Machine Learning Exercises 21

Today is about building your own Naive Bayes classifiers. We use the two datasets Ex21-training.csv and Ex21-test.csv. It is of interest to predict the binary variable Y from the two continuous features X_1, X_2 and a discrete feature X_3.

For Naive Bayes, we model the joint (multivariate) distribution of features (given class) as a product of the marginal (univariate) distributions for each feature. So in each class y we have

$$p(x_1, x_2, x_3|y) = p(x_1|y)\, p(x_2|y)\, p(x_3|y)\,.$$

Exercise 1. First perform some exploratory data analysis and describe what you see, like we did in lectures.

You can get quite a good feel for data both from looking at scatter plots and by looking at the empirical covariance matrix for each class. Do features look independent? Do the covariance matrices seem similar for the two classes?

If there are many features, naturally we cannot make a scatter plot of all features jointly, whereas the covariance matrices will be useful for any number of features. Scatter plots of single features or pairs of features are also useful for investigations, but evidently do not tell the full story of how features vary jointly.

Estimating the univariate feature distributions

Given n observations, the histogram with binwidth h estimates a pdf $f(x)$ by the proportion of observations in the bin containing x scaled by the width of the bin (h):

$$\hat{f}(x) = \frac{\text{No. of observations in bin containing x}}{nh}$$

For the histogram, the bins are at *fixed positions*.

Exercise 2. Using the five data points drawn in Figure 1, sketch by hand two different histograms made with a binwidth of 1.

Alternatively, we can make a kernel density estimate using a "sliding window" so that $\hat{f}(x)$ is instead estimated by the proportion of observations in the window $[x-h; x+h]$ containing x, where h is the *bandwidth*[1]

$$\hat{f}(x) = \frac{\text{No. of observations in } [x-h; x+h]}{n \cdot 2h}.$$

So here the window is always *centered around the value* x that we are interested in finding the value $f(x)$ for, rather than non-overlapping bins at fixed values as in the histogram. The formula corresponds to placing a *kernel* at every single observation and then summing up their values measured at the point of interest x – hence the name. In this case our kernel is constant 1 in the interval $[x_0 - h; x_0 + h]$ for every single observation. So summing over all observations will exactly count the number of observations that are at most length h away from the point x of interest.

Exercise 3. Using the five data points drawn in Figure 1, sketch by hand the kernel density estimate you get by using instead a window of width 1 (i.e. bandwidth of 1/2) centered around the value of interest.

[1] Note that the exact definition of the parameter bandwidth can differ slightly between implementations, so some scaling may be applied to it internally. As the parameter is used only for tuning the granularity of the density estimate, its numerical interpretation is not of any importance.

In Python, you can use `KernelDensity` from scikit-learn with the 'tophat' kernel. If you want a smooth density estimate, then you can use instead the 'Gaussian' kernel. That would place a Gaussian bell curve in each observation rather than a constant function, so the further away from x they are, the less they contribute to the estimated pdf value $f(x)$.

Training a Naive Bayes Classifier

Exercise 4. Train a Naive Bayes Classifier on the data, where you use histograms to estimate the probability density functions for X_1 and X_2 and the probability mass function for X_3. You decide on the binwidth.

Exercise 5. Train a slightly different Naive Bayes Classifier, where you use instead a kernel density estimate (with "tophat kernel") to estimate the probability density functions for X_1 and X_2. You decide on the bandwidth h – the width of the window is $2h$. For X_3, you can use again the empirical probabilities.

Exercise 6. How could you go about choosing in a systematic way the parameter that is the bandwidth of the kernel density estimator?

Exercise 7. Compute the test error for your preferred Naive Bayes Classifier.

Exercise 8. Use your exploratory data analysis to discuss whether QDA and/or LDA are appropriate to use for these data. A key assumption of the LDA and QDA is that features are jointly Gaussian, which implies that each feature should follow a univariate Gaussian distribution. So it can be an idea to plot each feature separately, e.g. as a histogram: If the distribution does not look Gaussian, the joint distribution also will not be Gaussian. We use all of such investigations to guide the choice of model and further improvements.

Exercise 9. Use the two continuous features X_1 and X_2 to train a classifier based on either linear or quadratic discriminant analysis (LDA or QDA).

