Αναγνώριση Προτύπων

Εργασία 1

Ονοματεπώνυμο: Μωραΐτη Παναγιώτα

ΑΜ: 58054

Πίνακας περιεχομένων

Άσκηση 1

Από τα δεδομένα της άσκησης έχουμε ότι:

n: Normal, s: Spam, m: Malicious

Οι a priori πιθανότητες (εκ των προτέρων) των κλάσεων είναι οι ακόλουθες:

$$P(s) = 0.3$$
$$P(m) = 0.1$$

Ο παρακάτω πίνακας μας δίνει την κατανομή της εκτιμηθείσας επικινδυνότητας για καθεμία από τις κατηγορίες email. Δηλαδή μας δίνει το likelihood $P(\mathrm{D} \mid \omega)$, όπου το ω είναι μία από τις τρεις κατηγορίες email.

	Normal	Spam	Malicious
D=1	0.5	0.05	0.02
D=2	0.23	0.15	0.13
D=3	0.16	0.4	0.15
D=4	0.1	0.3	0.3
D=5	0.01	0.1	0.4

Πολύ εύκολα μπορούμε να υπολογίσουμε την a priori πιθανότητα της κλάσης n.

$$P(\mathrm{n}) = 1 - \mathrm{P(s)} - \mathrm{P(m)} = 1 - 0.3 - 0.1 = 0.6$$

α) Να υπολογίσετε την απόφαση που θα παίρνει το σύστημα για κάθε πιθανή τιμή D με βάση τον κανόνα απόφασης Bayes

Σύμφωνα με τον κανόνα απόφασης Bayes, επιλέγεται η απόφαση που ελαχιστοποιεί το ρίσκο.

$$R(a_i \mid x) = \sum_{\substack{i \neq j, \\ j=1}}^{c} [\lambda(a_i \mid \omega_j) P(\omega_j \mid \mathrm{x})]$$

Δεν έχουμε κάποια πληροφορία για τα κόστη λ, οπότε θεωρούμε μηδενική απώλεια για κάθε σωστή ταξινόμηση και μοναδιαία απώλεια για κάθε λάθος ταξινόμηση.

$$\lambda(a_i \mid \omega_j) = \begin{cases} 0, \text{ if i=j} \\ 1, \text{ if i} \neq \text{ j} \end{cases}$$

Οπότε έχουμε: $R(a_i \mid x) = \sum_{\substack{i \neq j, \\ j=1}}^{c} P(\omega_j \mid \mathrm{x})$

Για να ελαχιστοποιήσουμε το ρίσκο πρέπει να επιλέξουμε την κλάση που μεγιστοποιεί την a posteriori (εκ των υστέρων) πιθανότητα (MAP).

$$P(\omega_i \mid \mathrm{x}) = \frac{P(\mathrm{x} \mid \omega_i)P(\omega_i)}{\mathrm{P(x)}} = \frac{likelihood * prior}{evidence} \quad \text{(posterior)}$$

Οπότε έχουμε:

$$\mathrm{P}(\omega_i \mid \mathrm{D}) = \max[\mathrm{P(n \mid D)},\ \mathrm{P(s \mid D)},\ \mathrm{P(m \mid D)}] = \left[\frac{\mathrm{P(D \mid n)\,P(n)}}{P(\mathrm{D})},\ \frac{\mathrm{P(D \mid s)\,P}(s)}{P(\mathrm{D})},\ \frac{\mathrm{P(D \mid m)\,P}(m)}{P(\mathrm{D})} \right]$$

Κάνουμε τους υπολογισμούς για D=1, 2, 3, 4, 5.

$$\mathrm{P}(\omega_i \mid \mathrm{D}=1) = \max[\mathrm{P(n \mid D}=1),\ \mathrm{P(s \mid D}=1),\ \mathrm{P(m \mid D}=1)] =$$

$$\max\left[\frac{\mathrm{P(D}=1 \mid \mathrm{n})\,\mathrm{P(n)}}{P(\mathrm{D}=1)},\ \frac{\mathrm{P(D}=1 \mid s)\,\mathrm{P}(s)}{P(\mathrm{D}=1)},\ \frac{\mathrm{P(D}=1 \mid m)\,\mathrm{P}(m)}{P(\mathrm{D}=1)} \right] =$$

$$\max\left[\frac{0.5*0.6}{P(\mathrm{D}=1)},\ \frac{0.05*0.3}{P(\mathrm{D}=1)},\ \frac{0.02*0.1}{P(\mathrm{D}=1)} \right] =$$

$$\max\left[\frac{0.3}{P(\mathrm{D}=1)},\ \frac{0.015}{P(\mathrm{D}=1)},\ \frac{0.002}{P(\mathrm{D}=1)} \right] = \frac{0.3}{P(\mathrm{D}=1)},\ \text{οπότε επιλέγουμε n: normal}$$

$$\mathrm{P}(\omega_i \mid \mathrm{D}=2) = \max[\mathrm{P(n \mid D}=2),\ \mathrm{P(s \mid D}=2),\ \mathrm{P(m \mid D}=2)] =$$

$$\max\left[\frac{\mathrm{P(D}=2 \mid \mathrm{n})\,\mathrm{P(n)}}{P(\mathrm{D}=2)},\ \frac{\mathrm{P(D}=2 \mid s)\,\mathrm{P}(s)}{P(\mathrm{D}=2)},\ \frac{\mathrm{P(D}=2 \mid m)\,\mathrm{P}(m)}{P(\mathrm{D}=2)} \right] =$$

$$\max\left[\frac{0.23*0.6}{P(\mathrm{D}=2)},\ \frac{0.15*0.3}{P(\mathrm{D}=2)},\ \frac{0.13*0.1}{P(\mathrm{D}=2)} \right] =$$

$$\max\left[\frac{0.138}{P(\mathrm{D}=2)},\ \frac{0.045}{P(\mathrm{D}=2)},\ \frac{0.013}{P(\mathrm{D}=2)} \right] = \frac{0.138}{P(\mathrm{D}=2)},\ \text{οπότε επιλέγουμε n: normal}$$

$$\mathrm{P}(\omega_i \mid \mathrm{D}=3) = \max[\mathrm{P(n \mid D}=3),\ \mathrm{P(s \mid D}=3),\ \mathrm{P(m \mid D}=3)] =$$

$$\max\left[\frac{\mathrm{P(D}=3 \mid \mathrm{n})\,\mathrm{P(n)}}{P(\mathrm{D}=3)},\ \frac{\mathrm{P(D}=3 \mid s)\,\mathrm{P}(s)}{P(\mathrm{D}=3)},\ \frac{\mathrm{P(D}=3 \mid m)\,\mathrm{P}(m)}{P(\mathrm{D}=3)} \right] =$$

$$\max\left[\frac{0.16*0.6}{P(\mathrm{D}=3)},\ \frac{0.4*0.3}{P(\mathrm{D}=3)},\ \frac{0.15*0.1}{P(\mathrm{D}=3)} \right] =$$

$$\max\left[\frac{0.096}{P(\mathrm{D}=3)},\ \frac{0.12}{P(\mathrm{D}=3)},\ \frac{0.015}{P(\mathrm{D}=3)} \right] = \frac{0.12}{P(\mathrm{D}=3)},\ \text{οπότε επιλέγουμε s: spam}$$

$$P(\omega_i \mid \mathrm{D}=4) = \max[P(\mathrm{n} \mid \mathrm{D}=4),\ P(\mathrm{s} \mid \mathrm{D}=4),\ P(\mathrm{m} \mid \mathrm{D}=4)] =$$

$$\max\left[\frac{P(\mathrm{D}=4 \mid \mathrm{n})P(\mathrm{n})}{P(\mathrm{D}=4)},\ \frac{P(\mathrm{D}=4 \mid s)P(s)}{P(\mathrm{D}=4)},\ \frac{P(\mathrm{D}=4 \mid m)P(m)}{P(\mathrm{D}=4)}\right] =$$

$$\max\left[\frac{0.1*0.6}{P(\mathrm{D}=4)},\ \frac{0.3*0.3}{P(\mathrm{D}=4)},\ \frac{0.3*0.1}{P(\mathrm{D}=4)}\right] =$$

$$\max\left[\frac{0.06}{P(\mathrm{D}=4)},\ \frac{0.09}{P(\mathrm{D}=4)},\ \frac{0.03}{P(\mathrm{D}=4)}\right] = \frac{0.09}{P(\mathrm{D}=4)},$$ οπότε επιλέγουμε s: spam

$$P(\omega_i \mid \mathrm{D}=5) = \max[P(\mathrm{n} \mid \mathrm{D}=5),\ P(\mathrm{s} \mid \mathrm{D}=5),\ P(\mathrm{m} \mid \mathrm{D}=5)] =$$

$$\max\left[\frac{P(\mathrm{D}=5 \mid \mathrm{n})P(\mathrm{n})}{P(\mathrm{D}=5)},\ \frac{P(\mathrm{D}=5 \mid s)P(s)}{P(\mathrm{D}=5)},\ \frac{P(\mathrm{D}=5 \mid m)P(m)}{P(\mathrm{D}=5)}\right] =$$

$$\max\left[\frac{0.01*0.6}{P(\mathrm{D}=5)},\ \frac{0.1*0.3}{P(\mathrm{D}=5)},\ \frac{0.4*0.1}{P(\mathrm{D}=5)}\right] =$$

$$\max\left[\frac{0.006}{P(\mathrm{D}=5)},\ \frac{0.03}{P(\mathrm{D}=5)},\ \frac{0.04}{P(\mathrm{D}=5)}\right] = \frac{0.04}{P(\mathrm{D}=5)},$$ οπότε επιλέγουμε m: malicious

Επίσης $\sum_k P(\mathrm{D}=\mathrm{k} \mid \omega_j) = 1$, δηλαδή αν αθροίσουμε τις στήλες του πίνακα κατακόρυφα για μια συγκεκριμένη κλάση j το άθροισμα είναι μονάδα.

Το P(D) δε μας απασχολεί καθώς για κάθε συγκεκριμένο D είναι το ίδιο για τις τρεις κλάσεις, οπότε μπορούμε να καταλάβουμε ποιος είναι ο μεγαλύτερος όρος χωρίς να υπολογίσουμε τον παρονομαστή του κλάσματος.

