Prof. Dr. Frank Deinzer
Technical University of Applied Sciences
Würzburg-Schweinfurt

Portfolio Exam 1

Portfolio 1 Tasks (Tentative date: 14.12.2023)

Portfolio 1 is based on the exercises from the previous semester. You can use the program code from the previous semester if you want.

The exam will consist of us talking one-on-one about some randomly selected tasks, discussing your program code, and having a conversation about the background of the tasks.

The deliverable for this task is the source code of your implementations as basis for the discussion. Send it as email latest **one day before** the examination day.

Task P1.1

Multinomial distributed random numbers

a) Create a multinomial distribution from the example file `anthrokids.csv`: Use the column *age* and round the floating point values to obtain integers. Count the frequencies of the integers and use them as basis of the distribution.

b) Sample from the multinomial distribution.

c) How can you check if your samples really follow the distribution from 1 a)?

Task P1.2

Uniformly distributed random numbers

a) Implement a way to generate uniformly distributed one-dimensional random numbers. Make sure that your implementation can be parametrized.

b) Use the random number generator from 2 a) to create a sample set of random numbers. Estimate the parameters of the underlying uniform distribution from this sample set. What do you observe? How accurate is your parameter estimate? What happens if the size of the sample set changes?

c) How can you check if your random numbers really follow the distribution from 2 a)?

Task P1.3

Normally distributed random numbers

a) Implement a way to generate normally distributed one-dimensional random numbers with mean 0 and standard deviation of 1.0.

b) Use the random number generator from 3 a) to create a sample set of random numbers. Estimate the parameters of the underlying normal distribution from this sample set. What do you observe? How accurate is your parameter estimate? What happens if the size of the sample set changes?

c) How can you check if your random numbers really follow the distribution from 3 a)?

Task P1.4

Eponentially distributed random numbers

a) Implement a way to generate exponentially distributed one-dimensional random numbers. Make sure that your implementation can be parametrized.

b) Use the random number generator from 4 a) to create a sample set of random numbers. Estimate the parameters of the underlying exponential distribution from this sample set. What do you observe? How accurate is your parameter estimate? What happens if the size of the sample set changes?

c) How can you check if your random numbers really follow the distribution from 4 a)?

Task P1.5

Normally distributed multivariate random numbers

a) Implement a way to generate normally distributed random vectors of dimensions 2, 5 and 10 with mean vector $\boldsymbol{\mu} = \mathbf{0}$ and covariance matrix $\boldsymbol{\Sigma} = \boldsymbol{I}$.

b) Use the random vector generator from 5 a) to create a sample set of random vectors. Estimate the parameters of the underlying normal distribution from this sample set. What do you observe? How accurate is your parameter estimate? What happens if the size of the sample set changes?

c) Implement a way to generate normally distributed random vectors of different dimensions with mean vectors $\boldsymbol{\mu} \neq \mathbf{0}$ and covariance matrices $\boldsymbol{\Sigma} \neq \boldsymbol{I}$ of your choice.

d) Use the random vector generator from 5 c) to create a sample set of random vectors. Estimate the parameters of the underlying normal distribution from this sample set. What do you observe? How accurate is your parameter estimate? What happens if the size of the sample set changes? What if the dimensionality changes?

Task P1.6

Basic Importance Sampling example

a) Given a set of N random numbers x that follow a standard normal distribution $x \sim \mathcal{N}(0.0, 1.0)$.

Estimate the probability that these random numbers satisfy the condition $x < \theta$ for any θ with and without Importance Sampling. Think about suitable Importance Sampling proposal functions.

b) Plot the estimation result using the two techniques (with and without Importance Sampling) as a function of the parameter θ in a meaningful way.

