Machine Learning Course – Exam – July 23, 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 (10 points)

Let's consider a 2-class problem in a one-dimensional feature space bounded in [0,1], i.e., $x \in [0,1]$. The class-conditional densities are: $p(x|\omega_1) = 2 - 2x$, and $p(x|\omega_2) = 2x$, both defined in [0,1]. Assume that the prior probabilities of the two data classes are $P_1 = P_2$, and that the cost of errors of class ω_2 is 1.5 times that of class ω_1, that is, $\lambda_{12} = 1.5\lambda_{21}$.

- (5 points) Compute the minimum-risk decision regions, and the Bayesian decision regions, and compute the classification error in both cases.
- (3 points) Plot the joint distributions $P_k p(x|\omega_k)$ for k=1,2 on the one-dimensional feature space, along with the minimum-risk and the Bayesian decision regions.
- (2 points) In the plot, highlight the area(s) corresponding to the additional error incurred when using the minimum-risk decision. One class shows a higher error. *Which one? Why?*

Exercise 2 (10 points)

Let us consider a 3-class problem in R^2 (two-dimensional feature space), where the likelihood of each class is Gaussian and given as $p(x|\omega_i) = N(\boldsymbol{\mu_i}, \Sigma_i)$, with

$\Sigma_i = \sigma^2\mathbf{I}$; $\mu 1 = \begin{pmatrix}-1\\-1\end{pmatrix}$; $\mu 2 = \begin{pmatrix}+1\\-1\end{pmatrix}$; $\mu 3 = \begin{pmatrix}1\\1\end{pmatrix}$;

and prior probabilities $P_1 = P_2 = P_{3.}$ Compute the decision boundaries and plot them.

Exercise 3 (13 points)

Given the n=6 two-dimensional data points **x**, and their labels **y**:

$$\mathbf{x} = \begin{bmatrix} -2 & 1 \\ 1 & -1 \\ -1 & -1 \\ 1 & 1 \\ 1 & 2 \\ 0 & 1 \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} -1 \\ -1 \\ -1 \\ 1 \\ 1 \\ 1 \end{bmatrix},$$

find the linear discriminant function using gradient descent to minimize the following objective:

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

Initialize $\boldsymbol{w} = [0.1, 0.5]^T$, $b = 0$, the gradient step size $\eta = 0.1$, the parameter $\lambda = 5.0$, and the threshold on the termination condition $\theta = 0.2$. Use the l1 norm to compute the termination condition.

- (8 points) State the gradient-descent learning algorithm, compute $\boldsymbol{w}, b$ for the first two iterations of the algorithm, and check if it converges.
- (5 points) Plot the initial decision boundary along with the training points, and how it changes during the first two iterations of the algorithm (note that these are three hyperplanes in total).

Exercise 4 (only for students who are repeating the second part)
Given the two-dimensional data points **x**, and the initial k=3 centroids **v**,

$$\mathbf{x} = \begin{bmatrix} -1 & 0 \\ 0 & -1 \\ 2 & 2 \\ 1 & 0 \\ 3 & 3 \end{bmatrix}, \qquad \mathbf{v} = \begin{bmatrix} 1 & 1 \\ 2 & 2 \\ 1 & -1 \end{bmatrix},$$

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 $\sum_{i=1}^{n} \left\| \boldsymbol{x}_i - \boldsymbol{v}_k \right\|_1$ (being $\boldsymbol{v}_k$ the closest centroid to $\boldsymbol{x}_i$) and 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.

Solutions

Exercise 1

Computation of the Bayesian regions. The prior probabilities are clearly $P(\omega_1) = 0.5; P(\omega_2) = 0.5$. The Bayesian decision regions are obtained by noting that:

$$\frac{p(x|\omega_1)}{p(x|\omega_2)} > \frac{P(\omega_2)}{P(\omega_1)} = 1$$

This means that $(2 - 2x_b)/2x_b = 1$, $x_b = 0.5$, and the decision regions are: $R_1[0, x_b]; R_2[x_b, 1]$.

Computation of the minimum-risk decision regions. In this case, we decide for class 1 if

$$\frac{p(x|\omega_1)}{p(x|\omega_2)} > \left(\frac{\lambda_{12} - \lambda_{22}}{\lambda_{21} - \lambda_{11}}\right)\frac{P(\omega_2)}{P(\omega_1)} = 1.5$$

This means that $(2 - 2x_r)/2x_r = 1.5$, and $x_r = 0.4$. The decision regions are: $R_1[0, x_r]; R_2[x_r, 1]$.

