MLSM 2023 - Quiz 3
29/09/2023 - 11/10/2023 at 23:59

Questions based on Lecture 4 and 5

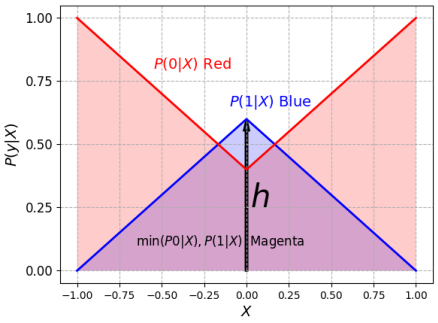
(1) (1.0 pt.)

This question is about computing Bayes error of Bayes classifiers. Let the inputs, x, be chosen from the interval $[-1, +1]$, and the labels, y, from the set $\{0, 1\}$. We are given the following conditional probabilities for all x and y to describe the relationship between x and y:

$$Pr(1|x) = \begin{cases} +hx + h & x \in [-1, 0], \\ -hx + h & x \in [0, +1], \end{cases}$$

$$Pr(0|x) = 1 - Pr(1|x) = \begin{cases} -hx + 1 - h & x \in [-1, 0], \\ +hx + 1 - h & x \in [0, +1], \end{cases}$$

where $h = 0.75$.



Assume that x has uniform distribution on $[-1, 1]$.

The question: what is the value of the Bayes error if $h = 0.75$?

Hint: You might solve this problem by computing the integral $\int_{-1}^{+1} \min(Pr(1|x), Pr(0|x)p(x)dx$, where $p(x)$ is the density function of the variable x, or by considering the shape of the minimum function.

(1) log(2)
(2) 0.5123
(3) 0.2917
(4) $1/\sqrt{2}$

(2) (1.0 pt.)

In applying the Perceptron algorithm on a data set, $\{(\mathbf{X}_i, y_i)\}_{i=1}^m$ $y_i = -1, +1, \forall i$, we might normalize all input vectors $\mathbf{X}$ with the largest absolute value of the input components. In vectorized Numpy form we have that:

`xnorm = np.max(np.abs(X)); X = X/xnorm`,

where X is a matrix containing the input vectors in its rows.

It is assumed that the margin γ is the largest value satisfying $y_i \mathbf{w}_* \mathbf{x}_i \geq \gamma$ for all $i = 1, \ldots, m$.

Question: what is the effect of this type of normalization? Assume that the two classes are linearly separable, and no bias term is included into the predictor function. In the questions, the expected number of iteration is denoted by t in the last item of Novikoff's theorem on the corresponding slide.

Select that answer which is true if this kind of normalization is applied!

(1) The normalization does not change the weight vector $\mathbf{w}$.
(2) In the Novikoff's Theorem, the number of expected iterations does not depend on this type of normalization.
(3) In the Novikoff's Theorem, the number of expected iterations increases when $xnorm > 1$.
(4) In the Novikoff's Theorem, the number of expected iterations decreases when $xnorm > 1$.

(3) (2.0 pt.)

This question, and also the next one, is about selecting the best performing hyperparameters by cross-validation. Here we assume that the full data set is used as training which is split into training and validation sets. In these scenarios we don't consider an independent test set. This model could be extended to a complete nested cross-validation.

Let the stochastic gradient algorithm for Logistic Regression be applied to find the best classifier on the Breast Cancer dataset of the Sklearn package. That algorithm is presented in Lecture 5. The potential values of the hyperparameters of the learning problem, the step size and number of iterations are shown in the program example below. The labels of the Breast Cancer dataset are of $\{0, 1\}$ which is converted into $\{-1, +1\}$.

In each training, initialize the weight vector, $\mathbf{w}$, to have zero components to reduce the dependence of the results on the random initialization.

Normalize the rows of the input matrix to have the L_∞ norm to be equal to 1, see the code example below. *Without that normalization overflow error might occur in the exponential function!*

The training examples are processed in the order appearing in the original data file, no randomization is applied to avoid the random fluctuation of the accuracy values. The hyperparameters are given by the following rules. The number of iteration is equal to 50. The step size is taken from a list $[0.1, \ldots, 1.0]$, see the python code below.

The implementation might start with these lines:

```
import numpy as np
from sklearn.datasets import load_breast_cancer
from sklearn.model_selection import KFold
from sklearn.metrics import roc_auc_score

# load the data
X, y = load_breast_cancer(return_X_y=True)  ## X input, y output

mdata,ndim = X.shape

## to convert the {0,1} output into {-1,+1}
y = 2*y - 1

## hyperparameters of the learning task of Question 3
## list of step sizes
leta = [0.1 * (i+1) for i in range(10)]

## number of iteration
iteration = 50

nfold = 5         ## number of folds

np.random.seed(12345)  ## fix the random seed to avoid random
fluctuation of the results

## to split the data into 5-folds we need
cselection = KFold(n_splits=nfold, random_state=None, shuffle=False)
```

```
## normalization
## scaling the rows by maximum absolute value, L infinite norm of columns
X /= np.outer(np.ones(mdata),np.max(np.abs(X),0))
```

The task is to run the Logistic Regression algorithm via the 5-fold cross-validation. The training and validation examples are selected by applying the KFold function from the `sklearn.model_selection` module. In each fold compute the ROC-AUC score, the area under the ROC curve as an accuracy measure on the corresponding validation set. . Run this procedure for all step size hyperparameters, see the list, `leta`, in the example code. Thus there are $50 = 10 * 5$ training and validation sets altogether. For each step size hyperparameter compute the average ROC-AUC score on the 5 folds.

Question: what is value of the maximum average score, and the corresponding value of the step size of the 10 different step sizes.

Round the numbers up to 2 decimals, and take the closest one shown in the possible answers..

Be aware, the Logistic Regression algorithm of Lecture 5 is not the same which is implemented in the Sklearn. use that version which is presented in the Lecture!

(1) $0.94, 0.6$
(2) $0.89, 0.1$
(3) $0.96, 0.7$
(4) $0.91, 0.3$

(4) (1.0 pt.)

In this question the code constructed in Question 3 is reused with the following modifications. The step size is fixed to 0.1, but the number of iterations is going through on a list $[10, \ldots, 100]$, see the python example code below. Every other part of the code: the algorithm of the logistic regression, the 5-fold cross validation and the row wise normalization by L_∞ norm remains the same. The accuracy is also measured in the same way, thus the ROC-AUC score is used. The weight vector, $\mathbf{w}$, is initialized to zeros in this question as well.

The hyperparameter setting needs to follow this example code.

```
## hyperparameters of the learning task of Question 4
## step size
eta = 0.1

## list of number of iterations
literation = [10*(i+1) for i in range(10)]
```

Run the procedure for all iteration hyperparameters, see the list, `literation`, in the example code above. Thus there are $50 = 10 * 5$ training and validation sets altogether in this task as well. For each iteration hyperparameter compute the average ROC-AUC score on the 5 folds.

Question: what is value of the maximum average score, and corresponding value of the iteration of the 10 different number of iterations.

Round the numbers up to 2 decimals, and take the closest one shown in the possible answers..

(1) $0.91, 100$
(2) $0.88, 50$

(3) $0.97, 30$

(4) $0.94, 90$