Machine Learning Exercises 7

Today you will reflect on multiclass classification with K-nearest neighbours and multinomial logistic regression. Both are examples of machine learning models that directly models the posterior class probabilities.

Note on visualising classifiers

Often it will make sense to make a plot showing either the true class or the predicted class for each observation in your dataset. More generally, you may wish to make a plot that shows the classification for *any* future data point that you could get, so as to visualise how the classification method assigns a class.

There are many options for visualising decision boundaries. A basic method is to colour the feature space according to the prediction that the classifier would make. This works particularly well for a two-dimensional feature space: Create a suitably fine grid over features, then for each point on the grid compute the classification and colour the point accordingly.

K-nearest neighbours

Exercise 1. Consider the tiny dataset in Figure 1 or draw a similarly simple one yourself.

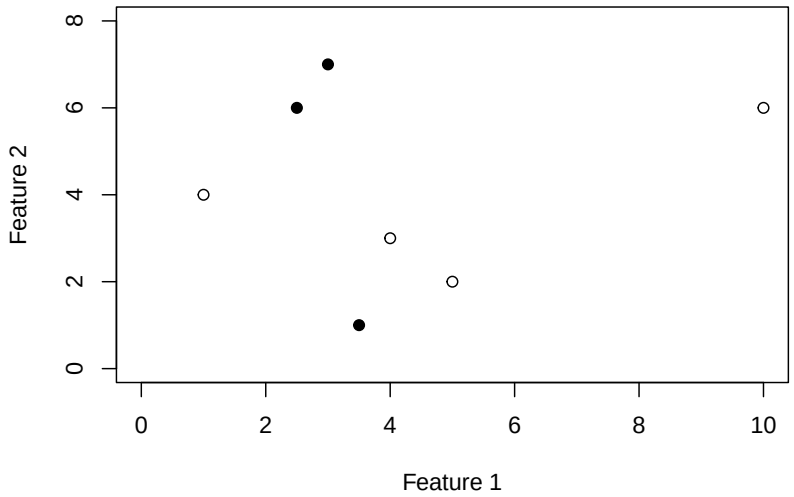

Figure 1: Dataset for binary classification.

A. Sketch by hand the decision boundaries for 1-nearest neighbour classification.

B. Give one or more examples of a new observation that under 3-NN would be classified as
   (a) *bullet* with a majority of 2 to 1.
   (b) *open circle* with a majority of 2 to 1.
   (c) *open circle* with a majority of 3 to 0.

C. Describe in terms of pseudocode how the algorithm can be implemented and reflect on the computational complexity of the method. For instance, consider the conceptually simple implementation that considers the distance to all training points each time a classification is made.

D. Explain why knn is an approximation to the Bayes classifier.

Exercise 2. Now use the two datasets Ex1-training.csv and Ex1-test.csv to build a knn classifier. There are two features (x_1 and x_2) and three classes (Black = 1, Red = 2, Blue = 3). The training data is seen in Figure 2.

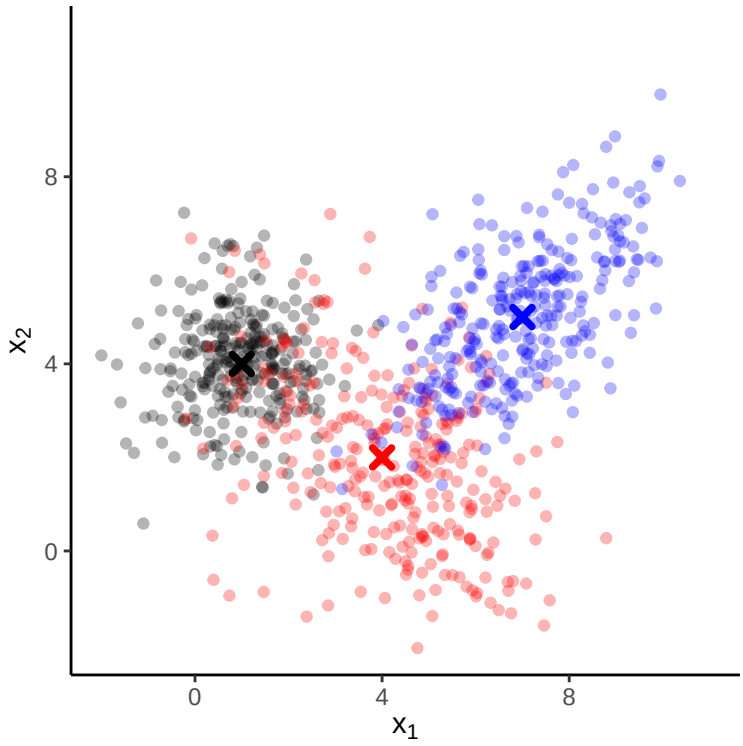

Figure 2: Training data with two features and three classes (Black = 1, Red = 2, Blue = 3).

A. Consider classification by 10-nearest neighbours. Compute the estimated posterior distribution of classes, $\widehat{\mathrm{P}}\left(Y = y | X = (x_1, x_2)\right)$, for a new observation with $(x_1, x_2) = (3, 3.5)$.

B. Classification by K-nearest neighbours is implemented by method `KNeighborsClassifier` from `sklearn.neighbors`. Train two classifiers, one based on the 5 nearest neighbours and the other based on the 10 nearest neighbours. Use the 10-NN classifier to check the result from A.

C. Visualise the two classifiers from B. by drawing their decision regions as described above.

D. For a suitable range of K plot the corresponding training and test errors (this is what we saw on Figure 2.17 from Introduction to Statistical Learning, where they plot against "flexibility" $1/K$). Describe what you see. Which K would you suggest based on this plot?

Multiclass classification with multinomial logistic regression

For this exercise, continue with the two datasets Ex1-training.csv and Ex1-test.csv.

Exercise 3. The *softmax function* takes a k-dimensional vector of real numbers $\eta = (\eta_1, \ldots, \eta_k)$ and returns a k-dimensional probability vector for the k classes as

$$\sigma(\eta_1, \ldots, \eta_k) = \left(\frac{e^{\eta_1}}{\sum_{i=1}^{k} e^{\eta_i}}, \ldots, \frac{e^{\eta_k}}{\sum_{i=1}^{k} e^{\eta_i}} \right)$$

Explain how multinomial logistic regression models posterior class probabilities via the softmax function.

Exercise 4. Train a multinomial logistic regression and visualise its decision regions. It is up to you how you want to include the features in the model (transformations? interaction term?)

You can use `LogisticRegression` from `sklearn` specifying `multi_class="multinomial"` – see page 150 in the Hands-on Machine Learning book, where they also excemplify how you can first specify the model without the data, then pass the dataset of interest (test/training) to the fitting method.