WS '19/'20 / 03-ME-712.07 / Prof. Dr. Frank Kirchner — Handed in on: 03.12.2019

Machine Learning for Autonomous Robots — Estimated accumulated time: **30 hours**

Solution of **group 11**: Andras Szabo, Fabrizio D'Ascenzo, Livio Guidotti, Carlo Attardi — Exercise sheet 3

All source code is hosted here.

Problem 3.1 (Nearest Neighbour Classifier)

a) We choosed the following two distance measurement:
Chebysev distance
Manhatten distance

b) Classification accuracy of the classifier for all tested neighbourhood sizes (1 ...100)

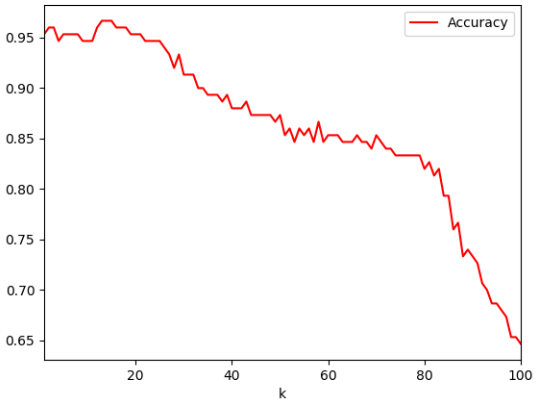
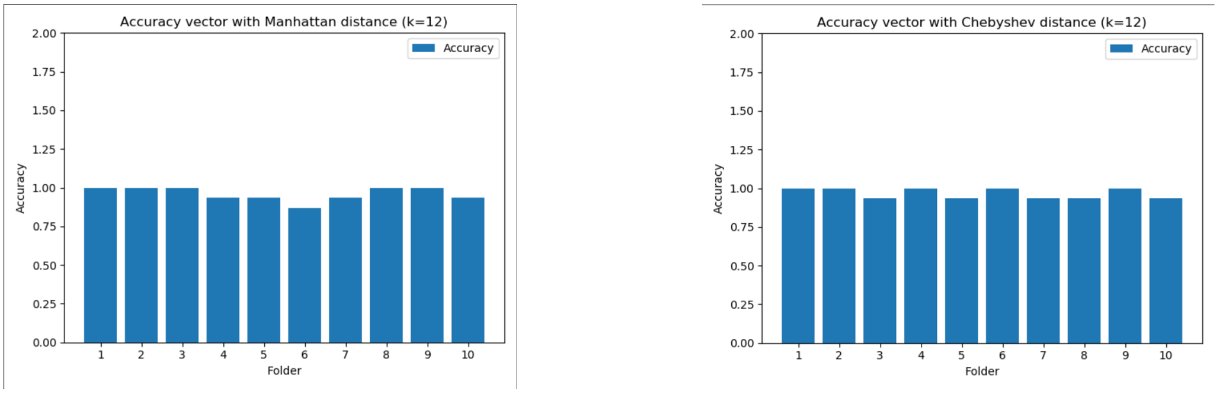


Abbildung 1: Classification accuracy for neighbours size 1..100, with 10-fold CV

The curve at the beginning go up slightly, until reaching his maximum in k=12, where the accuracy has a value of 0.9666666, and than decrease almost constantly.
It is explained by the fact that, by increasing the number of neighbrorns, are considered as neighborns of a sample in the testing set (considering the distance) also samples that are not actually "real"neighborns (i.e. not with the same class of the considered sample).
If we push the value of the neighbours size to the value of the training test (in our case 135) we are considering all the samples of the training test as neighborns, and that yields a random classification of the sample (a probability of about 0.3 of selecting the correct class), remembering that the considered sample is classified considering the most frequent class in the associated neighbours.

c) Classification with the best neighbourhood size: **12**



(a) Accuracy vector with Manhattan distance

(b) Accuracy vector with Chebysev distance

Abbildung 2: Accuracy vectors with the optimal neighbourhood size

The accuracy with the Manhattan distance is 0.9599999, that is lower than the one obtained with

the euclidean distance.
With the Chebyshev distance the obtained accuracy is 0.9666666, same as the one with the euclidean measurement.

