Assignment 2

Machine Learning: Algorithms and Theory

Prof. Ulrike von Luxburg / Diego Fioravanti / Moritz Haas / Tobias Frangen

Summer term 2018 — due **May 3rd**

Exercise 1 (kNN Implementation, 1+3+1+2+1+0 points)
In this exercise you will implement your own kNN classifier on a simulated dataset. In the file `Assignment02.ipynb` you will find a function called `get_gaussian2d_data_with_labels(n1, n2)` that produces a dataset $(X_i, Y_i)_{i=1..n}$ of $n = n_1 + n_2$ points. The X_i are sampled from 2 different 2D Gaussians and the Y_i are labels (1 or 2) denoting from which distribution was X_i sampled from.

(a) With $n_1 = n_2 = 25$ we store the training dataset in `train_data` and `train_labels`. `train_data` is a $n \times 2$-matrix where the i-th row represents $X_i \in \mathbb{R}^2$. `train_labels` is a $n \times 1$–matrix where the i–th row represents the class label $Y_i \in \{1, 2\}$. We plot the data using two different shapes for each class. We will use this data as our TRAINING SET. Generate a TEST SET with 100 data points of each class and plot it.

(b) Define a function `knnClassify` that takes as input `train_data`, `train_labels`, `test_data` and the number of neighbours k and returns the prediction `pred_labels` for `test_data`.

```
def knnClassify(train_data, train_labels, test_data, k=1)
```

Use this function to generate your predictions with $k = 3$ and plot them.

(c) Perform a reality check. In which cases will the prediction be correct and in which cases will errors occur? Write down your expectation. Afterwards, plot the test points using one colour if they are correctly classified and another if they are misclassified. Does the plot match your anticipations?

(d) We now want to evaluate the quality of our classifier. Write a function `empRisk` that takes as input the test data, test labels and your prediction and returns the EMPIRICAL RISK in terms of the $0-1$–loss.

```
def empRiskWith01loss(test_labels, pred_labels)
```

Plot the empirical risk for $k \in \{1, 3, 5, 7, 10, 15, 20\}$. Which k would you use in this setting?

(e) Generate a different training set with 500 data points for each class. Plot it and perform the same analysis as in (d). Which k performs best in that case? If it changed, can you explain why?

(f) *(Bonus)* Generate the training set with 1000 data points in each class. Do you observe any unexpected behavior? If yes, can you explain it?

Exercise 2 (Digit classification with kNN, 2+3 points)
We will now apply our classifier to a more realistic dataset from USPS. This dataset contains images of handwritten digits together with the value of the digit as labels (0 trough 9). Each image has 16×16 pixels and each pixel represents the brightness with values between 0 and 255. So the dimension of X is $16 \times 16 = 256$. We want to train a kNN classifier to recognise the digits that are shown in the images.

(a) Run the given code that you will find in `Assignment02.ipynb` for this exercise. This will load the USPS dataset and visualize a training point. As in the previous example, there is training and test data. Understand the structure of this dataset. Find out if the number of samples per label both in test and train is balanced or not, i.e. the amount of samples is the same for every digit.

(b) Apply your kNN classifier from Exercise 1 to this problem. Use different values for k and plot them as in Exercise 1 (d).

Depending on the implementation of your `kNN` classifier you may encounter runtime issues. If this happens you have to implement a more efficient calculation of the distances.

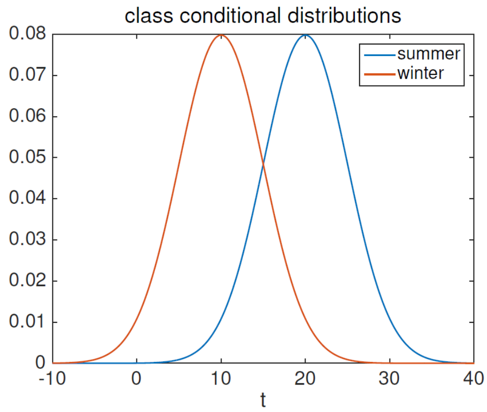
(Hint):

$$\|x-y\|^2 = \|x\|^2 - 2x^\top y + \|y\|^2 .$$

or you can import `euclidean_distances` from `sklearn.metrics.pairwise`.

Exercise 3 (Bayesian Decision Theory, 1 + 1 points)
Given a temperature measurement, we want to decide whether this measurement was made on a summer day or a winter day.

We assume that the density functions of the class conditional distributions $\mathrm{P}(\text{temp.} = t \mid j = \text{winter})$ and $\mathrm{P}(\text{temp.} = t \mid j = \text{summer})$ look as follows:



(a) Specify the decision rule for estimating whether its summer or winter based on the Maximum likelihood principle.

(b) Roughly sketch the posterior probabilities $\mathrm{P}(\text{winter} = j \mid \text{temp.} = t)$ and $\mathrm{P}(\text{summer} = j \mid \text{temp.} = t)$, and specify the decision function $f_{Bayes} : R \to \{summer, winter\}$ according to the Bayesian a posteriori criterion, assuming the following class prior probabilities:

- $\mathrm{P}(\text{summer}) = \mathrm{P}(\text{winter}) = 0.5$;
- $\mathrm{P}(\text{summer}) = 0.8$, $\mathrm{P}(\text{winter}) = 0.2$.

Exercise 4 (Empirical Risk Minimization and overfitting, 2+2+1+1 points)
To solve and understand the framework and assumptions of this exercise you may refer to Section 2.2 of the book *Understanding Machine Learning: From Theory to Algorithms* by Shalev-Shwartz and Ben-David (2014).

Suppose that we are given a finite number of examples $x_1, \ldots, x_n \in \mathbb{R}^d$ associated with labels $y_1, \ldots, y_n \in \{0, 1\}$. Consider the following predictor:

$$h_S(x) = \begin{cases} y_i & \text{if } \exists i \in \{1, \ldots, n\} \text{ s.t. } x = x_i \\ 0 & \text{otherwise.} \end{cases}$$

Namely, h_S "learned by heart" all the examples and associated labels.

(a) Prove that the empirical risk of h_S is zero for any given set of examples. In particular, note that this predictor may be chosen by an ERM algorithm.

(b) Suppose that the data is uniformly distributed on $[-2,2]^2$, and that the correct labels are assigned according to

$$f(x) = \begin{cases} 1 & \text{if } x \in [-1,1]^2 \\ 0 & \text{otherwise.} \end{cases}$$

Show that the error of the prediction rule h_S is $\frac{1}{4}$. We have found a predictor that performs well on the training data but not in the real world: this phenomenon is called OVERFITTING.

(c) Show that there exists a polynomial $p_S \in \mathbb{R}[X,Y]$ ($\mathbb{R}[X,Y]$ is the set of all polynomials in X and Y, so p_S can be written $p_S(X,Y) = a_{00} + a_{10}X + a_{01}Y + a_{11}XY + a_{20}X^2 + a_{02}Y^2 + \cdots$) such that $h_S(x) = 1$ if and only if $p_S(x) \geq 0$. Hence while the predictor h_S may seem somewhat unnatural, it can be learned while minimising the empirical risk on the class of thresholded polynomials.

(d) Can you think of a simple way to counter this overfitting phenomenon in the setting of the previous question?