ML4ENG Coursework – Part 1

Deadline for submission (enforced by Keats): 20 November 2022 at 16:00.

1 General Guidelines

This coursework concerns the problem of binary prediction for a heart attack using the data set [1]. To start, download the data set `dataset_heart_attack.mat` and the template file `template_cw1.m` from the module's Keats website. Once this is done:

1. Change the `template_cw1.m` to your k number. In the following, we will refer to this file as `k12345678.m.`
2. Open the `k12345678.m` file with your MATLAB editor. Note that the file contains a preamble, referred to as main body, which you should **not** modify, and the definition of several functions.
3. Follow the Instructions (Section 3 of this document) to fill in the details of the functions in the template file. The main body of k12345678.m has been divided into sections, with each section containing one or more functions to be completed. The functions in the k12345678.m file have been numbered according to the numbered list below in Section 3 (Instructions).
4. Once you have written the functions, verify `k12345678.m` runs without errors when the file is included in a folder containing **only** the file itself and the data set `dataset_heart_attack.mat.`
5. Check that **no** MATLAB toolbox was used: the output of the last main body line should be `matlab` with no further toolboxes.
6. Submit only the `k12345678.m` file on Keats. No other files are allowed.

IMPORTANT: Excessive printouts (caused by omitting ';') will incur a mark loss. The use of MATLAB toolboxes will also cause the subtraction of mark points. Please carefully follow matrix sizes and vector dimension (row or column).

2 DATA SET

The file `dataset_heart_attack.mat` contains a data set $\mathcal{D} = \{x_i, t_i\}_{i=1}^{N}$, which consists of $N = 303$ examples. Each example consists of:

1. Input vector x_i in $\mathbb{R}^{13}$, encompassing $d = 13$ medical features.
2. Its corresponding binary label $t_i \in \{0,1\}$, where 1 stands for high chance of heart attack and 0 for low chance as diagnosed by a medical expert.

The data is loaded into the workspace to have

Name	**Size**	**Type**	**Description**
`t`	$N \times 1$	Logical	Diagnosis (binary label): 1 = high chance of heart attack and 0 = low chance.
`X`	$N \times d$	Double	Data matrix (samples vectors as rows)
`x_titles`	$1 \times d$	String	Description for the *d* features in x

The input sample vector is denoted as

$$x = [x^{(1)} \quad \dots \quad x^{(d)}]^T,$$

and the inputs of the data sets are given by stacking up N samples

$$X = \begin{bmatrix} x_1^T \\ \vdots \\ x_N^T \end{bmatrix} = \begin{bmatrix} x_1^{(1)} & \cdots & x_1^{(d)} \\ \vdots & \ddots & \vdots \\ x_N^{(1)} & \cdots & x_N^{(d)} \end{bmatrix}.$$

The labels are also stacked up, forming the vector

$$t = \begin{bmatrix} t_1 \\ \vdots \\ t_N \end{bmatrix}.$$

The entries of the vector `x_titles` annotate the d features.

3 Instructions for Completing the Coursework

Section 1

Assume you are given the sensitivity and specificity values of a heart attack hard predictor $\hat{t}(\cdot)$. Furthermore, the prior of having a heart attack $p(t=1)$ is also known.

1. [10 points] Design the function

   ```
   function tn= true_negative(sens, spec, prior)
   ```

   that calculates the probability of a negative test to be correct, meaning $p(t=0|\hat{t}=0)$, by using Bayes' rule. All three arguments are scalars representing ratios in the interval $[0,1]$: `sens` is the sensitivity, `spec` is the specificity and `prior` is the prior.

Section 2

In this section, we split the full data set $\mathcal{D}$ into training $\mathcal{D}^{\mathrm{tr}}$ and test set $\mathcal{D}^{\mathrm{te}}$ using splitting ratio $\eta \in (0,1)$.

2. [10 points] Design the function

   ```
   function [X_tr, t_tr, X_te, t_te]= split_tr_te(X, t, eta)
   ```

   that splits the input data {`X`, `t`} set into two disjoint data set. The training set {`X_tr`, `t_tr`} should have the **last** $N^{\mathrm{tr}} = \mathrm{round}(\eta N)$ samples and labels, and the test set {`X_te`, `t_te`} the **first** $N^{\mathrm{te}} = N - N^{\mathrm{tr}}$. Here η stands for the ratio of the training data set size from the entire data set. Note that this partition involves no randomness.

The main body of code splits the data set using $\eta = 0.7$.

Section 3

3. [10 points] Design the function

   ```
   function loss = detection_error_loss(t_hat, t)
   ```

   that computes the empirical detection-error loss $L_{\mathcal{D}} = E_{(\mathrm{x},\mathrm{t})\sim p_{\mathcal{D}}(x,t)}[1(\mathrm{t}\neq\hat{t}(\mathrm{x}))]$ of binary predictions `t_hat` (as a vector $\hat{t}$ which operated over some input $\hat{t}(\mathrm{x})$ which is not given here) with respect to the true targets `t`, both vectors of the same length.