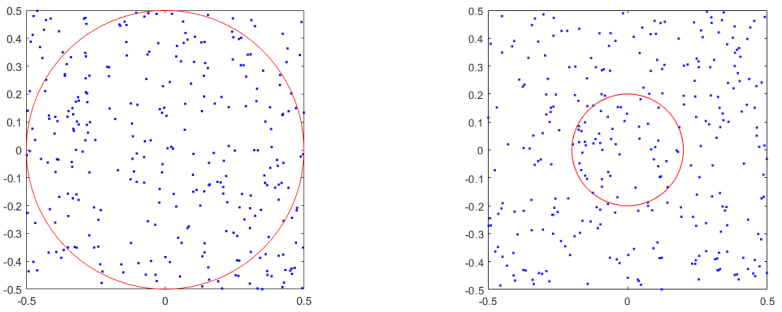

Fig. 1: Square with side length 1, circles with radius $r = 0.5$ resp. $r = 0.2$ and uniformly distributed samples within the square.

Task P1.7

Circle area estimation using Importance Sampling

a) Given a square with side length 1. Place a circle of varying radius $r \leq 0.5$ in the center of this square. Generate uniformly distributed random samples $\boldsymbol{x}$ in the area of the interior of the square. See Fig. 1 for an illustration how the scenario could look like.

b) Estimate the area covered by the circle by counting the percentage of samples that fall into the interior of the circle. I.e. all samples that have a distance smaller than r from the center of the circle. How accurate is your estimate compared to the real area $r^2\pi$ of the circle?

c) What happens to your estimation when r gets very small (e.g. $r = 10^{-10}$)? Find a solution for the problem using Importance Sampling. Use and compare two different proposal distributions.

d) Think about the proper number of samples required for a stable estimate of the area. Find a quality measure for the number of samples and plot your quality as a function of the number of samples.

Task P1.8

Realize an implementation of the EM algorithm in a programming language of your choice. Do not use a pre-built implementation. A description of the algorithm is part of the lecture, but can also be found in [2].

Test your implementation on the following scenarios, especially examining the accuracy and stability of the estimate. Vary the initialization of the parameters.

a) Generate a synthetic sample set from a mixture distribution of your choice: n mixture components and correspondingly many multidimensional normal distribution components. Estimate the mixture parameters from this sample set using the correct number of mixture components

b) Change the mean vectors and the covariance matrices for the mixture components. Does this affect the EM algorithm?

c) Vary the size of the sample set and repeat the scenarios from 8 a) and 8 b). What do you observe?

d) Vary the number of mixture components using more or less then used to create the sample set. What do you observe?

e) What happens to the EM estimation process if you significantly increase the dimensionality of the distribution?

Task P1.9

Exploring more exercise data.

a) The file `Examples.zip` contains the three files `Example1.csv`, `Example2.csv` and `Example3.csv`. These are sample sets. Load each of the `.csv` files.

Each *column* contains one sample of the sample set. The first row contains the class number Ω_κ. Rows two and three contain the two-dimensional feature vectors $\boldsymbol{c}$.

For each file, plot all the features with different colors for the classes.

b) A very famous sample set is *Fisher's Iris data set* [3]. The data set consists of 50 samples from each of the three flower species Iris Setosa, Iris Virginica and Iris Versicolor. Four features were measured from each sample: the length and the width of the sepals and petals, in centimeters.

The file `iris-numclass.csv` (in zip file `FisherIris.zip`) contains all 150 samples of the sample set. Here each *row* contains one sample. The class number in column 1 contains the species (Iris

setosa, Iris virginica and Iris versicolor). The columns 2-5 contain the measurements of each sample (sepal length, sepal width, petal length, petal width).

Plot all the features (maybe in different combinations of 2 features) with different colors for the classes.

