Maura Pintor, PhD Student

 maurapintor

 https://maurapintor.github.io

 @maurapintor

 https://www.linkedin.com/in/maura-pintor

 maura.pintor@unica.it

Exercise 3

Let us suppose that we want to discriminate between normal and intrusive network traffic, namely, two data classes ω_N, normal traffic, and ω_{INT}, intrusive network traffic. We suppose to use a single *feature* x to characterize traffic data (one-dimensional feature space), and we assume that the model of the network traffic is the following:

$$P(\omega_N) = \frac{1}{2}; P(\omega_{INTR}) = \frac{1}{2}$$

$$p(x/\omega_i) = \frac{1}{\sqrt{2\pi}\,\sigma} \exp\left[-\frac{1}{2}\left(\frac{x-\mu_i}{\sigma} \right)^2 \right];$$

$$\mu_N = 0; \mu_{INTR} = 4; \;\; \sigma_N = \sigma_{INTR} = 1;$$

Let the cost of missing the detection of intrusion be ten times higher than the opposite error (a normal traffic is wrongly recognized as an intrusion).

a) Determine the decision regions using the likelihood ratio, without considering the costs of errors.

$$P\{x \in R_N, x \in \omega_{INTR}\} + P\{x \in R_{INTR}, x \in \omega_N\} =$$

$$P\{x \in R_N / \omega_{INTR}\} P(\omega_{INTR}) + P\{x \in R_{INTR} / \omega_N\} P(\omega_N) =$$

How to compute this integral?

$$\int_{-\infty}^{x^*} p(x \mid \omega_{INTR}) P(\omega_{INTR}) dx + \int_{x^*}^{\infty} p(x \mid \omega_N) P(\omega_N) dx =$$

$$\frac{1}{2}\left[\frac{1}{\sqrt{2\pi}} \int_{-\infty}^{2} \exp\left[-\frac{1}{2}(x-4)^2\right] dx + \int_{2}^{\infty} \frac{1}{\sqrt{2\pi}} \exp\left[-\frac{1}{2}(x)^2\right] dx\right] =$$

$$\frac{1}{2}\left[\frac{1}{\sqrt{2\pi}} \int_{-\infty}^{-2} \exp\left[-\frac{1}{2}(y)^2\right] dy + \int_{2}^{\infty} \frac{1}{\sqrt{2\pi}} \exp\left[-\frac{1}{2}(x)^2\right] dx\right] =$$

$$\frac{1}{2}[0.0228 + 0.0228] = 0.0228$$

Exercise 3 (10 points)

Given the two-dimensional training points $\mathbf{x}_{tr}$, along with their labels $\mathbf{y}_{tr}$, and a set of test examples $\mathbf{x}_{ts}$, with their labels $\mathbf{y}_{ts}$: $\mathbf{x}_{tr} = \begin{bmatrix} -1 & 0 \\ 0 & -1 \\ 2 & 2 \\ 1 & 0 \\ 3 & 3 \end{bmatrix}$, $\mathbf{y}_{tr} = \begin{bmatrix} 2 \\ 2 \\ 0 \\ 1 \\ 0 \end{bmatrix}$, $\mathbf{x}_{ts} = \begin{bmatrix} 1 & 1 \\ 2 & 2 \\ 0 & 0 \\ 1 & 2 \end{bmatrix}$, $\mathbf{y}_{ts} = \begin{bmatrix} 0 \\ 0 \\ 2 \\ 1 \end{bmatrix}$, classify the points in $\mathbf{x}_{ts}$ with a k-NN algorithm with k=1, using the l2 distance as the distance metric. The distance matrix computed by comparing $\mathbf{x}_{ts}$ against $\mathbf{x}_{tr}$ is given below:

```
[[ 2.24  2.24  1.41  1.00  2.83]
 [ 3.61  3.61  0.00  2.24  1.41]
 [ 1.00  1.00  2.83  1.00  4.24]
 [ 2.83  3.16  1.00  2.00  2.24]]
```

Classify with KNN, k=2?

Exercise 1 (13 points)
Given the n=6 two-dimensional data points **x**, and their labels **y**

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

find the linear discriminant function using a **batch gradient-descent algorithm** to minimize the following objective function (using the so-called *ramp loss*):

$L(\boldsymbol{w}, b) = \frac{1}{n}\sum_{i=1}^{n} [\max(0,1 - y_i f(\boldsymbol{x}_i)) - \max(0, -y_i f(\boldsymbol{x}_i))] + \frac{\lambda}{2}\boldsymbol{w}^T\boldsymbol{w}$, where $f(\boldsymbol{x}_i) = \boldsymbol{w}^T\boldsymbol{x}_i + b$.

The gradient of $l(yf(x)) = [\max(0,1 - yf(\boldsymbol{x})) - \max(0, -yf(\boldsymbol{x}))]$ with respect to the classifier parameters is not null only when $0 < yf(\boldsymbol{x}) < 1$, and in that case it equals $-y\,\boldsymbol{x}$ and $-y$, respectively when computed with respect to $\boldsymbol{w}$ and b.

Initialize $\boldsymbol{w} = [0.2, \ -1]^T$, $b = 0$, the gradient step size $\eta = 0.5$, the parameter $\lambda = 0.5$, and the threshold on the termination condition $\theta = 0.7$.

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.
- 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.
- Plot the ramp loss function $l(z) = [\max(0,1 - z) - \max(0, -z)]$, with respect to z. Explain why the gradient is not null only when $0 < z < 1$.
- The ramp loss is more robust to outliers in the training data. Can you explain why?

https://github.com/unica-ml/ml/blob/master/exams/20180531-pre2a.pdf

Links used today

- MachineLearningCheatSheet.pdf
- ML-tutor-04-whiteboard