University of Stavanger

Stavanger, November 19, 2021

Laboratory exercise 3

ELE520 Machine learning

A PDF version of the report of the student solution to the exercise including figures and answers to questions shall be submitted on CANVAS.[1]

In this problem the purpose is to visualise the use of parametric and non parametric estimation techniques for the minimum error rate classifiers from theoretical exercise 3. Training data are stored in the *pickle* file *lab3.p* and can be downloaded from CANVAS.

For both the estimation approaches it might be useful to apply *norm2D* to compute the estimated denisty function values. (This demands some considerations for the Parzen-technique). For the nearest neighbourhood method you have to make a new function.

It is recommended to make a script, e.g. *labsol3.ipynb*, for the commands needed for solving the problems. As for the previous laboratory exercise, you might find the code listing of pseudo code for a proposed solution shown in listing 1 useful. See the previous laboratory exercise for further comments on iterations, conditions and data structures. Also see the solution proposal for the actual python code which also includes a function for plotting.

Problem 1

a) Estimate $\boldsymbol{\mu}_i$ from each data set $\mathcal{X}_i$ for $i = 1, 2$. (Hint: Use the numpy command `mean`.)

b) Estimate $\boldsymbol{\Sigma}_i$ from each data set $\mathcal{X}_i$ for $i = 1, 2$.(Hint: Use the numpy command `cov`.)

c) Estimate the discriminant function (scaled probability density function) for class ω_1 and plot this with red surface color (`facecolor='r';`). Define 25 points of computation along each axis so that you will be using 25^2 points of computation.

d) Estimate the discriminant function for class ω_2 and plot it with blue surface (`facecolor='b';`) so that the two functions are shown in the same figure.

[1] If it is not possible to export the Jupyter notebook to PDF, the ipynb and py files can be submitted.

Trygve Eftestøl, Professor
Faculty of Science and Technology
Department of Electrical and Computer Engineering
Kjølv Egelands hus
University of Stavanger
N-4036 Stavanger.
Telephone: +47 51 83 10 00.
Telefax: +47 51 83 17 50.
E-mail: trygve.eftestol@uis.no
www.ux.his.no/~trygve-e

e) Identify the decision border and decision regions. Compare with the discriminant functions that were plotted in laboratory exercise 2.

f) Repeat subtaske c-d for the Parzen classifier with window size $h_1 = 0.5$.

g) Repeat the previous subtask for the Parzen classifier with window size $h_1 = 5.$.

h) Repeat the previous subtask the k_N-nearest neighbour classifier with $k_n = 1$.

i) Repeat the previous subtask the k_N-nearest neighbour classifier with $k_n = 3$.

j) Repeat the previous subtask the k_N-nearest neighbour classifier with $k_n = 5$. (Why will this not work?)

k) Add functionality so that the figure display the a posteriori probability for the two classes.

Comments to the code listing Note that the data are loaded every time the labsol3 function is called. It might be an option to load the data in the notebook script, and then call the function with modified to pass the data into the function upon call.

```python
import sys
import getopt
import numpy as np
import scipy.io as io
from pdffuns import norm2D, knn2D, classplot
import pickle


def labsol3(met='ML', discr='pxw', prm = []):

    # Initialise values:
    # - axes, x1 and x2

    # Load data from pickle files
    # - load into variable X, X[k], k = 0,...,M-1
    # - determine number of classes, M, from X
    # - determine feature dimension, l, from X[0]

    # Estimate prior probabilities, Pwi[k].
    # - iterate over classes, k = 0,...,M-1
    #  - determine number of feature vectors, N[k]
    # - determine Pwi

    # Initialise method specific parameters
    # - on condition of met
    #  - Maximum likelihood: my, Sgm (empty)
    #  - Parzen window: h1 from prm
    #  - kn nearest neighbor: knn from prm

    # - parameters, my[i] and Sgm[i], i = 0,...,M-1
    # - prior probabilities, Pw[i], i = 0,...,M-1

    # Determine class specific probability density functions, pxw[
    i], i = 0,...,M-1
    # - initialise pxw as empty list
    # - initialise total density function, p as zero
    # - iterate over classes, k = 0,...,M-1
    #  - on condition of met:  Maximum likelihood:
    #   - estimate parameters my[k], Sgm[k]
    #   - determine pxw[k] by using norm2D
    #  - on condition of met: Parzen window:
    #   - initialise pxw[k]
    #   - determine hn
    #   - iterate over samples in X[k], i = 1,...,N[k]
    #    - let x be feature vector number i in X[k]
    #    - use norm2D to determine pN
    #    - update pxw[k] by adding pN
    #   - update pxw by dividing by N[k]
    #  -  on condition of met: kn-nearest neighbor:
    #   - use knn2D to determine pxw[k] from X[k]
    #  - update p

    # Determine discriminant functions, g[k], k = 0,...,M-1
```

```
53      # - initialise g as empty list
        # - iterate over classes, k = 0,...,M-1
55      #  - on condition of discr determine selected discriminant
        function
        #   - Scaled pdfs
57      #   - Posterior probability
        #   - pdfs (not really discriminant functions)
59
        # Plotting
61      # - plot discriminant functions
```

Code Listing 1: labsol3.py