Maura Pintor, PhD Student

 maurapintor

 https://maurapintor.github.io

 @maurapintor

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

 maura.pintor@unica.it

Exercise 1

Let us suppose that we want to design an anti-spam filter based on the frequency (number of occurrences) of the word "viagra" in the e-mail body. Intuitively, a higher frequency of this keyword suggests a higher level of "spamminess" (level/degree of spam content) of the e-mail. Indeed, real anti-spam filters usually compute a "score" which is assigned to each incoming e-mail, based on the frequency of some keywords contained in the body message (e.g., the keyword "viagra").

How to define the priors?

1) Define a simple model (e.g., a linear one) for the probability density functions $p(x \mid \omega_{SPAM})$ and $p(x \mid \omega_{MAIL})$ of the classes "spam" and "ham".

2) Find the Bayesian decision rule (let us indicate the Bayesian threshold with the letter x*) and compute the probability of error for the two cases of priors $P(\omega_{MAIL}) = P(\omega_{SPAM})$ and $P(\omega_{MAIL}) = 10\, P(\omega_{SPAM})$.

3) Plot the decision regions and the area under the probability density functions corresponding to the probability of error.

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:

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

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

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

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.

b) Specify the loss (cost) matrix that satisfies the above assumption.

Coefficients of the cost matrix?

c) Determine the decision regions that minimize the risk, and compute the related classification error.

Example 2 – a proper choice of the cost (loss) matrix

How to define the cost of reject?

$$\Lambda = \begin{pmatrix} \lambda_R & \lambda_R \\ \lambda_{AA} & \lambda_{AS} \\ \lambda_{SA} & \lambda_{SS} \end{pmatrix} = \begin{pmatrix} \lambda_R & \lambda_R \\ \lambda_C & \lambda_E \\ \lambda_E & \lambda_C \end{pmatrix} = \begin{pmatrix} 0.3 & 0.3 \\ 0 & 1 \\ 1 & 0 \end{pmatrix}$$

The cost of each decision is:

$$\lambda = \begin{cases} \lambda_C = 0, Correct\ action \\ \lambda_R = 0.3, reject \\ \lambda_C = 1, error \end{cases}$$

The threshold *T* of the Chow's rule is:

$$T = \frac{\lambda_E - \lambda_R}{\lambda_E - \lambda_C} = \frac{1 - 0.3}{1} = 0.7$$

ERROR PROBABILITY: (intrusion labeled as normal traffic) + (normal traffic labeled as intrusion)

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

$$= P(\omega_{Intr}) \cdot P\{x \in R_N | \omega_{Intr}\} + P(\omega_N) \cdot P\{x \in R_{Intr} | \omega_N\} =$$

$$= P(\omega_{Intr}) \cdot \int_{-\infty}^{x^*} p(x|\omega_{Intr})\, dx + P(\omega_N) \cdot \int_{x^*}^{\infty} p(x|\omega_N)\, dx =$$

$$= P(\omega_{Intr}) \cdot \int_{-\infty}^{x^*} N(4,1)\, dx + P(\omega_N) \cdot \int_{x^*}^{\infty} N(0,1)\, dx =$$

How to solve this?

$$= 0.0025 + 0.0386 = 0.0411$$

Note: x* = 1.424

Error function erf(x) and complementary error function erfc(x)

$$(1)\ erf(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt$$

$$erfc(x) = \frac{2}{\sqrt{\pi}} \int_x^{\infty} e^{-t^2} dt$$

$$(2)\ erf(x) + erfc(x) = 1$$

More info here: https://stats.stackexchange.com/questions/187828/how-are-the-error-function-and-standard-normal-distribution-function-related
Video explaining the Gaussian-to-erf transformation: https://youtu.be/26QbWYBCw7Y

Exercise 1

Given the following patterns belonging to three different classes A, B, and C

A	1.1 1.3	1.7 1.4	1.2 2.0	1.6 1.9
B	2.7 1.4	2.6 1.2	2.2 2.0	2.2 1.3
C	1.4 2.5	1.2 2.4	1.8 2.6	1.5 2.9

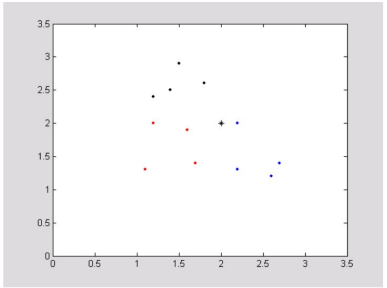
We want to classify the unknown pattern:

x_t=(2; 2)'