Αν θέλαμε να το υπολογίσουμε ισχύει: $\mathrm{P}(\mathrm{D}) = \sum_j P(\mathrm{D} \mid \omega_j)\,\mathrm{P}(\omega_j)$

Για παράδειγμα,

$$\mathrm{P}(\mathrm{D}=1) = \sum_j P(\mathrm{D}=1 \mid \omega_j)\,\mathrm{P}(\omega_j) = P(\mathrm{D}=1 \mid \mathrm{n})\,\mathrm{P}(n) + P(\mathrm{D}=1 \mid s)\,\mathrm{P}(s) + P(\mathrm{D}=1 \mid m)\,\mathrm{P}(m) =$$

$$0.3 + 0.015 + 0.002 = 0.317$$

β) Να υπολογίσετε το ολικό σφάλμα ταξινόμησης

Το σφάλμα για ένα χαρακτηριστικό κ είναι:

$$P_{error,k} = 1 - \max[\mathrm{P}(\mathrm{n} \mid \mathrm{D}=k),\ \mathrm{P}(\mathrm{s} \mid \mathrm{D}=k),\ \mathrm{P}(\mathrm{m} \mid \mathrm{D}=k)]$$

Το συνολικό σφάλμα είναι:

$$P_{error,total} = 1 - \sum_k \max_i (\mathrm{P}(\omega_i, \mathrm{D} = k)) =$$

$$1 - \sum_k \max_i (\mathrm{P}(\omega_i \mid \mathrm{D} = k)\,\mathrm{P}(\mathrm{D} = \mathrm{k})) = 1 - \sum_k \max_i (\mathrm{P}(\mathrm{D} = k \mid \omega_i)\,\mathrm{P}(\omega_i)) =$$

$$1 - \sum_k \left[\max_i \left[\frac{\mathrm{P}(\mathrm{D} = k \mid \omega_i)\,\mathrm{P}(\omega_i)}{P(\mathrm{D} = k)} P(\mathrm{D} = k) \right] \right] =$$

$$1 - \sum_k \max_i \left[\mathrm{P}(\mathrm{D} = k \mid \omega_i)\,\mathrm{P}(\omega_i) \right] =$$

$$1 - (\mathrm{P}(\omega_n \mid \mathrm{D} = 1) + \mathrm{P}(\omega_n \mid \mathrm{D} = 2) + \mathrm{P}(\omega_s \mid \mathrm{D} = 3) + \mathrm{P}(\omega_s \mid \mathrm{D} = 4) + \mathrm{P}(\omega_m \mid \mathrm{D} = 5)) =$$

$$1 - (0.3 + 0.138 + 0.12 + 0.09 + 0.04) = 1 - 0.688 = 0.312$$

γ) Να υπολογίσετε το ολικό σφάλμα ταξινόμησης στην περίπτωση που δεν γνωρίζατε τίποτα για τις a priori πιθανότητες των κλάσεων (τις θεωρούσατε ισοπίθανες)

Αν οι κλάσεις είναι ισοπίθανες έχουμε: $P(n) = P(\mathrm{s}) = \mathrm{P}(\mathrm{m}) = \mathrm{c}$, οπότε:

$$P(n) + P(\mathrm{s}) + \mathrm{P}(\mathrm{m}) = 1 \Rightarrow 3c = 1 \Rightarrow c = 1/3$$

$$\mathrm{P}(\omega_i \mid \mathrm{D} = 1) = \max[\mathrm{P}(\mathrm{n} \mid \mathrm{D} = 1),\ \mathrm{P}(\mathrm{s} \mid \mathrm{D} = 1),\ \mathrm{P}(\mathrm{m} \mid \mathrm{D} = 1)] =$$

$$\max\left[\frac{\mathrm{P}(\mathrm{D} = 1 \mid \mathrm{n})\,\mathrm{P}(\mathrm{n})}{P(\mathrm{D} = 1)},\ \frac{\mathrm{P}(\mathrm{D} = 1 \mid s)\,\mathrm{P}(s)}{P(\mathrm{D} = 1)},\ \frac{\mathrm{P}(\mathrm{D} = 1 \mid m)\,\mathrm{P}(m)}{P(\mathrm{D} = 1)} \right] =$$

$$\max\left[\frac{0.5c}{P(\mathrm{D} = 1)},\ \frac{0.05c}{P(\mathrm{D} = 1)},\ \frac{0.02c}{P(\mathrm{D} = 1)} \right] = \frac{0.5 * \frac{1}{3}}{P(\mathrm{D} = 1)} = \frac{1}{6P(\mathrm{D} = 1)} \simeq \frac{0.1667}{P(\mathrm{D} = 1)},$$ οπότε επιλέγουμε n: normal

$$\mathrm{P}(\omega_i \mid \mathrm{D} = 2) = \max[\mathrm{P}(\mathrm{n} \mid \mathrm{D} = 2),\ \mathrm{P}(\mathrm{s} \mid \mathrm{D} = 2),\ \mathrm{P}(\mathrm{m} \mid \mathrm{D} = 2)] =$$

$$\max\left[\frac{\mathrm{P}(\mathrm{D} = 2 \mid \mathrm{n})\,\mathrm{P}(\mathrm{n})}{P(\mathrm{D} = 2)},\ \frac{\mathrm{P}(\mathrm{D} = 2 \mid s)\,\mathrm{P}(s)}{P(\mathrm{D} = 2)},\ \frac{\mathrm{P}(\mathrm{D} = 2 \mid m)\,\mathrm{P}(m)}{P(\mathrm{D} = 2)} \right] =$$

$$\max\left[\frac{0.23c}{P(\mathrm{D} = 2)},\ \frac{0.15c}{P(\mathrm{D} = 2)},\ \frac{0.13c}{P(\mathrm{D} = 2)} \right] = \frac{0.23 * \frac{1}{3}}{P(\mathrm{D} = 2)} \simeq \frac{0.0767}{P(\mathrm{D} = 2)},$$ οπότε επιλέγουμε n: normal

$$\mathrm{P}(\omega_i \mid \mathrm{D} = 3) = \max[\mathrm{P}(\mathrm{n} \mid \mathrm{D} = 3),\ \mathrm{P}(\mathrm{s} \mid \mathrm{D} = 3),\ \mathrm{P}(\mathrm{m} \mid \mathrm{D} = 3)] =$$

$$\max\left[\frac{\mathrm{P}(\mathrm{D} = 3 \mid \mathrm{n})\,\mathrm{P}(\mathrm{n})}{P(\mathrm{D} = 3)},\ \frac{\mathrm{P}(\mathrm{D} = 3 \mid s)\,\mathrm{P}(s)}{P(\mathrm{D} = 3)},\ \frac{\mathrm{P}(\mathrm{D} = 3 \mid m)\,\mathrm{P}(m)}{P(\mathrm{D} = 3)} \right] =$$

$$\max\left[\frac{0.16c}{P(\mathrm{D} = 3)},\ \frac{0.4c}{P(\mathrm{D} = 3)},\ \frac{0.15c}{P(\mathrm{D} = 3)} \right] = \frac{0.4 * \frac{1}{3}}{P(\mathrm{D} = 3)} \simeq \frac{0.1333}{P(\mathrm{D} = 3)},$$ οπότε επιλέγουμε s: spam

$$\mathrm{P}(\omega_i \mid \mathrm{D}=4) = \max[\mathrm{P}(\mathrm{n} \mid \mathrm{D}=4),\ \mathrm{P}(\mathrm{s} \mid \mathrm{D}=4),\ \mathrm{P}(\mathrm{m} \mid \mathrm{D}=4)] =$$

$$\max\left[\frac{\mathrm{P}(\mathrm{D}=4 \mid \mathrm{n})\,\mathrm{P}(\mathrm{n})}{P(\mathrm{D}=4)},\ \frac{\mathrm{P}(\mathrm{D}=4 \mid s)\,\mathrm{P}(s)}{P(\mathrm{D}=4)},\ \frac{\mathrm{P}(\mathrm{D}=4 \mid m)\,\mathrm{P}(m)}{P(\mathrm{D}=4)}\right] =$$

$$\max\left[\frac{0.1c}{P(\mathrm{D}=4)},\ \frac{0.3c}{P(\mathrm{D}=4)},\ \frac{0.3c}{P(\mathrm{D}=4)}\right] = \frac{0.3*\frac{1}{3}}{P(\mathrm{D}=4)} \simeq \frac{0.1}{P(\mathrm{D}=4)},$$

οπότε δεν ξέρουμε ποια από τις κλάσεις s, m να επιλέξουμε.

$$\mathrm{P}(\omega_i \mid \mathrm{D}=5) = \max[\mathrm{P}(\mathrm{n} \mid \mathrm{D}=5),\ \mathrm{P}(\mathrm{s} \mid \mathrm{D}=5),\ \mathrm{P}(\mathrm{m} \mid \mathrm{D}=5)] =$$

$$\max\left[\frac{\mathrm{P}(\mathrm{D}=5 \mid \mathrm{n})\,\mathrm{P}(\mathrm{n})}{P(\mathrm{D}=5)},\ \frac{\mathrm{P}(\mathrm{D}=5 \mid s)\,\mathrm{P}(s)}{P(\mathrm{D}=5)},\ \frac{\mathrm{P}(\mathrm{D}=5 \mid m)\,\mathrm{P}(m)}{P(\mathrm{D}=5)}\right] =$$

$$\max\left[\frac{0.01c}{P(\mathrm{D}=5)},\ \frac{0.1c}{P(\mathrm{D}=5)},\ \frac{0.4c}{P(\mathrm{D}=5)}\right] = \frac{0.4*\frac{1}{3}}{P(\mathrm{D}=5)} \simeq \frac{0.1333}{P(\mathrm{D}=5)},$$ οπότε επιλέγουμε m: malicious

Το συνολικό σφάλμα είναι:

$$P_{error,total} = 1 - \sum_k \max(\mathrm{P}(\omega_i \mid \mathrm{D}=k)\,\mathrm{P}(\mathrm{D}=k)) =$$

$$1 - (\mathrm{P}(\omega_n \mid \mathrm{D}=1) + \mathrm{P}(\omega_n \mid \mathrm{D}=2) + \mathrm{P}(\omega_s \mid \mathrm{D}=3) + \mathrm{P}(\omega_s \mid \mathrm{D}=4) + \mathrm{P}(\omega_m \mid \mathrm{D}=5)) =$$

$$1 - (0.5 + 0.23 + 0.4 + 0.3 + 0.4) * \frac{1}{3} = 1 - 1.83 * \frac{1}{3} = 1 - 0.61 = 0.39$$

Παρατηρούμε ότι σε σχέση με το ερώτημα β το σφάλμα αυξάνεται λίγο. Αυτό μπορεί να οφείλεται στην αβεβαιότητα που έχουμε για τις a priori πιθανότητες (τις θεωρήσαμε ίσες ενώ αυτό μπορεί να μην ισχύει).

