Tutorial 4: Classification with Gaussians

1. Consider a pattern recognition problem with two classes A and B. Each class is modelled by a class-conditional Gaussian density. Class A is parameterised by mean $\mu_A = 4$ and variance $\sigma_A^2 = 1$; class B is parameterised by mean $\mu_B = 2$ and variance $\sigma_B^2 = 4$.

   (a) On the same graph sketch the probability density function for each class.

   SOLUTION:

   (sketch)

   (b) The following three test items are observed:

   $$x_1 = 3$$
   $$x_2 = 4$$
   $$x_3 = 8$$

   Assume that the classes have equal prior probabilities. To which classes should these points be assigned?

   SOLUTION:

   This is a two-class problem, define a discriminant function as the log ratio of posterior probabilities

   $$y(x) = \ln \frac{P(A|x)}{P(B|x)} = \ln \frac{p(x|A)\,P(A)}{p(x|B)\,P(B)}$$

   And since in this case the priors are equal

   $$y(x) = \ln \frac{p(x|A)}{p(x|B)}$$

   The decision boundary between the classes is defined by $y(x) = 0$. If $y(x) > 0$ assign x to class A, otherwise assign to B. Since we are using Gaussian pdfs:

   $$\begin{aligned} y(x) &= \ln \frac{p(x|A)}{p(x|B)} = \ln p(x|A) - \ln p(x|B) \\ &= -\ln(\sqrt{2\pi}) - \ln \sigma_A - \frac{(x-\mu_A)^2}{2\sigma_A^2} + \ln(\sqrt{2\pi}) + \ln \sigma_B + \frac{(x-\mu_B)^2}{2\sigma_B^2} \\ &= -\ln \sigma_A + \ln \sigma_B - \frac{(x-\mu_A)^2}{2\sigma_A^2} + \frac{(x-\mu_B)^2}{2\sigma_B^2} \\ &= \ln 2 - \frac{(x-4)^2}{2} + \frac{(x-2)^2}{8} \end{aligned}$$

   For data points $x_1 = 3$, $x_2 = 4$,$x_3 = 8$:

   $$y(3) = \ln 2 - \frac{1}{2} + \frac{1}{8} > 0$$
   $$y(4) = \ln 2 + \frac{1}{2} > 0$$
   $$y(8) = \ln 2 - \frac{16}{2} + \frac{36}{8} < 0$$

   Assign x_1 and x_2 to class A, assign x_3 to B

(c) You are told that the prior probability of class B is twice that of class A. To which classes would you now assign points x_1, x_2, x_3?

SOLUTION:

In this case the priors are not equal

$$\begin{aligned} y(x) &= \ln \frac{P(A|x)}{P(B|x)} = \ln \frac{p(x|A)\,P(A)}{p(x|B)\,P(B)} \\ &= \ln \frac{p(x|A)\,P(A)}{p(x|B)(2P(A))} = \ln \frac{p(x|A)}{2p(x|B)} \\ &= \ln 2 - \frac{(x-4)^2}{2} + \frac{(x-2)^2}{8} - \ln 2 \\ &= \frac{(x-4)^2}{2} + \frac{(x-2)^2}{8} \end{aligned}$$

The change in the priors shifts the discriminant function by $-\ln 2$.

In this case for data points $x_1 = 3$, $x_2 = 4$,$x_3 = 8$:

$$\begin{aligned} y(3) &= -\frac{1}{2} + \frac{1}{8} < 0 \\ y(4) &= \frac{1}{2} > 0 \\ y(8) &= -\frac{16}{2} + \frac{36}{8} < 0 \end{aligned}$$

Assign x_2 to class A, assign x_1 and x_3 to B

(d) What are the benefits and drawbacks of using Gaussian probability density functions as a generative model for real world pattern recognition problems?

SOLUTION:

Main benefits of using Gaussians

- Straightforward to estimate parameters (mean and (co)variance) from data
- Computationally efficient
- Straightforward interpretation of parameters (mean:location; variance:dispersion)
- Many real-world classes are reasonably well modelled by Gaussians

Drawbacks:

- Many classes not well-modelled by Gaussians:
  - Cannot model distributions with multiple modes
  - Not a good fit to distributions with long tails
  - Not a good fit to skew, non-symmetric distributions
- Not well-suited to discrete or categorical data
- Not well-suited to sequential data
- Not always straightforward to estimate full covariance matrix

2. In a two-class pattern classification problem, with classes A and B, each class is modelled using a one-dimensional Gaussian probability density function:

$$\begin{aligned} p(x|A) &= \mathcal{N}(x; \mu_A, \sigma_A^2) \\ p(x|B) &= \mathcal{N}(x; \mu_B, \sigma_B^2). \end{aligned}$$

Assume the classes have equal prior probabilities, and that $\mu_A \neq \mu_B$ and $\sigma_A^2 \neq \sigma_B^2$.