Computation of the Bayesian error. The Bayesian error is given as:

$$P(error|x \in \omega_1)P_1 + P(error|x \in \omega_2)P_2 = P_1\int_{x_b}^{1}(2 - 2x)dx + P_2\int_{0}^{x_b}(2x)\,dx =$$
$$0.5[2x - x^2]_{0.5}^{1} + 0.5[x^2]_0^{0.5} = 0.5(1 - 0.75) + 0.5(0.25) = 0.125 + 0.125 = 0.25$$

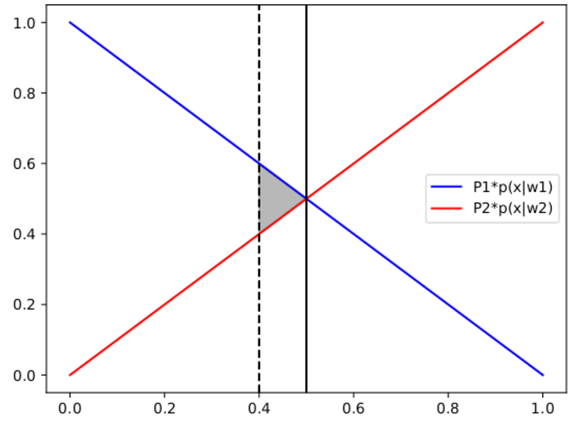
Computation of the minimum-risk error. The minimum-risk error is given as:

$$P(error|x \in \omega_1)P_1 + P(error|x \in \omega_2)P_2 = P_1\int_{x_r}^{1}(2 - 2x)dx + P_2\int_{0}^{x_r}(2x)\,dx =$$
$$0.5[2x - x^2]_{0.4}^{1} + 0.5[x^2]_0^{0.4} = 0.5(1 - 0.64) + 0.5(0.16) = 0.18 + 0.08 = 0.26$$

According to the condition $\lambda_{12} = 1.5\lambda_{21}$, the threshold is shifted to the left (towards class 1), from x=0.5 to x=0.4, in order to reduce the number of errors on samples of class 2 (which have a higher cost).

The total error increases. The Bayesian error is 0.25, while the minimum-risk error becomes 0.26. The error probability for patterns belonging to class 2 decreases from 0.125 to 0.08. *(Note that, in this case, the error could also be computed directly from the plot, by computing the area of the involved triangles, without actually computing the integrals)*.

The joint distributions are represented in the plot. The Bayesian threshold is represented with a solid black line whereas the minimum-risk threshold is represented with a dotted black line. The additional error incurred when using the minimum-risk decision criterion is highlighted in grey.



Exercise 2

The generalized discriminant function for Gaussian distributions is:

$$g(x) = -\frac{1}{2}x^T\Sigma^{-1}x + \mu^T\Sigma^{-1}x - \frac{1}{2}\mu^T\Sigma^{-1}\mu + \ln p(\omega) - \frac{1}{2}\ln|\Sigma|$$

In this case, the covariance matrix is isotropic, and equal for all classes. Even the priors are the same. Thus, the above expression can be simplified as: $g(x) = \mu^T x - \frac{1}{2}\mu^T\mu$

Notably, the second term is also equal for all classes ($\mu^T\mu = 2$ for all classes), and thus this term can also be removed from the discriminant function. Therefore, we obtain: $g(x) = \mu^T x$.

Accordingly, for each class we have

$g_1(x) = \mu_1^T x = -x_1 - x_2$; $g_2(x) = \mu_2^T x = +x_1 - x_2$; $g_3(x) = \mu_3^T x = +x_1 + x_2$

Let us now compute the class boundaries between each pair of classes.

Class boundary between class 1 and class 2. We start by finding x^* for which $g_1(x^*) = g_2(x^*)$:

$(\mu_1 - \mu_2)^T x = 0$, which implies $x_1 = 0$

This boundary is aligned with the y-axis (i.e., the x_2 values) and holds only for the subset of points for which it holds that

$g_1(x^*) = g_2(x^*) > g_3(x^*)$

$-x_2 > x_2$ which implies $x_2 < 0$

This boundary is thus aligned with and active for the non-positive part of the y-axis.

