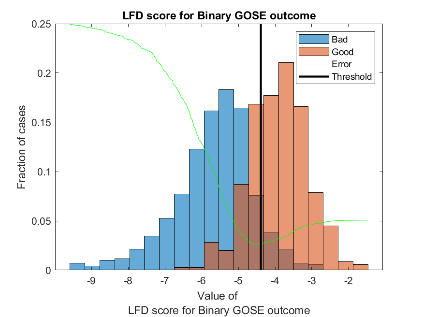
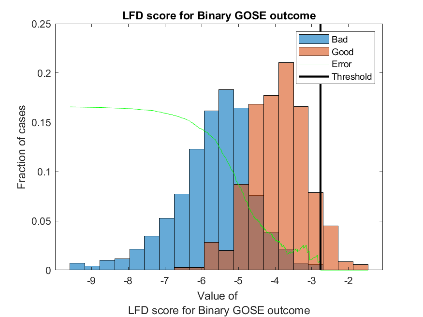
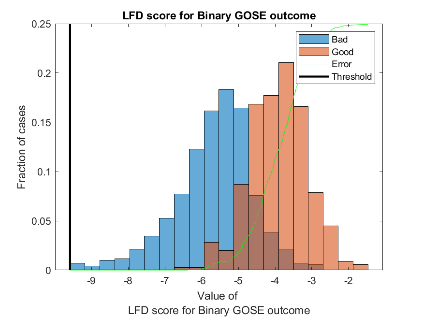
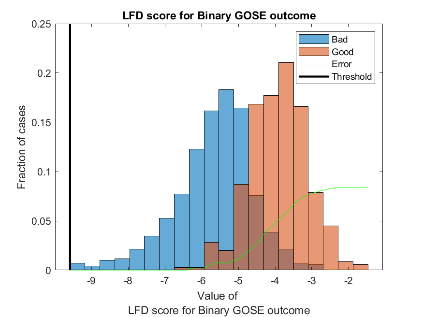
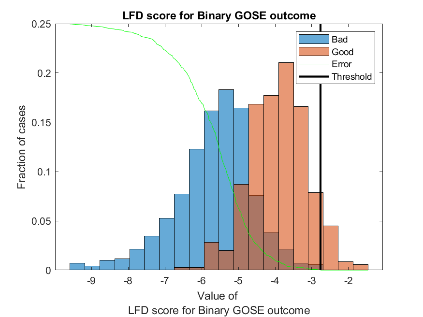
abc  
def  
g

Figure . Results of search of optimal threshold for a) accuracy, b) balanced accuracy, c) F1 score, d) NPV, e) PPV, f) TNR, and g) TPR.

# References

1. Reference will be added after acceptance of paper
2. Mansouri, K.; Ringsted, T.; Ballabio, D.; Todeschini, R.; Consonni, V. Quantitative structure–activity relationship models for ready biodegradability of chemicals. J. Chem. Inf. Model. 2013, 53, 867–878.
3. QSAR biodegradation. UCI Machine Learning Repository. Available online: https://archive.ics.uci.edu/ml/datasets/QSAR+biodegradation (accessed on 31 March 2024).