(a) Write down a suitable discriminant function for this problem.

SOLUTION:

For this two class problem (recalling $P(A) = P(B)$) a suitable discriminant function is given by:

$$\begin{aligned} y(x) &= \ln \frac{p(x|A)\,P(A)}{p(x|B)\,P(B)} = \ln \frac{p(x|A)}{p(x|B)} = \ln \frac{N(x;\mu_A,\sigma_A^2)}{N(x;\mu_B,\sigma_B^2)} \\ &= -\frac{1}{2}\ln\sigma_A^2 - \frac{(x-\mu_A)^2}{2\sigma_A^2} + \frac{1}{2}\ln\sigma_B^2 + \frac{(x-\mu_B)^2}{2\sigma_B^2} \\ &= -\frac{1}{2}\ln\frac{\sigma_A^2}{\sigma_B^2} - \frac{1}{2}\left(\frac{(x-\mu_A)^2}{\sigma_A^2} - \frac{(x-\mu_B)^2}{\sigma_B^2}\right) \end{aligned}$$

(b) Derive the quadratic equation in x that defines the decision boundary between the classes.

SOLUTION:

The decision boundary is defined by $y(x) = 0$. Expanding and collecting terms leads to the following quadratic equation:

$$-\frac{1}{2}\left(\frac{1}{\sigma_A^2} - \frac{1}{\sigma_B^2}\right)x^2 + \left(\frac{\mu_A}{\sigma_A^2} - \frac{\mu_B}{\sigma_B^2}\right)x + \left(-\frac{1}{2}\ln\frac{\sigma_A^2}{\sigma_B^2} - \frac{1}{2}\left[\frac{\mu_A^2}{\sigma_A^2} - \frac{\mu_B^2}{\sigma_B^2}\right]\right) = 0$$

$$\begin{aligned} w_2x^2 + w_1x + w_0 &= 0 \\ w_2 &= -\frac{1}{2}\left(\frac{1}{\sigma_A^2} - \frac{1}{\sigma_B^2}\right) \\ w_1 &= \frac{\mu_A}{\sigma_A^2} - \frac{\mu_B}{\sigma_B^2} \\ w_0 &= -\frac{1}{2}\ln\frac{\sigma_A^2}{\sigma_B^2} - \frac{1}{2}\left[\frac{\mu_A^2}{\sigma_A^2} - \frac{\mu_B^2}{\sigma_B^2}\right] \end{aligned}$$

3. The notes stated without proof that the sample mean (μ_{ML}) and sample variance (σ^2_{ML}) are the maximum likelihood solutions for the parameters of a one-dimensional Gaussian. Consider the log likelihood of a Gaussian with mean μ and variance σ^2, given a set of N data points $\{x^1, \ldots, x^N\}$:

$$\begin{aligned} L = \ln p(\{x^1,\ldots,x^n\}|\mu,\sigma^2) &= -\frac{1}{2}\sum_{n=1}^{N}\left(\frac{(x_n-\mu)^2}{\sigma^2} - \ln\sigma^2 - \ln(2\pi)\right) \\ &= -\frac{1}{2\sigma^2}\sum_{n=1}^{N}(x_n-\mu)^2 - \frac{N}{2}\ln\sigma^2 - \frac{N}{2}\ln(2\pi)\,. \end{aligned}$$

By maximising the log likelihood function with respect to μ show that we obtain the maximum likelihood estimate for the mean:

$$\mu_{ML} = \frac{1}{N}\sum_{n=1}^{N} x_n.$$

This is the sample mean, the mean of the N observed values.

SOLUTION:

Rewrite L by expanding the quadratic and noting that $\sum_{n=1}^{N} \mu^2 = N\mu^2$

$$L = -\frac{1}{2\sigma^2}\left(\sum_{n=1}^{N} x_n^2 + N\mu^2 - 2\mu\sum_{n=1}^{N} x_n\right) - \frac{N}{2}\ln\sigma^2 - \frac{N}{2}\ln(2\pi)$$

Take the partial derivative with respect to μ:

$$\frac{\partial L}{\partial \mu} = -\frac{1}{2\sigma^2}\left(2N\mu - 2\sum_{n=1}^{N} x_n\right)$$

The maximum likelihood estimate μ_{ML} is the value of μ when L is maximal, so $\partial L/\partial\mu = 0$, so:

$$-\frac{1}{2\sigma^2}\left(2N\mu_{ML} - 2\sum_{n=1}^{N} x_n\right) = 0$$

$$2N\mu_{ML} = 2\sum_{n=1}^{N} x_n$$

$$\mu_{ML} = \frac{1}{N}\sum_{n=1}^{N} x_n$$

The idea of optimising an objective function (in this case the log likelihood) is central to learning parameters from data. In the lectures we shall have started to cover gradient descent training of single layer networks. In this case the differences are finding the minimum of an error function rather than the maximum of a log likelihood function, and the fact that there is usually no closed form solution for single layer networks (in the case when there are nonlinearities, such as a sigmoid transfer function) so iterative techniques such as gradient descent must be used.