Class boundary between class 1 and class 3. Let us find the points x^* for which $g_1(x^*) = g_3(x^*)$:

$(\mu_1 - \mu_3)^T x = 0$, which implies $x_1 = -x_2$

The boundary holds only for the subset of points for which it holds that

$g_1(x^*) = g_3(x^*) > g_2(x^*)$, i.e., $0 > x_1 - x_2$

Now, if we substitute $x_1 = -x_2$ in the above inequality, we have $x_2 > 0$.

Alternatively, one may substitute $x_2 = -x_1$ in the above inequality and obtain $x_1 < 0$.

These conditions are clearly equivalent, as together with the boundary equation $x_1 = -x_2$ both identify the top-left quadrant of the cartesian space (where the boundary is active).

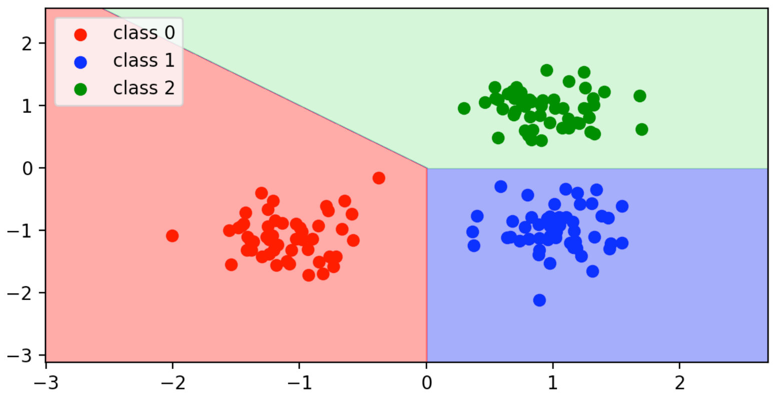
Class boundary between class 2 and class 3. Let us find the points x^* for which $g_2(x^*) = g_3(x^*)$:

$(\mu_2 - \mu_3)^T x = 0$, which implies $x_2 = 0$

This boundary is aligned with the x-axis (i.e., the x_1 values) and holds only for the subset of points for which it holds that $g_2(x^*) = g_3(x^*) > g_1(x^*)$. This implies that $x_1 > 0$

This boundary is thus aligned with and active for the positive part of the x-axis.

This plot shows the class boundaries for the three Gaussian classes with $\sigma = 0.3$



Exercise 3

The algorithm is:

```
begin initialize w, $\theta$, $\eta$, $k=0$
    repeat
        $\boldsymbol{w}=\boldsymbol{w} - \eta\, \nabla_{\boldsymbol{w}} L\ \ (\boldsymbol{w}, b)$
        $b=b - \eta\, \nabla_b L\ \ (\boldsymbol{w}, b)$
    until $\eta\, (\| \nabla_{\boldsymbol{w}} L\ \ (\boldsymbol{w}, b)\| + | \nabla_b L\ \ (\boldsymbol{w}, b)|) < \theta$
```

We need to compute the derivatives of the objective function w.r.t. ***w*** and *b:*

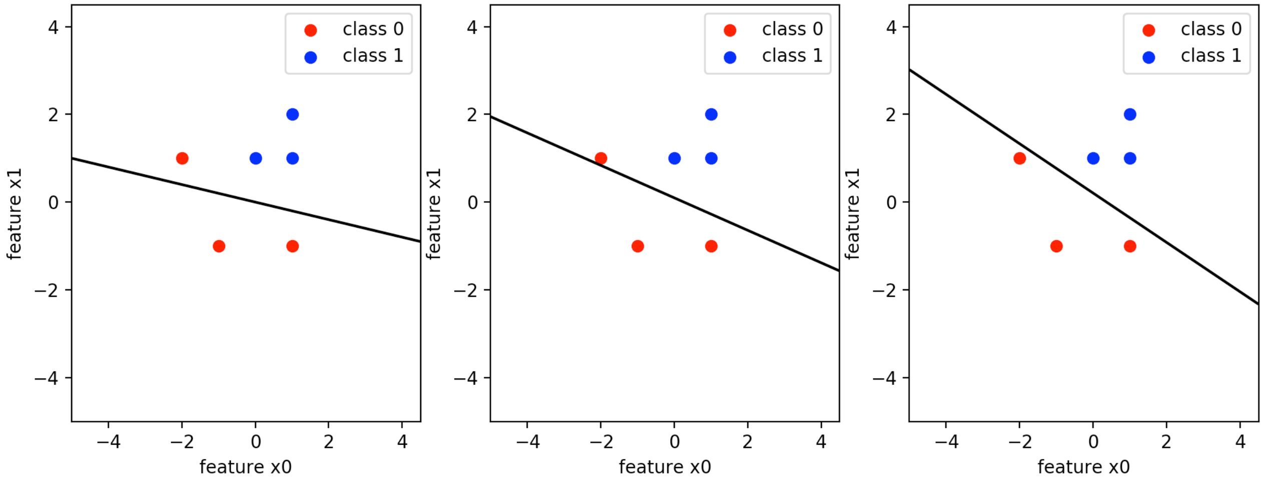
$$\nabla_{\boldsymbol{w}} L = \frac{1}{n}\sum_{i=1}^{n}(\boldsymbol{w}^T\boldsymbol{x}_i + b - y_i)\,\boldsymbol{x}_i \quad + \lambda\,\boldsymbol{w}$$