Άσκηση 2

Έχουμε τα παρακάτω δεδομένα.

$$\Omega = \{\omega_1, \omega_2\}$$

$$P(\mathrm{x} \mid \omega_1) = \mathrm{N}(2, 0.5)$$
$$P(\mathrm{x} \mid \omega_2) = \mathrm{N}(1.8, 0.2)$$
$$P(\omega_1) = 1/4$$
$$\lambda = \begin{pmatrix} 0 & 1 \\ 3 & 0 \end{pmatrix}$$

Εύκολα μπορούμε να βρούμε:

$$P(\omega_1) + P(\omega_2) = 1 \Rightarrow P(\omega_2) = 1 - 1/4 = 3/4$$

α) Να βρεθεί η βέλτιστη λύση (κανόνας απόφασης) και να υπολογισθεί το ολικό κόστος

Η απόφαση κατά Bayes επιλέγει τη δράση που ελαχιστοποιεί το ρίσκο.

$$R(\alpha_1 \mid \mathrm{x}) = \lambda_{11}\,\mathrm{P}(\omega_1 \mid \mathrm{x}) + \lambda_{12}\,\mathrm{P}(\omega_2 \mid \mathrm{x})$$
$$R(\alpha 2 \mid \mathrm{x}) = \lambda_{21}\,\mathrm{P}(\omega_1 \mid \mathrm{x}) + \lambda_{22}\,\mathrm{P}(\omega_2 \mid \mathrm{x})$$

Αν ισχύει $R(\alpha_1 \mid \mathrm{x}) < R(\alpha_2 \mid \mathrm{x})$, Το κριτήριο δίνει την απόφαση ω_1

Έχουμε:

$$R(\alpha_1 \mid \mathrm{x}) < R(\alpha_2 \mid \mathrm{x}) =>$$
$$\lambda_{11}\,\mathrm{P}(\omega_1 \mid \mathrm{x}) + \lambda_{12}\,\mathrm{P}(\omega_2 \mid \mathrm{x}) < \lambda_{21}\,\mathrm{P}(\omega_1 \mid \mathrm{x}) + \lambda_{22}\,\mathrm{P}(\omega_2 \mid \mathrm{x}) =>$$
$$\mathrm{P}(\omega_1 \mid \mathrm{x})(\lambda_{11} - \lambda_{21}) < \mathrm{P}(\omega_2 \mid \mathrm{x})(\lambda_{22} - \lambda_{12}) =>$$

$$\frac{\mathrm{P}(\mathrm{x} \mid \omega_1)\,\mathrm{P}(\omega_1)(\lambda_{21} - \lambda_{11})}{P(\mathrm{x})} > \frac{\mathrm{P}(x \mid \omega_2)\,\mathrm{P}(\omega_2)(\lambda_{12} - \lambda_{22})}{P(\mathrm{x})} =>$$
$$\frac{\mathrm{P}(\mathrm{x} \mid \omega_1)}{\mathrm{P}(x \mid \omega_2)} > \frac{\lambda_{12} - \lambda_{22}}{\lambda_{21} - \lambda_{11}}\,\frac{\mathrm{P}(\omega_2)}{\mathrm{P}(\omega_1)} =>$$

$$\frac{\mathrm{P}(\mathrm{x} \mid \omega_1)}{\mathrm{P}(x \mid \omega_2)} > \frac{1-0}{3-0}\,\frac{0.75}{0.25} =>$$
$$\frac{\mathrm{P}(\mathrm{x} \mid \omega_1)}{\mathrm{P}(x \mid \omega_2)} > \frac{0.75}{3*0.25} =>$$
$$\frac{\mathrm{P}(\mathrm{x} \mid \omega_1)}{\mathrm{P}(x \mid \omega_2)} > \frac{0.75}{0.75} = 1 =>$$
$$\frac{\mathrm{P}(\mathrm{x} \mid \omega_1)}{\mathrm{P}(x \mid \omega_2)} > 1$$

Γνωρίζουμε ότι για την κανονική κατανομή (Gauss) ισχύει: $\mathrm{N}(\mu, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}}\, e^{-\frac{(\mathrm{x}-\mu)^2}{2\sigma^2}}$

Αντικαθιστώ τις εκ των προτέρων πιθανότητες.

$$\frac{\frac{1}{\sigma_1\sqrt{2\pi}}e^{-\frac{(x-\mu_1)^2}{2\sigma_1^2}}}{\frac{1}{\sigma_2\sqrt{2\pi}}e^{-\frac{(x-\mu_2)^2}{2\sigma_2^2}}} > 1 \Rightarrow \frac{\sigma_2}{\sigma_1}e^{\frac{(x-\mu_2)^2}{2\sigma_2^2}-\frac{(x-\mu_1)^2}{2\sigma_1^2}} > 1 \Rightarrow \ln\left(\frac{\sigma_2}{\sigma_1}e^{\frac{(x-\mu_2)^2}{2\sigma_2^2}-\frac{(x-\mu_1)^2}{2\sigma_1^2}}\right) > \ln(1) \Rightarrow$$

$$\ln\left(\frac{\sigma_2}{\sigma_1}\right)+\ln\left(e^{\frac{(x-\mu_2)^2}{2\sigma_2^2}-\frac{(x-\mu_1)^2}{2\sigma_1^2}}\right) > 0 \Rightarrow \ln\left(\frac{\sigma_2}{\sigma_1}\right)+\frac{(x-\mu_2)^2}{2\sigma_2^2}-\frac{(x-\mu_1)^2}{2\sigma_1^2} > 0 \Rightarrow$$

$$\ln\left(\frac{\sqrt{0.2}}{\sqrt{0.5}}\right)+\frac{(x-1.8)^2}{2(\sqrt{0.2})^2}-\frac{(x-2)^2}{2(\sqrt{0.5})^2} > 0 \Rightarrow \ln\left(\frac{\sqrt{0.2}}{\sqrt{0.5}}\right)+\frac{(x-1.8)^2}{2*0.2}-\frac{(x-2)^2}{2*0.5} > 0 \Rightarrow$$

$$\ln\left(\frac{\sqrt{0.2}}{\sqrt{0.5}}\right)+\frac{(x-1.8)^2}{0.4}-\frac{(x-2)^2}{1} > 0 \Rightarrow 0.4*\ln\left(\frac{\sqrt{0.2}}{\sqrt{0.5}}\right)+(x-1.8)^2-0.4*(x-2)^2 > 0 \Rightarrow$$

$$0.4*(-0.458145)+x^2-2*1.8x+1.8^2-0.4*(x^2-2*2x+2^2) > 0 \Rightarrow$$

$$-0.183258+x^2-3.6x+3.24-0.4x^2+1.6x-1.6) > 0 \Rightarrow$$

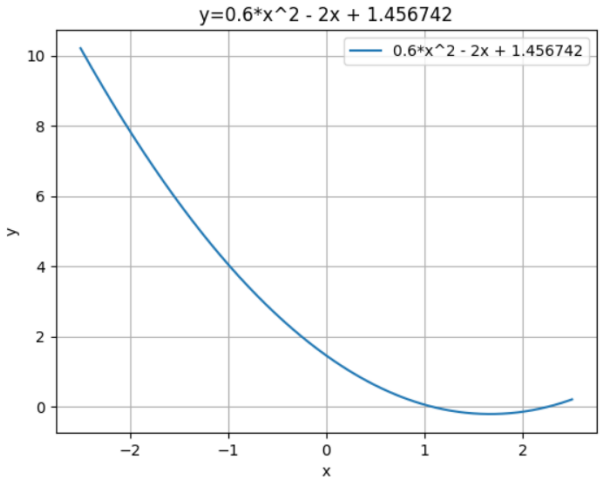
$$0.6x^2-2x+1.64-0.183258 > 0 \Rightarrow$$

$$0.6x^2-2x+1.456742 > 0$$

Οι ρίζες τις παραπάνω ανίσωσης είναι x1=2.25817 και x2=1.07516.

```
[ ]  roots = np.roots([0.6, -2, 1.456742])
     print("Roots:", roots)

     Roots: [2.25816852 1.07516481]
```

Παρακάτω φαίνεται και το διάγραμμα της συνάρτησης. Εύκολα μπορούμε να δούμε ότι ανάμεσα στις ρίζες η συνάρτηση είναι αρνητική και σε όλα τα υπόλοιπα διαστήματα είναι θετική (το γνωρίζουμε και από τη θεωρία). Οπότε η παραπάνω ανισότητα ισχύει σε όλα τα διαστήματα που η συνάρτηση είναι θετική.



Οπότε για $x \in (-\infty, 1.07516) \cup (2.25817, +\infty)$ επιλέγεται η κλάση ω_1, ενώ για

$x \in (1.07516, 2.25817)$ επιλέγεται η κλάση ω_2.

Το συνολικό κόστος είναι:

$$Cost = \int_{R1} R(\alpha_1 \mid \mathrm{x})P(\mathrm{x})\,\mathrm{dx} + \int_{R2} R(\alpha_2 \mid \mathrm{x})P(\mathrm{x})\,\mathrm{dx}$$, βιβλίο Duda σελ. 11

Οπότε έχουμε:

$$Cost = \int_{R1} R(\alpha_1 \mid \mathrm{x})P(\mathrm{x})\,\mathrm{dx} + \int_{R2} R(\alpha_2 \mid \mathrm{x})P(\mathrm{x})\,\mathrm{dx} =$$

$$\int_{R1} (\lambda_{11}\,\mathrm{P}(\omega_1 \mid \mathrm{x}) + \lambda_{12}\,\mathrm{P}(\omega_2 \mid \mathrm{x}))P(\mathrm{x})\,\mathrm{dx} + \int_{R2} (\lambda_{21}\,\mathrm{P}(\omega_1 \mid \mathrm{x}) + \lambda_{22}\,\mathrm{P}(\omega_2 \mid \mathrm{x}))P(\mathrm{x})\,\mathrm{dx} =$$

$$\int_{R1} (\lambda_{11}\frac{\mathrm{P}(x \mid \omega_1)\,\mathrm{P}(\omega_1)}{P(\mathrm{x})} + \lambda_{12}\frac{\mathrm{P}(x \mid \omega_2)\,\mathrm{P}(\omega_2)}{P(\mathrm{x})})P(\mathrm{x})\,\mathrm{dx} + \int_{R2} (\lambda_{21}\frac{\mathrm{P}(x \mid \omega_1)\,\mathrm{P}(\omega_1)}{P(\mathrm{x})} + \lambda_{22}\frac{\mathrm{P}(x \mid \omega_2)\,\mathrm{P}(\omega_2)}{P(\mathrm{x})})P(\mathrm{x})\,\mathrm{dx} =$$

$$\int_{R1} (\lambda_{11}\,\mathrm{P}(x \mid \omega_1)\,\mathrm{P}(\omega_1) + \lambda_{12}\,\mathrm{P}(x \mid \omega_2)\,\mathrm{P}(\omega_2))\,\mathrm{dx} + \int_{R2} (\lambda_{21}\,\mathrm{P}(x \mid \omega_1)\,\mathrm{P}(\omega_1) + \lambda_{22}\,\mathrm{P}(x \mid \omega_2)\,\mathrm{P}(\omega_2))\,\mathrm{dx} =$$

$$\int_{R1} \lambda_{11}\,\mathrm{P}(x \mid \omega_1)\,\mathrm{P}(\omega_1)\,\mathrm{dx} + \int_{R1} \lambda_{12}\,\mathrm{P}(x \mid \omega_2)\,\mathrm{P}(\omega_2)\,\mathrm{dx} + \int_{R2} \lambda_{21}\,\mathrm{P}(x \mid \omega_1)\,\mathrm{P}(\omega_1)\,\mathrm{dx} + \int_{R2} \lambda_{22}\,\mathrm{P}(x \mid \omega_2)\,\mathrm{P}(\omega_2)\,\mathrm{dx} =$$

$$\mathrm{P}(\omega_1)\left[\lambda_{11}\int_{R1} \mathrm{P}(x \mid \omega_1)\,\mathrm{dx} + \lambda_{21}\int_{R2} \mathrm{P}(x \mid \omega_1)\,\mathrm{dx}\right] + \mathrm{P}(\omega_2)\left[\lambda_{12}\int_{R1} \mathrm{P}(x \mid \omega_2)\,\mathrm{dx} + \lambda_{22}\int_{R2} \mathrm{P}(x \mid \omega_2)\,\mathrm{dx}\right] =$$

$$\frac{1}{4}\left[0*\int_{R1} \mathrm{P}(x \mid \omega_1)\,\mathrm{dx} + 3*\int_{R2} \mathrm{P}(x \mid \omega_1)\,\mathrm{dx}\right] + \frac{3}{4}\left[1*\int_{R1} \mathrm{P}(x \mid \omega_2)\,\mathrm{dx} + 0*\int_{R2} \mathrm{P}(x \mid \omega_2)\,\mathrm{dx}\right] =$$

$$\frac{3}{4}\int_{R2} \mathrm{P}(x \mid \omega_1)\,\mathrm{dx} + \frac{3}{4}\int_{R1} \mathrm{P}(x \mid \omega_2)\,\mathrm{dx}$$

Για να επιλύσουμε τα ολοκληρώματα μπορούμε να χρησιμοποιήσουμε τις ιδιότητες της κανονικής κατανομής που φαίνονται και στην παρακάτω εικόνα.