In the main code, this function runs over two suggested hard predictors: the one following the `sex` feature and the other following `fbs` feature, which is the binary variable 1(fasting blood sugar > 120 mg/dl), with 1() being the indicator function.

Section 4

We wish to operate over the next loss function $\ell(t, \hat{t})$

$t \ \backslash \ \ \hat{t}$	0	1
0	0	10
1	3	0

4. [10 points] Design the function

   ```
   function loss = loss_func(t_hat, t)
   ```

   that computes the empirical loss $L_{\mathcal{D}} = E_{(\mathrm{x},\mathrm{t})\sim p_{\mathcal{D}}(x,t)}[\ell(\mathrm{t}, \hat{t}(\mathrm{x}))]$ of binary predictions `t_hat` (as a vector $\hat{t}$ which operated over some input $\hat{t}(\mathrm{x})$ which is not given here) with respect to the true targets `t`, both vectors of the same length.

Section 5

In this section, we train hard predictors based on the available training data. To this end, we consider linear predictors using a different number of features $M \in \{0,1, \dots, 13\}$. A predictor using M features selects the first M features of the inputs

$$u_M(x) = \left[1, x^{(1)}, x^{(2)}, \dots, x^{(M)}\right]^T \in \mathbb{R}^{M+1}$$

as feature vector. Recall that x is the $d = 13$-dimensional input feature vector. The model class is accordingly defined as

$$\mathcal{H}_M = \{\hat{t}(\cdot\,|\theta_M) = \theta_M^T \cdot u_M(x) | \theta_M \in \mathbb{R}^{M+1}\}.$$

To train the predictors for a given order M, we optimize the model parameter vectors θ_M using the quadratic loss by solving a standard least squares problem over training data matrix. The LS function is provided.

5. [10 points] Design the function

   ```
   function out = X_M(X,M)
   ```

   with input data matrix `X` of size $N \times d$ (N is input depended, can be extracted by the dimensionality of `X`) and order `M` $\in \mathbb{R}$ that produces the data matrix of size $N \times (M + 1)$ using the feature mapping $u_M(\cdot)$

$$X_M = \begin{bmatrix} (u_M(x_1))^T \\ \vdots \\ (u_M(x_N))^T \end{bmatrix}.$$

Section 6

This section visualises the predictor $\hat{t}_2$ on the two-dimensional space of input variables $x^{(1)}$ and $x^{(2)}$. To this end, it spans the space using a grid and it predicts for each sample in that grid `X_gr` the outcome of the two predictors. Since the LS prediction $\theta_2^T \cdot u_2(x)$ is continuous and not binary, clipping to the interval $[0,1]$ is done, and hard thresholding as

$$\hat{t}^{\text{hard}}(x|\theta_2) = \begin{cases} 1, & \theta_2^T \cdot u_2(x) > 0.5 \\ 0, & \text{otherwise} \end{cases}$$

is applied to determine the decision region. The labelled test set is illustrated on top of the predictors' outcomes.

6. [10 points] Design the function

   `function out = linear_combiner(X, theta)`

   that applies the predictor $\theta^T \cdot u(x)$ (with `theta` for theta of arbitrary length $M + 1$ and the data matrix `X` of size $N \times (M+1)$) to each input features sample in data matrix `X` (i.e., to each row $u(x)$ of the matrix).

Section 7

We further evaluate the mean square error (MSE) loss on its binary targets of a predictor. We use the test set for this purpose.

7. [10 points] Design the function

   `function out = mse_loss(t_hat, t)`

   that computes the empirical MSE loss of prediction `t_hat` using the true labels `t`, both vectors of the same length.

Section 8

We now wish to see the dependency of the MSE loss over the order M.

8. [20 points] Design the function

   `function out = mse_vs_M(X_tr, t_tr, X_te, t_te)`

   that uses all given samples in the split data sets (see section 2 for arguments details. For each $M = 0,1,\ldots,13$, it trains using the entire training data (solving an LS problem) a model set using model class of order `M`, and then computes the empirical

MSE test loss of the predictor using the true test labels `t_te`. The output is a column vector `out`$\in \mathbb{R}^{14}$, with the test losses of $[L_{\mathcal{D}^{\mathrm{te}}}(\theta_0), L_{\mathcal{D}^{\mathrm{te}}}(\theta_1), \ldots, L_{\mathcal{D}^{\mathrm{te}}}(\theta_{13})]^T$.

Once coded, a graph will be shown.

Section 9

In this section, the input matrix features are reversed, and will follow the same steps.

9. [10 points] Add a two-line comments in `function` **`discussion`**`()` why the MSE test loss of the reversed feature is not identical to the original ordering. Infer which feature group is more useful for heart attack prediction - is it the lower indexed features or the higher ones?

4 REFERENCES

[1] https://www.kaggle.com/datasets/rashikrahmanpritom/heart-attack-analysis-prediction-dataset

[2] https://uk.mathworks.com/academia/tah-portal/kings-college-london-30860095.html