4. (a) Estimate the mean vector $\hat{\boldsymbol{\mu}}_i$ and covariance matrix $\hat{\boldsymbol{\Sigma}}_i$ for each class $i = 1, 2$ in terms of maximum likelihood. (It is advisable to do this at least by hand without using a calculator!)

SOLUTION:

The ML estimators of the mean vector and covariance matrix for a class are given by:

$$\hat{\boldsymbol{\mu}} = \frac{1}{N}\sum_{j=1}^{N} \mathbf{x}_j$$

$$\hat{\boldsymbol{\Sigma}} = \frac{1}{N}\sum_{j=1}^{N} (\mathbf{x}_j - \boldsymbol{\mu})(\mathbf{x}_j - \boldsymbol{\mu})^T$$

Using the training samples, we will have:

$$\hat{\boldsymbol{\mu}}_1 = \begin{pmatrix} 2 \\ 2 \end{pmatrix}, \qquad \hat{\boldsymbol{\Sigma}}_1 = \begin{pmatrix} \frac{1}{2} & 0 \\ 0 & 2 \end{pmatrix}$$

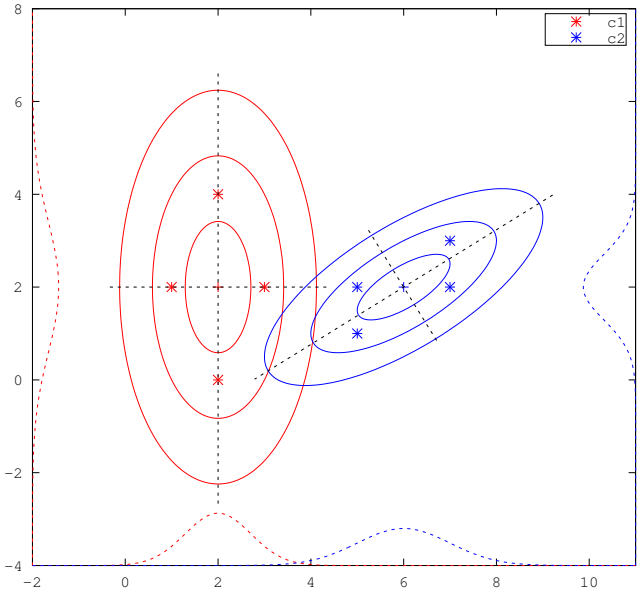
$$\hat{\boldsymbol{\mu}}_2 = \begin{pmatrix} 6 \\ 2 \end{pmatrix}, \qquad \hat{\boldsymbol{\Sigma}}_2 = \begin{pmatrix} 1 & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{pmatrix}$$

When you use Matlab's 'cov()' function to check your manually-calculated covariance matrices, you should code something like this 'cov(X,1)' rather than just 'cov(X)' to get the right covariance in terms of ML estimation.

(b) Using the parameters obtained above, sketch the contours of the normal distribution for each class.

SOLUTION:

The following figure depicts the estimated Gaussian distributions as well as the training samples, where the three contours of each distribution correspond to 1σ, 2σ and 3σ.



(c) Using Matlab/Python, find the eigen values and eigen vectors of $\hat{\mathbf{\Sigma}}_i$, and discuss how they correspond to the sketches. [non-examinable]

SOLUTION:

Eigen values and eigen vectors are as follows:

$$
\begin{array}{lll}
C_1: & \lambda_1 = 2, & \mathbf{v}_1 = (0,1)^T \\
 & \lambda_2 = 0.5, & \mathbf{v}_2 = (1,0)^T \\
C_2: & \lambda_1 = 1.31, & \mathbf{v}_1 = (-0.851,-0.526)^T \\
 & \lambda_2 = 0.191, & \mathbf{v}_2 = (0.526,-0.851)^T
\end{array}
$$

Assuming eigen values are sorted in decreasing order so that $\lambda_1 \geq \lambda_2$, the eigen vector for λ_1 corresponds to the major axis of the ellipse, whereas the eigen vector for λ_2 corresponds to the minor axis. The length of each axis is proportional to the square root of the corresponding eigen value. In another word, the variance of each axis is equal to the corresponding eigen value.