If X is a normal random variable with mean μ and variance σ^2, i.e, $X \sim N(\mu, \sigma^2)$, then

$$f_X(x) = \frac{1}{\sigma\sqrt{2\pi}}\exp\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\},$$

$$F_X(x) = P(X \le x) = \Phi\left(\frac{x-\mu}{\sigma}\right),$$

$$P(a < X \le b) = \Phi\left(\frac{b-\mu}{\sigma}\right) - \Phi\left(\frac{a-\mu}{\sigma}\right).$$

Η αθροιστική συνάρτηση πιθανότητας (Cumulative Density Function, CDF) της κανονικής κατανομής φαίνεται στην παρακάτω εικόνα. Εκτός από τον παρακάτω τύπο μπορούμε να υπολογίσουμε κάποια τιμή της συνάρτησης αυτής χρησιμοποιώντας γνωστούς πίνακες.

The CDF of the standard normal distribution is denoted by the Φ function:

$$\Phi(x) = P(Z \leq x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} \exp\left\{-\frac{u^2}{2}\right\} du.$$

As we will see in a moment, the CDF of any normal random variable can be written in terms of the Φ function, so the Φ function is widely used in probability. Figure 4.7 shows the Φ function.

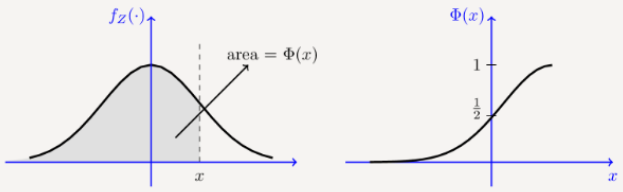


Fig.4.7 - The Φ function (CDF of standard normal).

Με τη βοήθεια των παραπάνω ιδιοτήτων έχουμε:

$$\int_{R2} \mathrm{P}(x \mid \omega_1)\,\mathrm{dx} = P(1.07516 < \mathrm{x} < 2.25817 \mid \omega_1) = \Phi\left(\frac{2.25817 - 2}{\sqrt{0.5}}\right) - \Phi\left(\frac{1.07516 - 2}{\sqrt{0.5}}\right) =$$

$$\Phi(0.3651) - \Phi(-1.3079) = 0.6425 - 0.0955 = 0.547$$

$$\int_{R2} \mathrm{P}(x \mid \omega_2)\,\mathrm{dx} = P(1.07516 < \mathrm{x} < 2.25817 \mid \omega_2) = \Phi\left(\frac{2.25817 - 1.8}{\sqrt{0.2}}\right) - \Phi\left(\frac{1.07516 - 1.8}{\sqrt{0.2}}\right) =$$

$$\Phi(1.0245) - \Phi(-1.6208) = 0.8472 - 0.0525 = 0.7947$$

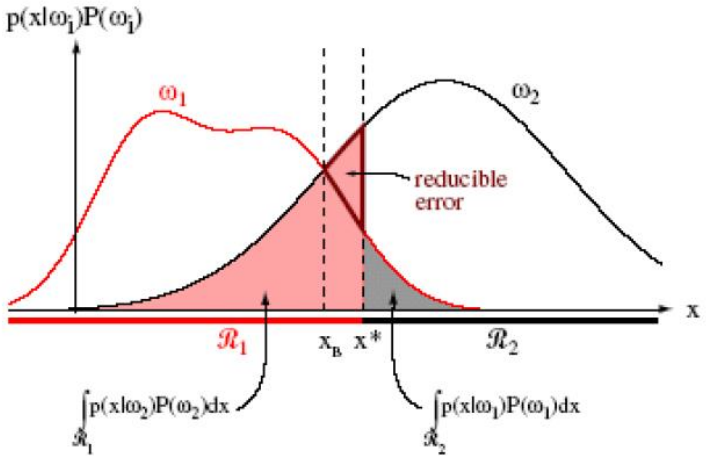
Τη CDF της κανονικής κατανομής μπορούμε είτε να την υπολογίσουμε με κάποιο calculator ή με τη βοήθεια του Matlab, όπως φαίνεται και παρακάτω.

```
Command Window

>> normcdf(0.3651)

ans =

    0.6425

>> normcdf(-1.3079)

ans =

    0.0955
```

```
>> normcdf(1.0245)

ans =

    0.8472

>> normcdf(-1.6208)

ans =

    0.0525
```

Από το παρακάτω σχήμα παρατηρούμε ότι:

$$\int_{R1} \mathrm{P}(x \mid \omega_2)\,\mathrm{dx} = P(\mathrm{x} < 1.07516 \cup x > 2.25817 \mid \omega_2) = 1 - \int_{R2} \mathrm{P}(x \mid \omega_2)\,\mathrm{dx} = 1 - 0.7947 = 0.2053$$



Άρα, το κόστος είναι:

$$Cost = \frac{3}{4}\int_{R2} \mathrm{P}(x \mid \omega_1)\,\mathrm{dx} + \frac{3}{4}\int_{R1} \mathrm{P}(x \mid \omega_2)\,\mathrm{dx} = \frac{3}{4} * 0.547 + \frac{3}{4} * 0.2053 = 0.75 * 0.7523 = 0.564$$

β) Να προσομοιωθεί η διαδικασία υπολογιστικά, δημιουργώντας τυχαία δείγματα που ακολουθούν την κανονική κατανομή και να εκτιμηθεί αριθμητικά το κόστος από την λύση του ερωτήματος α

Με τον παρακάτω κώδικα παράγω τυχαία δείγματα που ακολουθούν την κανονική κατανομή, με τη βοήθεια της συνάρτησης **np.random.normal**. Ο αριθμός των δειγμάτων μπορεί να ρυθμιστεί με τη μεταβλητή **number_of_samples**. Το seed το ρύθμισα έτσι ώστε να έχω κάθε φορά τα ίδια αποτελέσματα, επειδή τα δείγματα παράγονται με τυχαίο τρόπο αν δε ρυθμίσω το seed τότε τα αποτελέσματά μου θα διαφέρουν λίγο κάθε φορά που θα τρέχω το πρόγραμμα.

Προσοχή πρέπει να δοθεί στο γεγονός ότι τα παραγόμενα δείγματα για τις δύο κλάσεις πρέπει να ακολουθούν τις εκ των προτέρων πιθανότητες. $P(\omega_1) = 1/4$ και $P(\omega_2) = 3/4$

Στη συνέχεια με βάση τις ρίζες (όρια απόφασης) που έχουμε βρει μπορούμε να δούμε πόσα δείγματα κατηγοριοποιούνται σωστά στις δύο κλάσεις και πόσα κατηγοριοποιούνται λάθος και βρίσκουμε το αντίστοιχο ποσοστό τους. Τέλος, μπορούμε να υπολογίσουμε το συνολικό κόστος, το οποίο πρέπει να είναι σχεδόν ίσο με την τιμή που υπολογίσαμε στο ερώτημα α.

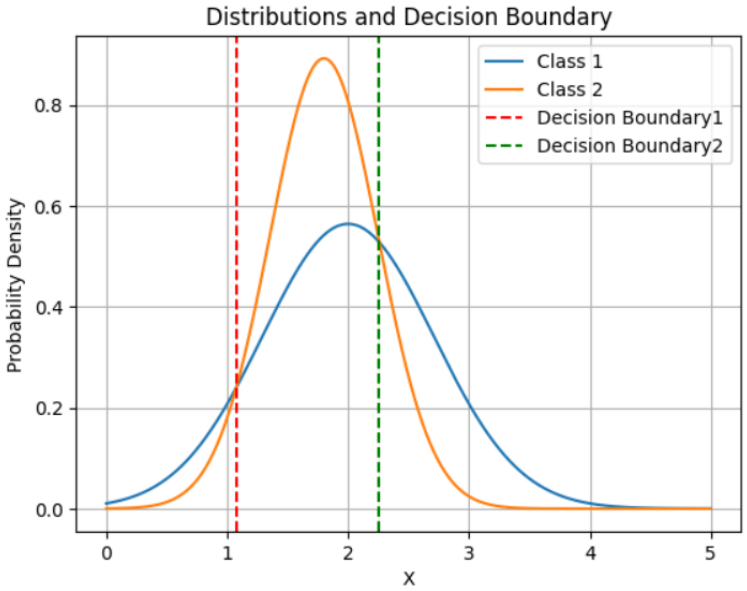
Τα αποτελέσματα της εκτύπωσης φαίνονται στην διπλανή εικόνα. Το συνολικό κόστος είναι 0.564, όσο δηλαδή βρήκαμε και στο ερώτημα α.

```
w1 classified as w1: 0.452
w1 misclassified as w2: 0.548
w2 misclassified as w1: 0.205
w2 classified as w2 0.795
The cost is: 0.564
```

Παρακάτω φαίνεται ο κώδικας σε python που προσομοιώνει τη διαδικασία.

```
rng = np.random.default_rng(seed=6)
number_of_samples = 100000 # Choose the number of samples

p_w1, p_w2 = 1/4, 3/4
w1_samples = int(p_w1 * number_of_samples)
w2_samples = int(p_w2 * number_of_samples)
l11, l12, l21, l22 = 0, 1, 3, 0
m1, m2 = 2, 1.8
s1, s2 = np.sqrt(0.5), np.sqrt(0.2)

# Create samples that follow normal distribution
class1 = rng.normal(m1, s1, w1_samples)
class2 = rng.normal(m2, s2, w2_samples)

# Find where the samples are classified
classify_w1_to_w2, classify_w1_to_w1, classify_w2_to_w1, classify_w2_to_w2 = 0, 0, 0,
0
root1 = 1.07516481
root2 = 2.25816852

for sample in class1:
  if root1 < sample < root2:
    classify_w1_to_w2 += 1
  else:
    classify_w1_to_w1 += 1

for sample in class2:
  if root1 < sample < root2:
    classify_w2_to_w2 += 1
  else:
    classify_w2_to_w1 += 1

# Find the probabilities (percentage)
classify_w1_to_w1 = classify_w1_to_w1 / w1_samples
classify_w1_to_w2 = classify_w1_to_w2 / w1_samples
classify_w2_to_w1 = classify_w2_to_w1 / w2_samples
classify_w2_to_w2 = classify_w2_to_w2 / w2_samples

# Calculate the cost
cost = p_w1 * ((l11 * classify_w1_to_w1) + (l21 * classify_w1_to_w2)) + p_w2 * ((l12 *
classify_w2_to_w1) + (l22 * classify_w2_to_w2))
```

Μπορούμε επίσης να κάνουμε ένα διάγραμμα με τις δύο κατανομές και τα όρια απόφασης. Από το διάγραμμα καταλαβαίνουμε και γραφικά ότι τα λάθη στην ταξινόμηση οφείλονται στην αλληλοεπικάλυψη των δύο κατανομών.



