Machine Learning Course

Second intermediate assessment – June 4, 2019

Students should do all the exercises to get the maximum score.
If you solve all the three exercises correctly, you get 33 points.
Please, justify carefully each answer.

Name: ……………………………. Surname: ……………………………. Student ID: ………..

Exercise 1 (11 points)

Given the n=6 two-dimensional data points **x**, and their labels **y**: $\mathbf{x} = \begin{bmatrix} -2 & 0 \\ 0 & -2 \\ -2 & -1 \\ 1 & 1 \\ 1 & 0 \\ 2 & 1 \end{bmatrix}$, $\mathbf{y} = \begin{bmatrix} -1 \\ -1 \\ -1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$,

find the linear discriminant function using a **batch gradient-descent algorithm** to minimize the following objective function (i.e., the L2-regularized *logistic loss*):

$$L(\boldsymbol{w}, b) = \frac{1}{n}\sum_{i=1}^{n} \ln(1 + e^{-y_i f(\boldsymbol{x}_i)}) + \frac{\lambda}{2}\, \boldsymbol{w}^T\boldsymbol{w}, \text{ where } f(\boldsymbol{x}_i) = \boldsymbol{w}^T\boldsymbol{x}_i + b.$$

Initialize $\boldsymbol{w} = [-1,\ 1]^T,\ b = 0.1$, the gradient step size $\eta = 1.0$, the parameter $\lambda = 1.0$, and the threshold on the termination condition $\theta = 1.1$.
Use the L1 norm to compute $|\nabla_w L\ \ (\boldsymbol{w}, b)| + |\nabla_b L\ \ (\boldsymbol{w}, b)|$ in the termination condition.

- State the gradient-descent learning algorithm, and compute the required gradients.
- Compute $\boldsymbol{w}, b$ for the first two iterations of the algorithm, and check if it converges.
- Plot the initial decision boundary along with the training points, and how it changes during the first two iterations of the algorithm.

Exercise 2 (12 points)

Given the two-dimensional data points **x** in Exercise 1, and the initial k=2 centroids $\mathbf{v} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$:

- Cluster the data points x using the k-means algorithm, reporting the clustering labels, the updated centroids and the objective function at each iteration of the algorithm. For simplicity, use the L1 (Manhattan) distance instead of the L2 (Euclidean) distance, both for computing the objective function $\sum_{i=1}^{n}\left|\left|\boldsymbol{x}_i - \boldsymbol{v}_k\right|\right|_1$ (being $\boldsymbol{v}_k$ the closest centroid to $\boldsymbol{x}_i$) and for computing the distances between the data points x and the centroids v. If a point has the same distance with respect to a number of centroids, assign it to the centroid with the lowest class index in this set (e.g., if the point has the same distance w.r.t. centroid 0 and 2, assign it to centroid 0).
- Make a two-dimensional plot displaying the data points (with a clear indication to explain to which cluster each point belongs to, after the last iteration) and the final centroids.
- Plot the decision boundaries of the nearest mean centroid classifier that uses the final centroids of the k-means algorithm as the estimated centroids of each class.

Exercise 3 (10 points)
Given the two-dimensional training points $\mathbf{x}_{\text{tr}}$, along with their labels $\mathbf{y}_{\text{tr}}$, and a set of test examples $\mathbf{x}_{\text{ts}}$, with their labels $\mathbf{y}_{\text{ts}}$

$$\mathbf{x}_{\text{tr}} = \begin{bmatrix} -1 & 1 \\ 1 & -1 \\ 3 & 2 \\ 1 & 2 \\ 3 & -1 \end{bmatrix}, \quad \mathbf{y}_{\text{tr}} = \begin{bmatrix} 2 \\ 2 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \quad \mathbf{x}_{\text{ts}} = \begin{bmatrix} 1 & 1 \\ 2 & 2 \\ 0 & 0 \\ 1 & 2 \end{bmatrix}, \quad \mathbf{y}_{\text{ts}} = \begin{bmatrix} 0 \\ 0 \\ 2 \\ 1 \end{bmatrix},$$

classify the points in $\mathbf{x}_{\text{ts}}$ with a k-NN algorithm with k=1, using the l2 distance as the distance metric. The distance matrix computed by comparing $\mathbf{x}_{\text{ts}}$ against $\mathbf{x}_{\text{tr}}$ is given below:

```
[[2.   2.   2.24 1.   2.83]
 [3.16 3.16 1.   1.   3.16]
 [1.41 1.41 3.61 2.24 3.16]
 [2.24 3.   2.   0.   3.61]]
```

- Compute the classification error. In case of equal (minimum) distances between a given test sample and a subset of the training points, assign the test sample to the class of the first point of the training set (from left to right in the distance matrix).
- Plot the decision function of the given k-NN classifier.