Problem 3.2 (Bayesian classification - Optimality and minimum risk)

a) The two Gaussian distribution have the same variance, therefore $p(x|y_2)$ is same as $p(x|y_1)$ but centered in -1, hence the point in which they intersect should be in the middle of their means.

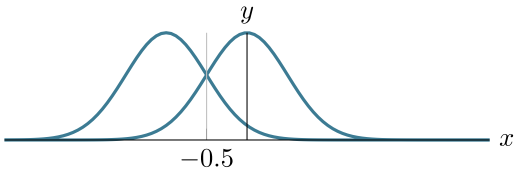


Abbildung 3: Expected threshold in -0.5

This hypotesis could be verified by calculating the equation

$$p(x|y_1) = p(x|y_2)$$

due to the assumption $p(y_1) = p(y_2)$, as it follows:

$$\frac{1}{\sqrt{\pi}}e^{-x^2} = \frac{1}{\sqrt{\pi}}e^{-(x+1)^2}$$

$$e^{-x^2} = e^{-(x+1)^2}$$

$$-x^2 = -(x+1)^2$$

$$x^2 = (x+1)^2$$

$$x^2 = x^2 + 2x + 1$$

$$2x = -1$$

$$x = -\frac{1}{2}$$

b) Compared to point (a), by applying the loss matrix given, a shift to the graph's left of the threshold is expected. This is caused due to the more importance given to the misclassification for $p(x|y_1)$

c) Here it is the calculations which demonstrate the previous intuition.

$$5p(x|y_1) = 3p(x|y_2)$$

$$5\frac{1}{\sqrt{\pi}}e^{-x^2} = 3\frac{1}{\sqrt{\pi}}e^{-(x+1)^2}$$

$$5e^{-x^2} = 3e^{-(x+1)^2}$$

$$\frac{5}{3} = e^{x^2-(x+1)^2}$$

$$\frac{5}{3} = e^{x^2 - x^2 - 2x - 1}$$

$$\frac{5}{3} = e^{-2x-1}$$

$$\frac{5e}{3} = e^{-2x}$$

$$\ln\left(\frac{5e}{3}\right) = -2x$$

$$\ln 5 - \ln 3 + \ln e = -2x$$

$$\ln 5 - \ln 3 + 1 = -2x$$

$$x = -\frac{1 + \ln 5 - \ln 3}{2}$$

d) By applying the loss matrix $L_{kj} = 1 - I_{kj}$ a threshold as calculated in point (a) is kept; assuming $p(y_1) = p(y_2)$, in this case the probability of a misclassification is the same for each class because of the fact that the posterior probability is equal.

In the case of different posterior probability the meaning of this matrix is clearly to choose the class with greater posterior probability, as if a loss matrix is not applied, because it does not assign significative values to a possible misclassification.

Problem 3.3 (Gradient Descent)

The gradient descent algorithm implementation source code is here.