Άσκηση 3

Με στόχο την ταξινόμηση δεδομένων με Bayesian κριτήρια, να εκτελεστούν τα παρακάτω βήματα:

α) Να υλοποιήσετε τρεις συναρτήσεις στη γλώσσα προγραμματισμού της αρεσκείας σας που να υπολογίζουν:

i. Την τιμή της συνάρτησης διάκρισης για μια δεδομένη κανονική κατανομή N (μ,Σ) d διαστάσεων και εκ των προτέρων πιθανότητα P(ωi)

Η συνάρτηση διάκρισης είναι της μορφής:

$$g_i(\mathbf{x}) = -\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_i)^T \boldsymbol{\Sigma}_i^{-1}(\mathbf{x}-\boldsymbol{\mu}_i) - \frac{d}{2}\ln(2\pi) - \frac{1}{2}\ln(|\boldsymbol{\Sigma}_i|) + \ln(P(\omega_i))$$

Παρακάτω φαίνεται η συνάρτηση που επιστρέφει την τιμή της συνάρτησης διάκρισης για μια δεδομένη κανονική κατανομή.

```
def calculate_discriminant_function(x, m, s, d, prior):
  if d > 1: # x is a vector
    S_inv = np.linalg.inv(s)
    det_S = round(abs(np.linalg.det(s)), 5)
    return -(1/2)*np.linalg.multi_dot([(x - m).T, S_inv, (x - m)]) -
(d/2)*np.log(2*np.pi) - (1/2)*np.log(det_S) + np.log(prior)
  else: # x is a number
    S_inv = 1/s
    det_S = abs(s)
    return -(1/2)*(x - m)*S_inv*(x - m) - (d/2)*np.log(2*np.pi) - (1/2)*np.log(det_S)
+ np.log(prior)
```

Οι συναρτήσεις **numpy.linalg.inv** και **numpy.linalg.multi_dot** περιμένουν σαν όρισμα ένα πίνακα, με αποτέλεσμα να δημιουργείται πρόβλημα όταν έχουμε διάσταση ένα, δηλαδή όταν έχουμε σαν όρισμα έναν αριθμό (error 0-dimensional array given. Array must be at least two-dimensional). Οπότε, διαχωρίζουμε τις περιπτώσεις σε d=1 και d>1. Για d=1 οι μόνες διαφορές είναι ότι κάποιες πράξεις γίνονται πιο εύκολα.

ii. Την Ευκλείδεια απόσταση μεταξύ δύο αυθαίρετων σημείων x1 και x2 σε χώρο d διαστάσεων.

Η Ευκλείδεια απόσταση (L2) για δύο διανύσματα δίνεται από τον παρακάτω τύπο:

$$d(x, y) = \sqrt{\sum_{i=1}^{n}(y_i - x_i)^2}$$

Σε περίπτωση που έχουμε d=1, τότε η ρίζα με το τετράγωνο απλοποιείται και μας μένει το απόλυτο της διαφοράς των δύο σημείων.

Παρακάτω φαίνεται η συνάρτηση που επιστρέφει την Ευκλείδεια απόσταση μεταξύ δύο αυθαίρετων σημείων x1 και x2.

```
def euclidian_distance(x1, x2, d):
  if d > 1: # x is a vector
    return np.sqrt(np.sum(np.square(x1 - x2)))
  else: # x is a number
    return abs(x1 - x2)
```

iii. Την απόστασης Mahalanobis μεταξύ του μέσου μ και ενός αυθαίρετου σημείου x σε χώρο d διαστάσεων, δεδομένου του πίνακα συνδιασποράς Σ

Η Mahalanobis απόσταση για δύο διανύσματα δίνεται από τον παρακάτω τύπο:

$$d_m = ((\underline{x} - \underline{\mu}_i)^T \Sigma^{-1} (\underline{x} - \underline{\mu}_i))^{\frac{1}{2}}$$

Παρακάτω φαίνεται η συνάρτηση που επιστρέφει την Mahalanobis απόσταση του μέσου μ και ενός αυθαίρετου σημείου x, δεδομένου του πίνακα συνδιασποράς Σ.

```
def mahalanobis_distance(x, m, s, d):
  if d > 1: # x is a vector
    S_inv = np.linalg.inv(s)
    return np.sqrt(np.linalg.multi_dot([(x - m).T, S_inv, (x - m)]))
  else: # x is a number
    S_inv = 1/s
    return np.sqrt((x - m)*S_inv*(x - m))
```

β) Υποθέτοντας πως οι υποκείμενες κατανομές των κλάσεων που δίνονται στο αρχείο δεδομένων είναι Gaussian, να κάνετε εκτίμηση των παραμέτρων των κατανομών με τη μέθοδο της μεγίστης πιθανοφάνειας (MLE) για τις περιπτώσεις που χρησιμοποιείται μόνο το πρώτο χαρακτηριστικό x1, το πρώτο και το δεύτερο χαρακτηριστικό (x1 και x2) και τέλος για όλα τα χαρακτηριστικά

Για την εκτίμηση των παραμέτρων των κατανομών θα χρησιμοποιήσουμε την τεχνική Maximum Likelihood Estimation. Για την κανονική κατανομή έχουμε:

$$\hat{\mu} = \frac{1}{n}\sum_{k=1}^{n} x_k \quad ; \quad \hat{\sigma}^2 = \frac{1}{n}\sum_{k=1}^{n}(x_k - \hat{\mu})^2$$

Αν έχω περισσότερα από ένα χαρακτηριστικά, τότε αντί για διακύμανση (variance), έχω τον πίνακας συνδιασποράς που υπολογίζεται από τον ακόλουθο τύπο:

$$\underbrace{\mathbf{C} = \frac{1}{n-1}\sum_{k=1}^{n}(\mathbf{x}_k - \boldsymbol{\mu})(\mathbf{x}_k - \hat{\boldsymbol{\mu}})^t}_{\textit{Sample covariance matrix}}$$

Η συνάρτηση **np.cov**, χρησιμοποιεί τον unbiased τύπο (διαίρεση με N-1 αντί για N).

Με τη χρήση των παραπάνω τύπων έχουμε:

```
mean_class1_x1 = np.mean(class1['x1'])
covariance_class1_x1 = np.cov(class1['x1'])

mean_class2_x1 = np.mean(class2['x1'])
covariance_class2_x1 = np.cov(class2['x1'])

mean_class3_x1 = np.mean(class3['x1'])
covariance_class3_x1 = np.cov(class3['x1'])

print(f"Mean for x1, Class 1: {mean_class1_x1}")
print(f"Mean for x1, Class 2: {mean_class2_x1}")
print(f"Mean for x1, Class 3: {mean_class3_x1}\n")

print(f"Covariance for x1, Class 1: {covariance_class1_x1}")
print(f"Covariance for x1, Class 2: {covariance_class2_x1}")
print(f"Covariance for x1, Class 3: {covariance_class3_x1}")
```

```
Mean for x1, Class 1: 0.9064102564102564
Mean for x1, Class 2: -1.4842499999999998
Mean for x1, Class 3: 3.6802439024390248

Covariance for x1, Class 1: 26.65228151147099
Covariance for x1, Class 2: 38.952850705128206
Covariance for x1, Class 3: 9.45306743902439
```

```
mean_class1_x1_x2 = np.mean(class1[['x1', 'x2']], axis=0)
covariance_class1_x1_x2 = np.cov(class1[['x1', 'x2']], rowvar=False)

mean_class2_x1_x2 = np.mean(class2[['x1', 'x2']], axis=0)
covariance_class2_x1_x2 = np.cov(class2[['x1', 'x2']], rowvar=False)

mean_class3_x1_x2 = np.mean(class3[['x1', 'x2']], axis=0)
covariance_class3_x1_x2 = np.cov(class3[['x1', 'x2']], rowvar=False)

print(f"Mean for x1_x2, Class 1: {list(mean_class1_x1_x2)}")
print(f"Mean for x1_x2, Class 2: {list(mean_class2_x1_x2)}")
print(f"Mean for x1_x2, Class 3: {list(mean_class3_x1_x2)}\n")

print(f"Covariance for x1_x2, Class 1:\n {covariance_class1_x1_x2}")
print(f"Covariance for x1_x2, Class 2:\n {covariance_class2_x1_x2}")
print(f"Covariance for x1_x2, Class 3:\n {covariance_class3_x1_x2}")
```

```
Mean for x1_x2, Class 1: [0.9064102564102564, -1.0448717948717952]
Mean for x1_x2, Class 2: [-1.4842499999999998, -1.1482499999999998]
Mean for x1_x2, Class 3: [3.6802439024390248, 1.244390243902439]

Covariance for x1_x2, Class 1:
 [[26.65228151 15.823661  ]
 [15.823661   18.38869933]]
Covariance for x1_x2, Class 2:
 [[38.95285071  7.96999481]
 [ 7.96999481 11.40803532]]
Covariance for x1_x2, Class 3:
 [[9.45306744 6.3758239 ]
 [6.3758239  6.85514024]]
```

```
mean_class1_x1_x2_x3 = np.mean(class1[['x1', 'x2', 'x3']], axis=0)
covariance_class1_x1_x2_x3 = np.cov(class1[['x1', 'x2', 'x3']], rowvar=False)

mean_class2_x1_x2_x3 = np.mean(class2[['x1', 'x2', 'x3']], axis=0)
covariance_class2_x1_x2_x3 = np.cov(class2[['x1', 'x2', 'x3']], rowvar=False)

mean_class3_x1_x2_x3 = np.mean(class3[['x1', 'x2', 'x3']], axis=0)
covariance_class3_x1_x2_x3 = np.cov(class3[['x1', 'x2', 'x3']], rowvar=False)

print(f"Mean for x1_x2_x3, Class 1: {list(mean_class1_x1_x2_x3)}")
print(f"Mean for x1_x2_x3, Class 2: {list(mean_class2_x1_x2_x3)}")
print(f"Mean for x1_x2_x3, Class 3: {list(mean_class3_x1_x2_x3)}\n")

print(f"Covariance for x1_x2_x3, Class 1:\n {covariance_class1_x1_x2_x3}")
print(f"Covariance for x1_x2_x3, Class 2:\n {covariance_class2_x1_x2_x3}")
print(f"Covariance for x1_x2_x3, Class 3:\n {covariance_class3_x1_x2_x3}")
```

```
Mean for x1_x2_x3, Class 1: [0.9064102564102564, -1.0448717948717952, -0.36179487179487185]
Mean for x1_x2_x3, Class 2: [-1.4842499999999998, -1.1482499999999998, -1.0092500000000002]
Mean for x1_x2_x3, Class 3: [3.6802439024390248, 1.244390243902439, 1.0560975609756098]

Covariance for x1_x2_x3, Class 1:
 [[26.65228151 15.823661    4.65561444]
 [15.823661   18.38869933 -0.83357476]
 [ 4.65561444 -0.83357476 20.78017827]]
Covariance for x1_x2_x3, Class 2:
 [[ 38.95285071   7.96999481 -18.84354801]
 [  7.96999481  11.40803532   0.42592429]
 [-18.84354801   0.42592429  19.70946865]]
Covariance for x1_x2_x3, Class 3:
 [[ 9.45306744  6.3758239   9.46632848]
 [ 6.3758239   6.85514024  5.76523256]
 [ 9.46632848  5.76523256 40.60765439]]
```