EXERCISE 1 - SOLUTION

The algorithm is:

begin initialize w, θ, η, $k=0$

repeat

$w = w - \eta\, \nabla_{w} L\ \ (\boldsymbol{w}, b)$

$b = b - \eta\, \nabla_{b} L\ \ (\boldsymbol{w}, b)$

until $\eta\,(\|\nabla_w L\ \ (\boldsymbol{w},b)\| + |\nabla_b L\ \ (\boldsymbol{w},b)|) < \theta$

We need to compute the derivatives of the objective function w.r.t. $\boldsymbol{w}$ and b:

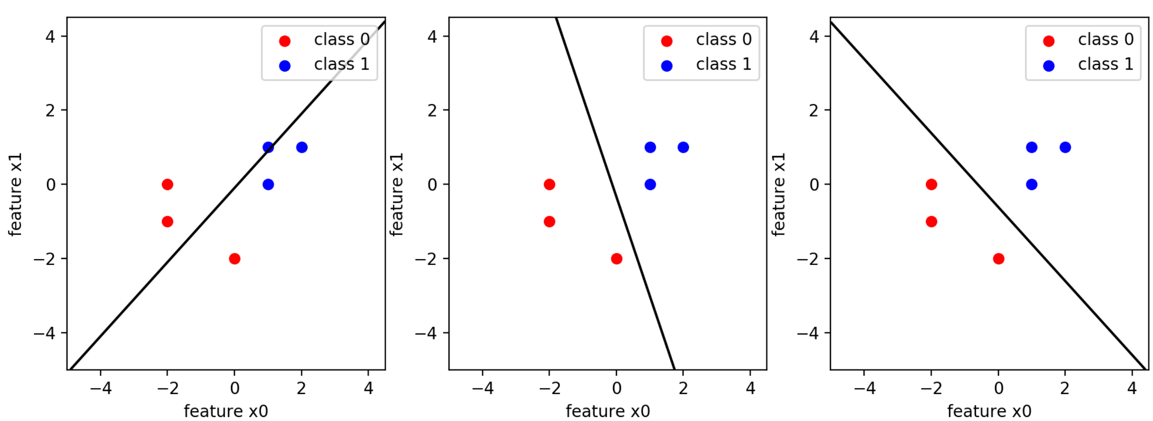
$$\nabla_{\boldsymbol{w}} L = \frac{1}{n}\sum_{i=1}^{n} -y_i \frac{e^{-y_i f(x_i)}}{1+e^{-y_i f(x_i)}}\boldsymbol{x}_i + \lambda\, \boldsymbol{w}$$

$$\nabla_b L = \frac{1}{n}\sum_{i=1}^{n} -y_i \frac{e^{-y_i f(\boldsymbol{x}_i)}}{1+e^{-y_i f(\boldsymbol{x}_i)}}$$

iter	*L(**w**,b) **	$\nabla_w L$	$\nabla_b L$	*w*	*b*	*term. cond.*	θ
0	2.145	[-1.982 0.634]	-0.021	[0.982 0.366]	0.121	2.64	1.1
1	0.759	[0.804 0.188]	0.012	[0.178 0.178]	0.109	1.0	1.1

*(*) The objective function here is computed for **w**,b **after update***

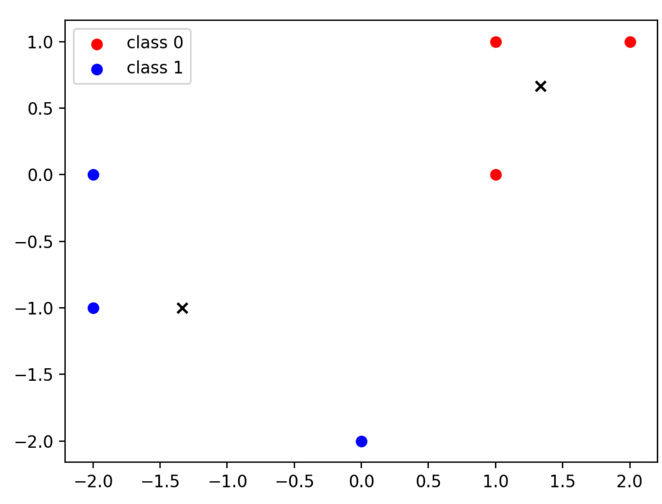
The algorithm has converged after the second iteration.



EXERCISE 2 – SOLUTION

iter	$\sum_{i=1}^{n} \lvert\lvert \boldsymbol{x}_i - \boldsymbol{v}_k \rvert\rvert_1$	*distance matrix*	*cluster assignments*	*current* $\boldsymbol{v}$
0	11.0	[[**4**. 4.] [4. **2**.] [5. **3**.] [**0**. 2.] [**1**. 1.] [**1**. 3.]]	0 1 1 0 0 0	[[1. 1.] [1. -1.]]
1	9.5	[[3. **2.5**] [3. **1.5**] [4. **1.5**] [**1.** 4.5] [**1.** 3.5] [**2.** 5.5]]	0 0 0 1 1 1	[[0.5 0.5] [-1. -1.5]]
2	7.33	[[4. **1.67**] [4. **2.33**] [5. **0.67**] [**0.67** 4.33] [**1.** 3.33] [**1.** 5.33]]	0 0 0 1 1 1	[[1.33 0.67] [-1.33 -1.]]

After iteration 2, the cluster assignments do not change anymore. Therefore, the algorithm stops. The final clustering (along with the centroids) is shown below.



EXERCISE 3 – SOLUTION

It is not difficult to see that the minimum distances per row are indexed as [3 2 0 3]:

```
[[2.   2.   2.24 1.   2.83]
 [3.16 3.16 1.   1.   3.16]
 [1.41 1.41 3.61 2.24 3.16]
 [2.24 3.   2.   0.   3.61]]
```

Recalling that the training labels are: [2 2 0 1 0], the test samples are thus classified as: yc = [1 0 2 1].
The true labels are: [0 0 2 1], and thus, the classification error is ¼ = 25%.