Exercise 10. How does K-nearest neighbours (with Euclidean distance) perform based on features X_1 and X_2 alone?

Exercise 11. Compare the performance of your preferred Naive Bayes Classifier, LDA/QDA and KNN.

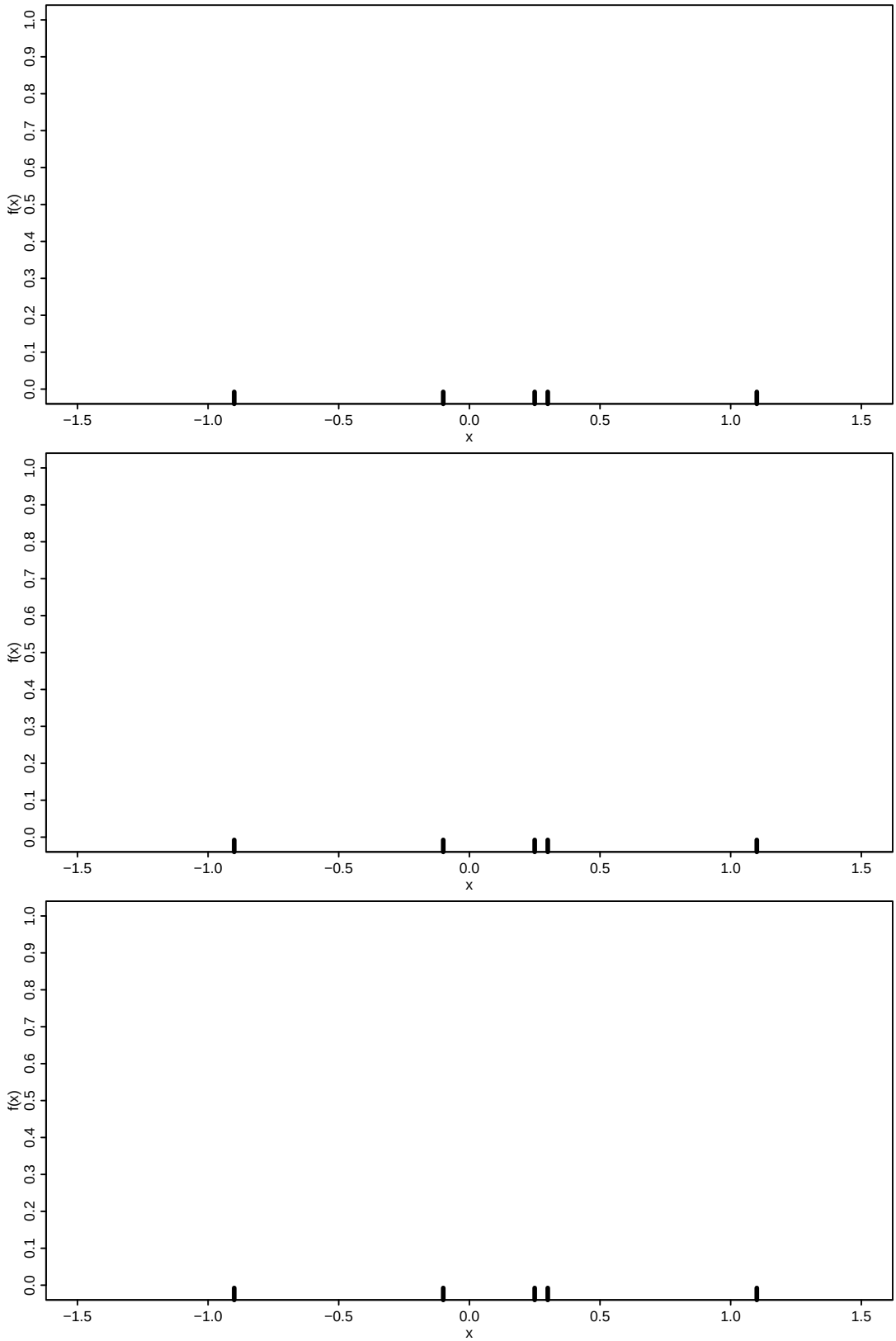

Figure 1: A simple dataset of five datapoints, as indicated on the x-axis. In the two top panels, draw two different histograms with a binwidth of 1. In the bottom panel, sketch a kernel density estimate that uses a constant kernel and a sliding window of width 1.