γ) Αφού επιλέξτε τον κατάλληλο ταξινομητή να προσδιορίσετε υπολογιστικά το εμπειρικό σφάλμα ταξινόμησης, δηλαδή το ποσοστό των σημείων που ταξινομείται εσφαλμένα, υποθέτοντας ότι οι a priori πιθανότητες είναι P(ω1)= P(ω2) = ½ , κάνοντας χρήση μόνο του πρώτου χαρακτηριστικού x1

Για να ταξινομήσουμε τα σημεία μπορούμε να χρησιμοποιήσουμε τη συνάρτηση διάκρισης που ορίσαμε στο ερώτημα α.

Όσον αφορά τους πίνακες συνδιακύμανσης, είναι αυθαίρετοι και διαφορετικοί και για τις τρείς κλάσεις. Οπότε, σύμφωνα με τις τρείς κατηγορίες που έχουμε αναφέρει:

1. $\Sigma_i = \sigma^2 \mathrm{I}$, ο πίνακας συνδιακύμανσης είναι διαγώνιος, έχοντας στην κύρια διαγώνιο την τιμή σ^2 και σε όλες τις άλλες θέσεις μηδενικές τιμές. Επίσης, οι πίνακες συνδιακύμανσης για όλες τις κατηγορίες είναι ίδιοι.

2. $\Sigma_i = \Sigma$, οι πίνακες συνδιακύμανσης για όλες τις κατηγορίες είναι ίδιοι, αλλά αυθαίρετοι.

3. $\Sigma_i = \mathrm{Α}\upsilon\theta\alpha\acute{\iota}\,\rho\varepsilon\tau\sigma$, οι πίνακες συνδιακύμανσης είναι διαφορετικοί και αυθαίρετοι για κάθε κατηγορία.

Το πρόβλημα ταξινόμησης που έχουμε εμπίπτει στην 3η κατηγορία. Οπότε, για την επίλυσή του θα χρησιμοποιήσουμε τον ταξινομητή Bayes (συνάρτηση διάκρισης). Το ίδιο ισχύει και για τα υπόλοιπα δύο ερωτήματα της άσκησης, στα οποία επίσης θα χρησιμοποιήσουμε τη συνάρτηση διάκρισης.

Ο ταξινομητής Bayes λαμβάνει υπόψιν του και τους πίνακες συνδιακύμανσης και τις εκ των προτέρων πιθανότητες. Ο ταξινομητής που χρησιμοποιεί την απόσταση Mahalanobis λαμβάνει υπόψιν μόνο τους πίνακες συνδιακύμσνσης, ενώ ο ταξινομητής που χρησιμοποιεί την Ευκλείδεια απόσταση δε χρησιμοποιεί τίποτα από τα δύο. Συνεπώς, η καλύτερη επιλογή που μας βοηθάει να χρησιμοποιήσουμε όλα τα δεδομένα που έχουμε είναι ο ταξινομητής Bayes (συνάρτηση διάκρισης). Εδώ γνωρίζουμε από τις εκ των προτέρων πιθανότητες, ότι P(ω3)=0. Αν χρησιμοποιούσαμε κάποιον άλλο ταξινομητή εκτός από τη συνάρτηση διάκρισης, τότε για τις εκ των προτέρων πιθανότητες θα υποθέταμε P(ω1)= P(ω1)= P(ω3), δηλαδή ισοπίθανες. Οπότε, θα καταλήγαμε σε λιγότερο ακριβές συμπέρασμα (μεγαλύτερο σφάλμα). Αυτός είναι ένας ακόμη λόγος που προτιμήσαμε να χρησιμοποιήσουμε τη συνάρτηση διάκρισης.

Ένα σημείο θα ταξινομηθεί στην κλάση, για την οποία η συνάρτηση διάκρισης έχει τη μεγαλύτερη τιμή.

Ισχύει ότι: $P(\omega 1) + P(\omega 2) + P(\omega 3) = 1 \Rightarrow P(\omega 3) = 0$

Οπότε το πρόβλημά μας μετατρέπεται σε πρόβλημα ταξινόμησης δύο κλάσεων.

Αν $g_1(\mathrm{x}) > g_2(\mathrm{x})$, τότε το σημείο ταξινομείται στην κλάση 1, διαφορετικά ταξινομείται στην κλάση 2.

Για να βρω το εμπειρικό σφάλμα ταξινόμησης, δηλαδή το ποσοστό των σημείων που ταξινομείται εσφαλμένα πρέπει να υπολογίσω για πόσα σημεία της κλάσης 1 ισχύει: $g_1(\mathrm{x}) < g_2(\mathrm{x})$ (λανθασμένη ταξινόμηση σε κλάση 2) και για πόσα σημεία της κλάσης 2 ισχύει: $g_1(\mathrm{x}) > g_2(\mathrm{x})$ (λανθασμένη ταξινόμηση σε κλάση 1). Τέλος διαιρώ με τον συνολικό

αριθμό των σημείων για να βρω το ποσοστό. Το εμπειρικό σφάλμα φαίνεται στην παρακάτω εικόνα.

```
The points misclassified using feature x1 are 32 out of 79, error is: 40.5063%
```

δ) Επαναλάβετε το προηγούμενο βήμα χρησιμοποιώντας πρώτα δύο χαρακτηριστικά (x1 και x2) και τέλος όλα τα χαρακτηριστικά. Σχολιάστε τα αποτελέσματά σας και ιδιαίτερα τη σχέση του εμπειρικού σφάλματος με τις διαστάσεις του προβλήματος

Για τα χαρακτηριστικά x1 και x2 έχουμε:

```
The points misclassified using features x1, x2 are 30 out of 79, error is: 37.9747%
```

Για τα χαρακτηριστικά x1, x2 και x3 έχουμε:

```
The points misclassified using features x1, x2, x3 are 17 out of 79, error is: 21.519%
```

Παρατηρούμε ότι όταν αυξάνουμε τα χαρακτηριστικά από ένα σε δύο το σφάλμα μειώνεται πολύ λίγο. Αυτό πιθανώς συμβαίνει επειδή αν και δίνεται περισσότερη πληροφορία που μπορεί να αξιοποιηθεί στην ταξινόμηση αυτή η πληροφορία μπορεί να εισάγει κάποιο θόρυβο. Για αυτό το λόγο δε βλέπουμε σημαντική βελτίωση.

Όταν αυξάνουμε τα χαρακτηριστικά σε τρία το σφάλμα μειώνεται ραγδαία (έχουμε σχεδόν το μισό σφάλμα σε σχέση με το αρχικό). Αυτό σημαίνει ότι το νέο χαρακτηριστικό περιλαμβάνει πολύτιμη πληροφορία που βοηθάει καθοριστικά στην ταξινόμηση. Γενικά, τα νέα χαρακτηριστικά αν είναι μη γραμμικώς εξαρτώμενα από τα προηγούμενα χαρακτηριστικά, μπορούν να συνεισφέρουν στην μείωση του σφάλματος (καλύτερη ταξινόμηση).

Βέβαια, η προσθήκη νέων χαρακτηριστικών δε βοηθάει πάντα στην εκμηδένιση των σφαλμάτων. Η χρήση περισσότερων παραμέτρων μπορεί να έχει αρνητικό αποτέλεσμα αν δεν γίνει αύξηση των αντίστοιχων δεδομένων εκπαίδευσης (Curse of dimensionality). Όσο τα δεδομένα μας είναι λίγα, δεν είναι εύκολο να δούμε σημαντική βελτίωση (σχεδόν εξάλειψη των σφαλμάτων).

ε) Επαναλάβετε την εκτίμηση του σφάλματος χρησιμοποιώντας όλα τα χαρακτηριστικά για P(ω1) = 0.8 και P(ω2) = P(ω3) = 0.1. Σε ποια περίπτωση μπορούμε να χρησιμοποιήσουμε καθέναν από τους ταξινομητές που υλοποιήσατε στο βήμα (α)

Όπως είπαμε και προηγουμένως, ένα σημείο θα ταξινομηθεί στην κλάση, για την οποία η συνάρτηση διάκρισης έχει τη μεγαλύτερη τιμή.

Αν $\max\left(g_1(\mathrm{x}), g_2(\mathrm{x}), g_3(\mathrm{x})\right) = g_i(\mathrm{x})$, τότε το σημείο ταξινομείται στην κλάση i.

Για να βρω το εμπειρικό σφάλμα ταξινόμησης πρέπει να υπολογίσω για πόσα σημεία της κλάσης 1 ισχύει: $g_1(\mathrm{x}) < g_2(\mathrm{x})$ ή $g_1(\mathrm{x}) < g_3(\mathrm{x})$ (λανθασμένη ταξινόμηση σε κλάση 2 ή 3) και να κάνω το ίδιο και για τις άλλες δύο κλάσεις αντίστοιχα.

Το εμπειρικό σφάλμα φαίνεται στην παρακάτω εικόνα.

```
The points misclassified using features x1, x2, x3 are 66 out of 120, error is: 55.0%
```

Το εμπειρικό σφάλμα αυξήθηκε, καθώς τώρα έχουμε μία νέα κλάση, άρα το πρόβλημα έγινε πιο δύσκολο σε σχέση με πριν.

Τους τρείς ταξινομητές που έχουν αναφερθεί θα τους χρησιμοποιούμε στις ακόλουθες περιπτώσεις:

1. Αν $\Sigma_i = \sigma^2 \mathrm{I}$, τότε η συνάρτηση διάκρισης ανάγεται στον ταξινομητή που χρησιμοποιεί την Ευκλείδεια απόσταση, άρα σε αυτήν την περίπτωση θα κάνουμε ταξινόμηση με βάση την Ευκλείδεια απόσταση. Τα χαρακτηριστικά είναι στατιστικώς ανεξάρτητα και έχουν την ίδια διασπορά.

$$\textit{Euclidean distance:} \quad d_\epsilon = \|\boldsymbol{x} - \boldsymbol{\mu_i}\|$$

Τα δείγματα βρίσκονται σε υπερ-σφαίρες ίσου μεγέθους. Οι επιφάνειες απόφασης είναι υπερεπίπεδα διάστασης d-1 (γραμμικές συναρτήσεις διάκρισης).