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

iter	*L(**w**,b)* *	$\nabla_{\boldsymbol{w}} L$	$\nabla_b L$	*w*	*b*	*t. cond.*	θ
0	0.869	[0.05 2.43]	0.25	[0.095 0.26]	-0.025	0.27	0.2
1	0.467	[-0.02 0.84]	0.10	[0.097 0.17]	-0.035	0.1	0.2

The algorithm has converged after the first two iterations.

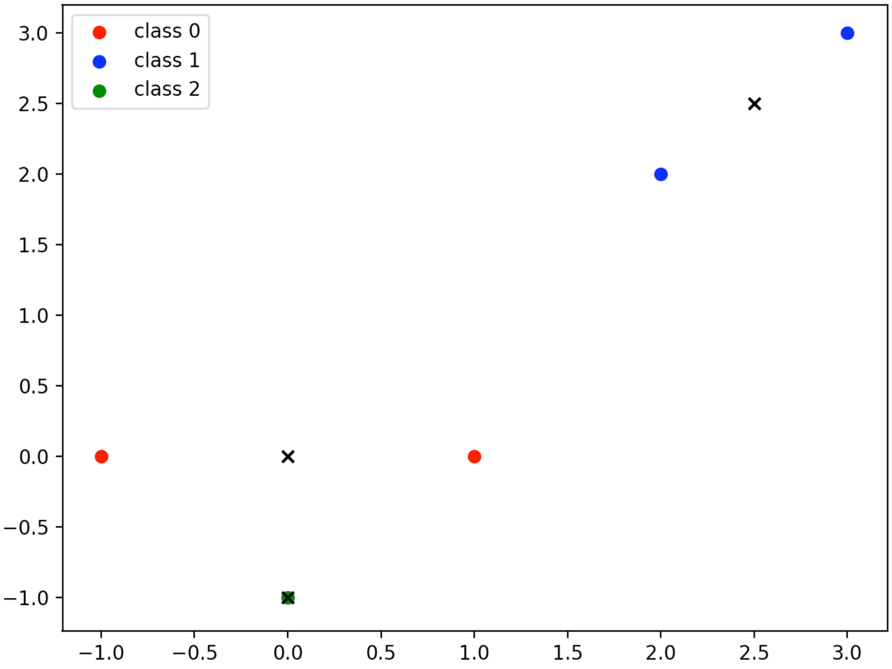
*(*) The objective function here is computed for **w**,b **after update***



Exercise 4 (only for students who are repeating the second part)

iter	$\sum_{i=1}^{n}\left\|\left\|\boldsymbol{x}_i - \boldsymbol{v}_k\right\|\right\|_1$	*distance matrix*	*cluster assignments*	*current* ***v***
0	7.0	[[3. 5. 3.] [3. 5. 1.] [2. 0. 4.] [1. 3. 1.] [4. 2. 6.]]	0 2 1 0 1	[[1. 1.] [2. 2.] [1. -1.]]
1	4.0	[[1. 6. 2.] [1. 6. 0.] [4. 1. 5.] [1. 4. 2.] [6. 1. 7.]]	0 2 1 0 1	[[0. 0.] [2.5 2.5] [0. -1.]]

After iteration 1, the cluster assignments do not change anymore. Therefore, the algorithm stops. The final clustering (along with the centroids) is shown below.


● class 0
● class 1
● class 2