DatA414 Tutorial 2: Classification

1. K-nearest neighbour (K-NN) classification

You are given data where the target is categorical and asked to fit a K-NN classifier to this data.

a) Describe how you would choose the value of K (the number of neighbours).

b) How does the number of training data points N affect the speed (i.e. computational cost) of making a prediction on a new test point.

c) What would be the classifcation accuracy on the training data be if we set $K = 1$?

2. Gaussian Bayes classifier

A training set consists of one-dimensional examples from two classes. The training examples from class 1 are $\{0.5, 0.1, 0.2, 0.4, 0.3, 0.2, 0.2, 0.1, 0.35, 0.25\}$ and the examples from class 2 are $\{0.9, 0.8, 0.75, 1.0\}$.

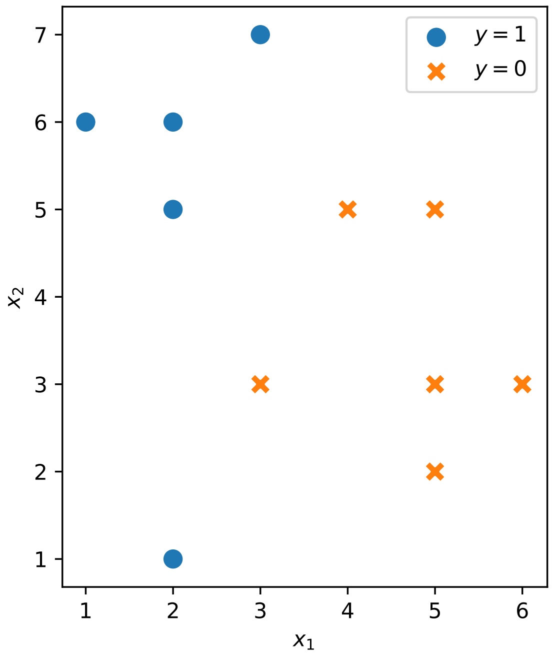
a) Fit a one-dimensional Gaussian to each class by matching the mean and variance. Also estimate the class probabilities $P(y = 1) = \pi_1$ and $P(y = 2) = \pi_2$ by matching the observed class fractions. (This procedure fits the model with maximum likelihood: it selects the parameters that give the training data the highest probability.) Sketch a plot of the scores $P(y)\, p(x \,|\, y)$ for each class y, as functions of input location x.

b) What is the probability that the test point $x = 0.6$ belongs to class 1? Mark the decision boundary/ies on your sketch, the location(s) where $P(\text{class } 1 \,|\, x) = P(\text{class } 2 \,|\, x) = 0.5$. You are not required to calculate the location(s) exactly.

c) Are the decisions that the model makes reasonable for very negative x and very positive x? Are there any changes we could consider making to the model if we wanted to change the model's asymptotic behaviour?

3. Naive Bayes classifier

Consider the following data set:



Suppose we make the naive Bayes assumption and we fit class-conditional Gaussians using maximum likelihood.

(a) For both $k = 0$ and $k = 1$, calculate rough values for $P(y = k) = \pi_k$ as well as the values of $\mu_{k,1}$, $\sigma^2_{k,1}$, $\mu_{k,2}$, $\sigma^2_{k,2}$ for the density $p(\mathbf{x}|y = k; \boldsymbol{\theta}) = \prod_{d=1}^{2} \mathcal{N}(x_d; \mu_{k,d}, \sigma^2_{k,d})$. You should therefore end up with a mean μ and variance σ^2 for each of the two dimensions for each of the two classes.

(b) Draw rough contours of the resulting class conditional densities $p(\mathbf{x}|y = 0; \boldsymbol{\theta})$ and $p(\mathbf{x}|y = 1; \boldsymbol{\theta})$ on the figure.

(c) Try to write rough Python code (on a piece of paper) to answer (a). I could ask you to do this in the exam. When you go home, try to see if this gives you the right answer.

4. Logistic regression

Suppose that, for the data in the question above, we fit a logistic regression model with a bias weight w_0, that is $P(y = 1 \,|\, \mathbf{x}; \mathbf{w}) = \sigma(w_0 + w_1 x_1 + w_2 x_2)$, by maximum likelihood, obtaining parameters $\hat{\mathbf{w}}$. Sketch a possible decision boundary corresponding to $\hat{\mathbf{w}}$. Is your answer unique? How many classification errors does your method make on the training set?

5. Maximum likelihood and logistic regression

Maximum likelihood logistic regression maximizes the log probability of the labels,

$$\sum_{n=1}^{N} \log P(y^{(n)} \,|\, \mathbf{x}^{(n)}; \mathbf{w})$$

with respect to the weights $\mathbf{w}$. As usual, $y^{(n)}$ is a binary label at input location $\mathbf{x}^{(n)}$.

The training data is said to be *linearly separable* if the two classes can be completely separated by a hyperplane. That means we can find a decision boundary

$$P(y^{(n)}\!=\!1 \,|\, \mathbf{x}^{(n)}; \mathbf{w}, w_0) = \sigma(\mathbf{w}^\top \mathbf{x}^{(n)} + w_0) = 0.5, \qquad \text{where } \sigma(a) = \frac{1}{1+e^{-a}}$$

such that all the $y\!=\!1$ labels are on one side (with probability greater than 0.5), and all of the $y\!\neq\!1$ labels are on the other side. (Note that here we have decided not to use the $x_0 = 1$ trick, i.e. the bias w_0 is treated separately.)

a) Show that if the training data is linearly separable with a decision hyperplane specified by $\mathbf{w}$ and w_0, the data is also separable with the boundary given by $\tilde{\mathbf{w}}$ and $\tilde{w}_0$, where $\tilde{\mathbf{w}}\!=\!c\mathbf{w}$ and $\tilde{w}_0\!=\!cw_0$ for any scalar $c\!>\!0$.

b) What consequence does the above result have for maximum likelihood training of logistic regression for linearly separable data?

Acknowledgements

Some of the questions above were modified from those in a course developed by Iain Murray.