2. Αν $\Sigma_i = \Sigma$, τότε η συνάρτηση διάκρισης ανάγεται στον ταξινομητή που χρησιμοποιεί την απόσταση Mahalanobis, άρα σε αυτήν την περίπτωση θα κάνουμε ταξινόμηση με βάση την απόσταση Mahalanobis.

$$\textit{Mahalanobis distance:} \quad d_m = \left((\boldsymbol{x} - \boldsymbol{\mu}_i)^T \Sigma^{-1} (\boldsymbol{x} - \boldsymbol{\mu}_i)\right)^{1/2}$$

Τα δείγματα δημιουργούν υπερ-ελλιψοειδείς ομάδες ίδιου μεγέθους και σχήματος με κέντρα τα μ_i. Οι επιφάνειες απόφασης είναι υπερεπίπεδα (γραμμικές συναρτήσεις διάκρισης).

3. Αν $\Sigma_i = \mathrm{Α}\upsilon\theta\alpha\acute{\iota}\rho\varepsilon\tau\sigma$, τότε η συνάρτηση διάκρισης δεν μπορεί να απλοποιηθεί περεταίρω.

$$g_i(x) = -\frac{1}{2}(x-\mu_i)^t \sum_i^{-1}(x-\mu_i) - \frac{d}{2}\ln 2\pi - \frac{1}{2}\ln|\Sigma_i| + \ln P(\omega_i)$$

Οι επιφάνειες απόφασης είναι hyperquadratics (μη γραμμικές συναρτήσεις διάκρισης, αλλά τετραγωνικές).

Οι περιπτώσεις 1 και 2 αναφέρονται σε ισοπίθανες κλάσεις με ίδιους πίνακες συνδιασποράς. Οι ταξινομητές που χρησιμοποιούνται σε τέτοια προβλήματα, είναι

ταξινομητές ελάχιστης απόστασης (minimum distance classifiers). Από αυτές τις δύο παραδοχές η συνάρτηση διάκρισης γίνεται:

$$g_i(\boldsymbol{x}) = -\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu}_i)^T \Sigma^{-1}(\boldsymbol{x} - \boldsymbol{\mu}_i)$$

Οπότε, η συνάρτηση διάκρισης μπορεί να χρησιμοποιηθεί σε οποιαδήποτε περίπτωση. Ο ταξινομητής με την απόσταση Mahalanobis μπορεί να χρησιμοποιηθεί σε μια ομάδα περιπτώσεων (ίδιοι πίνακες συνδιαποράς) και ο ταξινομητής με την Ευκλείδεια απόσταση μπορεί να χρησιμοποιηθεί σε μια ακόμη μικρότερη ομάδα περιπτώσεων (ίδιοι πίνακες συνδιαποράς και διαγώνιοι, τα στοιχεία της διαγωνίου ίσα με σ^2).

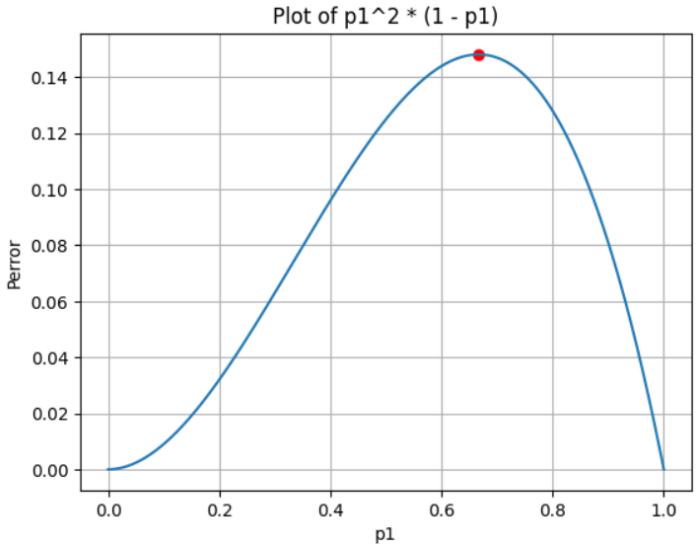
Άσκηση 4

Καταλήγετε στο συμπέρασμα πως η πιθανότητα σφάλματος ενός Bayesian συστήματος ταξινόμησης δύο κλάσεων εξαρτάται από την a priori πιθανότητα P_1 της κλάσης ω_1 σύμφωνα με τη σχέση:

$$P_{error}(p_1) = p_1^2(1-p_1),\ 0 \le p_1 \le 1$$

α) Εάν δεν γνωρίζετε το P1, για ποια τιμή του θα πρέπει να σχεδιάσετε τα όρια απόφασης του συστήματός σας ώστε να ελαχιστοποιήσετε το μέγιστο πιθανό σφάλμα και ποια θα είναι η πιθανότητα σφάλματος στην περίπτωση αυτή

Για την ελαχιστοποίηση του μέγιστου πιθανού σφάλματος, πρέπει να βρούμε από το διάγραμμα της συνάρτησης σφάλματος, πού παίρνει τη μέγιστη τιμή, δηλαδή σε ποιο σημείο το σφάλμα γίνεται μέγιστο (χειρότερη περίπτωση). Άρα, αν δε γνωρίζαμε το p1, θα πρέπει να σχεδιάζουμε το σύστημα, ώστε να ελαχιστοποιεί την χειρότερη περίπτωση κόστους. Οπότε, θα επιλέγαμε p1 = 0.666.



$$P_{error}(0.666) = 0.666^2(1-0.666) = 0.148$$

Ο υπολογισμός από το διάγραμμα:

```
max_value = max(y)
max_index = np.argmax(y) / points # 0-1 there are 1000 points

print(f"Max error is {round(max_value, 3)} for p1 {round(max_index, 3)}")
```

```
Max error is 0.148 for p1 0.666
```

Για να βρω το μέγιστο με υπολογισμούς έχω:

$$\frac{dP_{error}}{dp_1} = 2p_1(1-p_1) - p_1^{\,2}$$

$$\Theta\acute{\varepsilon}\tau\omega \ \frac{dP_{error}}{dp_1} = 0 => 2p_1(1-p_1) - p_1^{\,2} = 0 =>$$

$$-3p_1^{\,2} + 2p_1 = 0 => p_1(-3p_1 + 2) = 0 =>$$

$$p_1 = 0 \ \text{ή} \ p_1 = 2/3 = 0.666666....67 \sim 0.667$$

$$\text{Ο}\pi\acute{o}\tau\varepsilon :$$

$$P_{error}(0.667) = 0.667^2(1-0.667) = 0.148$$

β) Εάν ρυθμίσατε το σύστημά σας υποθέτοντας πιθανότητα P1=0.3, ποια θα είναι η πιθανότητα σφάλματος εάν η πραγματική a priori πιθανότητα της κλάσης ω1 είναι P1=0.7

Έχουμε ότι:

$$P_{error} = p_1^{\,2} * (1-p_1)$$

$$\frac{dP_{error}}{dp_1} = 2p_1(1-p_1) - p_1^{\,2}$$

Για να χαράξουμε την ευθεία που θα είναι κάθετη στην καμπύλη του σφάλματος, στο σημείο p1=0.3, χρειαζόμαστε ένα σημείο και την κλίση της καμπύλης σε αυτό το σημείο. Στην ουσία η ευθεία που θέλουμε να σχεδιάσουμε είναι η κλίση της καμπύλης του σφάλματος. Έχουμε:

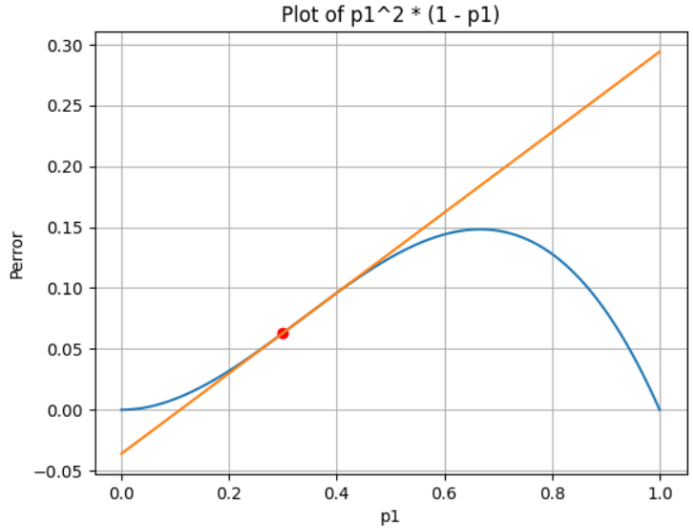
$$p_1 = 0.3 => P_{error}(0.3) = 0.3^2(1-0.3) = 0.063$$

$$\frac{dP_{error}(0.3)}{dp_1} = 2*0.3*(1-0.3) - 0.3^2 = 0.33, \ \kappa\lambda\acute{\iota}\sigma\eta$$

$$y - y_0 = a(\mathrm{x} - \mathrm{x}_0) => y = 0.33(\mathrm{x} - 0.3) + 0.063 =>$$

$$\mathrm{y} = 0.33\,\mathrm{x} - 0.099 + 0.063 => y = 0.33x - 0.036$$

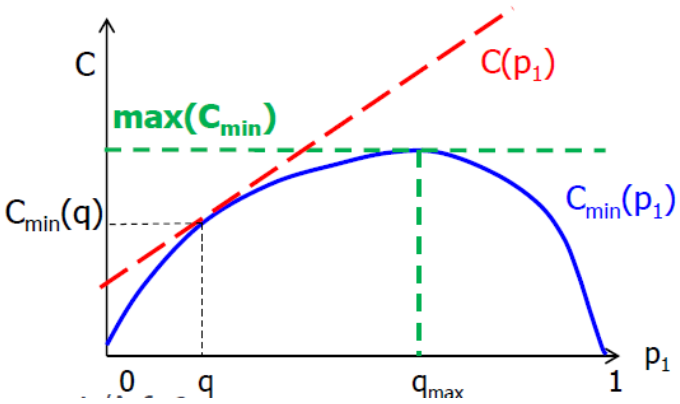
Η κλίση της καμπύλης στο σημείο της όπου ισχύει p1=0.3 (η ευθεία που χαράξαμε), παριστάνει την πιθανότητα σφάλματος που θα έχουμε αν ρυθμίσουμε το σύστημά μας, υποθέτοντας p1=0.3.



Αν το πραγματικό p1=0.7 η πιθανότητα σφάλματος είναι:

$$y = 0.33x - 0.036 \Rightarrow$$
$$y = 0.33 * 0.7 - 0.036 = 0.195$$

Το διπλανό σχήμα εξηγεί ότι, για p1=qmax για το οποίο η Cmin γίνεται μέγιστη είμαστε βέβαιοι ότι η πιθανότητα σφάλματος θα είναι πάντοτε ίση με αυτή την τιμή για κάθε p1.