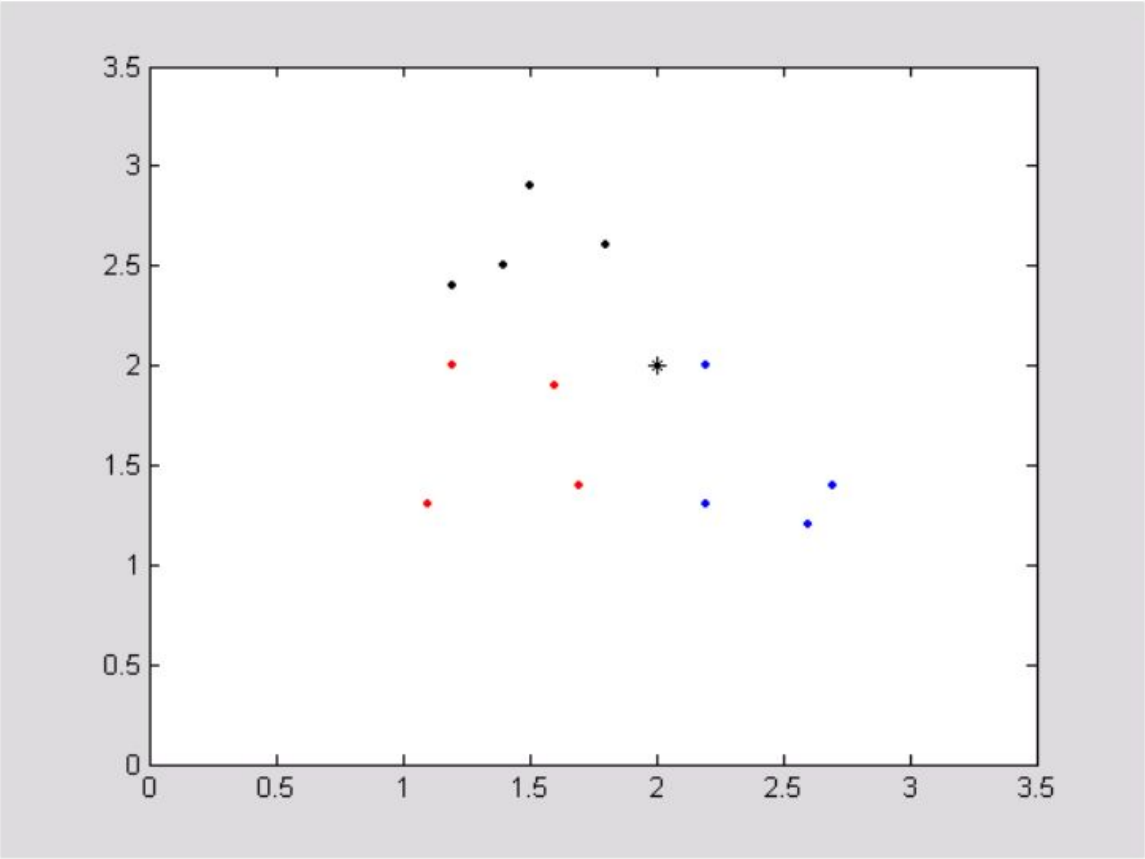
but we do not know from which probability distribution the pattern has been generated. Then, we can use a non-parametric method like the *k*-nn pattern classifier.

A) Classify the pattern x_t with values of *k*= 1, …, 4 using the *Euclidean* and the *Manhattan* distance.
Manhattan distance: |x1 – x2| + |y1 – y2|.

B) Use the "*leave-one-out"* method to select the best value of the "k" parameter between k=1 and k=4, using the *Euclidean* distance. The "*leave-one-out"* method works as follows:
1) Given the training set D with *n* patterns (12 patterns in this exercise)
2) for i=1, …, *n,* use the training set {D -{x_i}} and then classify the pattern x_i left out.
3) Repeat the point (2)
4) Compute the error probability (number of errors for the classifications of the *n* patterns left out)
You should use the above "*leave-one-out"* method for k=1 and k=4 and then select the value of the k parameter that provides the minimum error.



Class A: red points; Class B: blue points; Class C: black points;



Class A: red points; Class B: blue points; Class C: black points;

Links used today

- MachineLearningCheatSheet.pdf
- ML-tutor-02-whiteboard
- https://colab.research.google.com/drive/1pt3eLWoqeRTHHv5IyJeSnLhu7HbX-ikX?usp=sharing