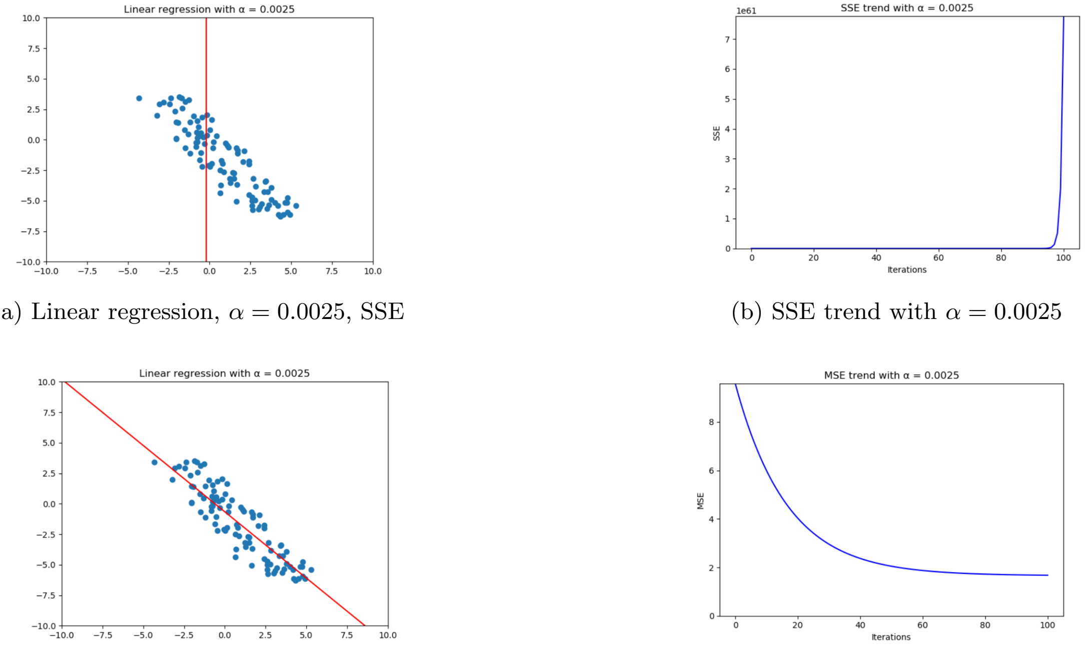
Problem 3.4 (Gradient Descent for Linear Regression)

a) The code for linear regression is here.

b) In our implementation of the linear regression with gradient descent we used, as loss function, the *sum of the squared error*, given by:

$$SSE(w) = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

where y_i is the actual value and $\hat{y}_i$ is the predicted value. Using this loss function and the given values of learning rate $\alpha = [0.0001, 0.001, 0.002, 0.0025]$, the algorithm converges for the first two values and doesn't converge for the others, how is possible to see in the figures 4a, 4b. In particular, with $\alpha = 0.0001$, the algorithm converges to $w^* = (-0.61597864, -1.15139026)$ with $SSE(w^*) = 165.2734$ and, with $\alpha = 0.001$, it converges to $w^* = (-0.59606998, -1.15497755)$ with $SSE(w^*) = 165.2379$. The divergence of the algorithm for the others two values of the learning rate suggests that the gradient descent overshoot the minimum of the loss function because the step done is "too large". This fact demostrates the importance of choosing a correct learning rate depending on the loss function that is used. Infact, for example, using the *Mean squared error* (MSE) instead of the SSE, how we can see in the fig. 4c, 4d, the algorithm converges also for the last two values, but doesn't reach the optimal value (with 100 iterations) for the opposite reason (now the step is too small).



(a) Linear regression, $\alpha = 0.0025$, SSE

(b) SSE trend with $\alpha = 0.0025$

(c) Linear regression, $\alpha = 0.0025$, MSE

(d) MSE trend with $\alpha = 0.0025$

Abbildung 4: Different behaviour SSE/MSE with $\alpha = 0.0025$

c) We can estimate the standard deviation of the noise ϵ from $SSE(w^*)$ with the *standard error of the model*:

$$\sigma = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

Infact, with linear regression, we are searching a function $\hat{y}(x) = w_0 + w_1 x$ that models the relation:

$$y(x) = w^T x + \epsilon$$

where ϵ, in our case, is a noise normally distributed.
So, if we can find an optimal value $w^* = (w_0, w_1)$, the quantity given by:

$$\sqrt{\frac{1}{n-1}SSE(w^*)}$$

could be a good estimation of the *standard deviation* σ of the noise ϵ. Thus, because $SSE(w^*) \approx 165$, $\sigma \approx 1.29$.