Για κάθε άλλη επιλογή του p1, η πιθανότητα σφάλματος C(p1) θα λαμβάνει τιμές μεγαλύτερες του max(Cmin) για κάποιες τιμές του p1.

Αυτό επαληθεύεται και με τα παραπάνω αποτελέσματα:

max(Cmin)=0.148, p1=0.666

C=0.195, p1=0.3 (επιλέξαμε 0.3, ενώ το πραγματικό είναι 0.7)

C=0.195 > 0.148

Τα αποτελέσματά μας επαληθεύτηκαν και με τη βοήθεια κώδικα.

```
# find error when real p1 is 0.7
x_real = 0.7
cost = 0.33 * x_real - 0.036
print(f"Cost for p1 = {x_real} is {round(cost, 3)} when we have regulate the system considered p1=0.3")
```

```
Cost for p1 = 0.7 is 0.195 when we have regulate the system considered p1=0.3
```

Έχουμε: $P(\omega_1 \mid \mathrm{x}) = \dfrac{P(\mathrm{x} \mid \omega_1)P(\omega_1)}{P(x \mid \omega_1)\mathrm{P}(\omega_1) + P(x \mid \omega_2)\mathrm{P}(\omega_2)}$

Η πιθανότητα σφάλματος (0.195) είναι μεγαλύτερη από την τιμή που υποθέσαμε (0.063). Το σύστημα θα ταξινομήσει λάθος τα δείγματα της κλάσης ω1 ως δείγματα της κλάσης ω2 με μεγαλύτερη συχνότητα, γιατί η εκ των προτέρων πιθανότητα για το p1 είναι στην πραγματικότητα μεγαλύτερη από αυτή που υποθέσαμε.

Άσκηση 5

Αναδρομική εκτίμηση Bayes, N=10, {κ,γ,κ,κ,κ,γ,κ,κ,γ,κ}.

Ποια είναι η πιθανότητα θ να φέρουμε κεφάλι;

$$p(\theta \mid D^0) = A \cdot \theta(1-\theta)^4, \text{για } 0 \le \theta \le 1 \quad (\text{beta distribution}).$$

Η πιθανότητα θ να φέρουμε κεφάλι, αν έχουμε ρίξει N=10 φορές το νόμισμα και έχουμε φέρει κ=7 φορές κεφάλι είναι:

$$\hat{\theta} = \frac{\kappa}{\mathrm{N}} = \frac{7}{10} = 0.7$$

α) Να υπολογισθεί το A

Γνωρίζουμε ότι για κάθε συνάρτηση πυκνότητας πιθανότητας (pdf) ισχύει:

$$\int_{-\infty}^{\infty} P(\theta \mid \mathrm{D}^0)d\theta = 1 \Rightarrow \int_{0}^{1} P(\theta \mid \mathrm{D}^0)d\theta = 1 \Rightarrow \int_{0}^{1} A*\theta*(1-\theta)^4 d\theta = 1$$

Επίσης για τη beta function γνωρίζουμε ότι:

$$\mathrm{B}(z_1, z_2) = \int_0^1 t^{z_1-1}(1-t)^{z_2-1}\,dt$$

$$\mathrm{B}(z_1, z_2) = \frac{\Gamma(z_1)\,\Gamma(z_2)}{\Gamma(z_1+z_2)}.$$

Για τη gamma function γνωρίζουμε ότι:

$$\Gamma(n) = (n-1)!\,.$$

Οπότε έχουμε:

$$\int_0^1 A*\theta*(1-\theta)^4 d\theta = 1 => A*\int_0^1 \theta^{2-1}*(1-\theta)^{5-1} d\theta = 1,\ \ z1=2 \text{ και } z2=5$$

$$=> A*\frac{\Gamma(z1)\Gamma(z2)}{\Gamma(z1+z2)} = 1 => A*\frac{\Gamma(2)\Gamma(5)}{\Gamma(7)} = 1 => A*\frac{(2-1)!(5-1)!}{(7-1)!} = 1 =>$$

$$A*\frac{(1)!(4)!}{(6)!} = 1 => A*\frac{1*24}{720} = 1 => A = 30$$

β) Να σχεδιασθούν σε κοινό διάγραμμα τα $P(\theta|D^1), P(\theta|D^5), P(\theta|D^{10})$

Γνωρίζουμε την Bayesian αναδρομική μεθοδολογία εκμάθησης:

$$p(\boldsymbol{\theta}/D^n) = \frac{p(\mathbf{x}_n/\boldsymbol{\theta})\, p(\boldsymbol{\theta}/D^{n-1})}{\int_{\boldsymbol{\theta}} p(\mathbf{x}_n/\boldsymbol{\theta})\, p(\boldsymbol{\theta}/D^{n-1})\, d\boldsymbol{\theta}}$$

$$p(\boldsymbol{\theta}/D^0) = p(\boldsymbol{\theta})$$

Για μία ρίψη ισχύει:

$$p(x_n|\boldsymbol{\theta}) = \begin{cases} \boldsymbol{\theta} & \text{εάν κεφάλι} \\ (1-\boldsymbol{\theta}) & \text{εάν γράμματα} \end{cases}$$

Για N ρίψεις:

$$p(\boldsymbol{\theta}|D^N) = \frac{\boldsymbol{\theta}^k(1-\boldsymbol{\theta})^{N-k}\, p(\boldsymbol{\theta}|D^0)}{\int \boldsymbol{\theta}^k(1-\boldsymbol{\theta})^{N-k}\, p(\boldsymbol{\theta}|D^0) d\boldsymbol{\theta}}$$

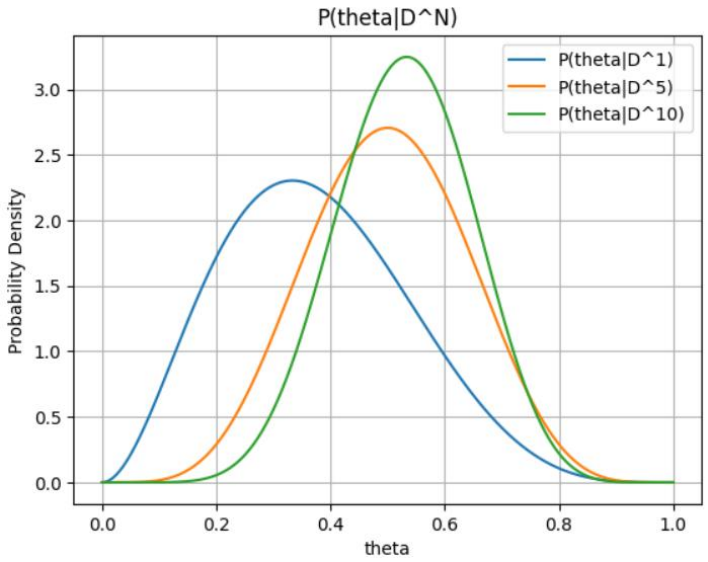
Για τις συναρτήσεις στα διαγράμματα δημιουργήσαμε μια λίστα y, η οποία θα περιέχει τις $P(\theta|D^1), P(\theta|D^5), P(\theta|D^{10})$. Κάθε ένα από τα τρία στοιχεία της λίστας περιέχει 1000 σημεία (επιλέξαμε το θ να έχει 1000 σημεία και να εκτείνεται από το 0 έως το 1).

Ο παρονομαστής που περιέχει το ολοκλήρωμα δημιουργείται με τη βοήθεια της συνάρτησης **quad**. Προσοχή πρέπει να δοθεί στο γεγονός ότι αυτή η συνάρτηση καλεί τη συνάρτηση που ορίσαμε (συνάρτηση num), χωρίς το όρισμα theta και τα υπόλοιπα ορίσματα δεν τα βάζουμε απλά δίπλα με κόμμα, αλλά τα βάζουμε στην παράμετρο args σαν tuple. Το N παίρνει διαδοχικά τις τιμές 1, 5, 10 για να υπολογιστουν οι ζητούμενες τιμές. Το κ μετράει στην ουσία τις φορές που έχουμε φέρει κεφάλι (από την αρχή μέχρι το N).

Ο κώδικας για τους υπολογισμούς φαίνεται παρακάτω.

```
coin = np.array([1, 0, 1, 1, 1, 0, 1, 1, 0, 1]) # heads=1, tails=0
points = 1000
theta = np.linspace(0, 1, points)
y = []

def num(theta, k, N):
  return (theta**k)*(1-theta)**(N-k)*A*beta(theta)

for N in [1, 5, 10]:
  k = (coin[0:N] == 1).sum() # count heads
  den = quad(num, 0, 1, args=(k, N))[0]
  y.append(num(theta, k, N)/den)
```

Το διάγραμμα με τις 3 συναρτήσεις φαίνεται στην παρακάτω εικόνα:



Με τη χρήση της συνάρτησης **trapz** μπορούμε να ελέγξουμε ότι το ολοκλήρωμα κάθε μιας από τις τρείς συναρτήσεις είναι 1: $\int_{-\infty}^{\infty} P(\theta \mid \mathrm{D}^{\mathrm{N}}) d\theta = 1,\ \text{για κάθε N}$

γ) Να βρεθεί (αριθμητικά) το $\mathrm{p}(\mathrm{x} = \gamma \mid \mathrm{D}^{10})$ μετά τη 10η ρίψη

Για να βρούμε την πιθανότητα να φέρουμε κορώνα μετά τη 10η ρίψη ($\mathrm{p}(\mathrm{x} = \kappa \mid \mathrm{D}^{10})$), αρκεί να εντοπίσουμε το σημείο όπου η κατανομή $P(\theta \mid D^{10})$ έχει την μεγαλύτερη τιμή.

Η πιθανότητα να φέρουμε γράμματα ($\mathrm{p}(\mathrm{x} = \gamma \mid \mathrm{D}^{10})$) είναι η συμπληρωματική πιθανότητα:

$$\mathrm{p}(\mathrm{x} = \gamma \mid \mathrm{D}^{10}) = 1 - \mathrm{p}(\mathrm{x} = \kappa \mid \mathrm{D}^{10}) = 1 - 0.533 = 0.467$$

Έχουμε:

$$p(x_n \mid \boldsymbol{\theta}) = \begin{cases} \boldsymbol{\theta} & \text{εάν κεφάλι} \\ (1-\boldsymbol{\theta}) & \text{εάν γράμματα} \end{cases}$$

Τα αποτελέσματα του κώδικα φαίνονται παρακάτω.

```
max_value = max(y[2])
max_index = np.argmax(y[2]) / points # 0-1 there are 1000 points

print(f"Probability Density is {round(max_value, 3)} for theta (P(x=k|D^10)) = {round(max_index, 3)}")
print(f"Theta (P(x=γ|D^10)) = {round(1-max_index, 3)}") # p(γ)=1-p(κ)
```

```
Probability Density is 3.249 for theta (P(x=k|D^10)) = 0.533
Theta (P(x=γ|D^10)) = 0.467
```