Task P1.10

The following excercise is based on the large data set from [1], available in the file `anthrokids.csv`. It contains many body measurements from children and adolescents.

a) Split the data set into two sets: one for male and one for female persons. We will only use the columns for the age as classes Ω_κ and the height measurements as feature c.

b) Split the male and the female datasets into training and test data sets. A ratio of 2:1 is suitable.

c) Use the training data sets to estimate the parameters for all classes: Estimate the parameters of normal distributions $p(c|\Omega_\kappa)$ for different ages (classes Ω_κ) given the features c (height measurements) of this class. Calculate the prior probabilities $p(\Omega_\kappa)$ of all classes. You might want to try with classes for ages of $3, 4, \ldots, 18$.

d) Go through all the data in the testing data sets and classifiy them with the optimal Bayes classifier:

- Use the feature c (height measurement) to evaluate the normal distributions $p(c|\Omega_\kappa)$ of all possible classes.
- Make a decision for the class Ω_κ that maximizes the posterior $p(\Omega_\kappa|c)$.
- Check if the decision is correct using the known real age.

e) At the end calculate the overall recognition rates for the male and female data sets. Are you satisfied with the result?

Task P1.11

The following excercise is based on the Iris data and the data from `Examples.zip`.

a) Split each data set into training and test data.

b) Implement a Bayes classifier that estimates a two-dimensional normal distribution for each class for the data sets from `Examples.zip`. Use them to classify the test samples. Are you satisfied with the classification results?

c) Implement a Bayes classifier using normal distributions for the Iris data set. Vary the used features from using all 4 measurements per sample down to using only 1 of the measurements. Which of the possible feature combinations perform best and which worst? What conclusions do you draw from the results?

d) Look for ways to visualize how your classifiers work in the case of two-dimensional features. The important thing here is: Which areas of the feature space are assigned to which class?

Task P1.12

We will try to improve the classification results from Exercise 11.

The critical point with the previous classifier is the choice of the underlying density function. If a unimodal density does not produce satisfying results, one can turn to mixture distributions.

Replace the classifier from Exercise 11 with one that estimates a mixture distribution in training. What results do you get with this classifier? How do you choose the number of mixture components?

Task P1.13

Comparing classifier performance.

a) Implement a classifier based on the idea of Parzen estimation.

For this purpose, realize a function to compute $p(\boldsymbol{c}|\Omega_\kappa)$ according to the idea of a Parzen estimation (kernel density estimation). Your function should take as input a classified sample set, the class κ to be evaluated, the covariance matrix $\boldsymbol{\Sigma}$ and of course the feature vector $\boldsymbol{c}$.

You can use $p(\boldsymbol{c}|\Omega_\kappa)$ and $p(\Omega_\kappa)$ to perform a Bayesian classification.

b) Implement a Nearest Neighbor classifier.

Write a function that takes a classified sample set, a parameter m, and the feature vector $\boldsymbol{c}$. Your function should now search for the m nearest neighbors within the sample set to the feature vector $\boldsymbol{c}$. Use this to calculate the probabilities $p(\Omega_\kappa|\boldsymbol{c}) = \frac{m_\kappa}{m}$ by counting within these neighbors the memberships m_κ to each class.

You can use $p(\Omega_\kappa|\boldsymbol{c})$ to perform a Bayesian classification.

c) You have previously examined the data sets from *Fisher's Iris data set* [3] and from `Examples.zip`. Now use these to evaluate the performance of different classification methods. Compare the achievable classification rates for the following classifiers:

- Simple normal distribution classifier with a unimodal, multivariate normal distribution as the underlying density. This corresponds in essence to the classification exercises for the Kids' Size Problem.
- Classification based on a Parzen estimate.
- Classification with the Nearest Neighbor classifier.
- If you have other classifiers available, feel free to include and compare their results.

Which classification methods would you choose for a practical application depending on the data set?

d) Take the entire *Fisher's Iris data set* [3] and ignore the membership of the samples to the different flower species.

Use your EM algorithm to estimate a mixture distribution with 3 components from this data. How do the individual mixture components relate to the species data? For visualization, you can limit yourself to 2 dimensions (sepal length/width or petal length/width).

References

[1] STAT 3202: Group Project I. `https://daviddalpiaz.github.io/stat3202-au18/project/proj-01/proj-01-E.html`, Autumn 2018, OSU.

[2] Marc Peter Deisenroth, A. Aldo Faisal, and Cheng Soon Ong. *Mathematics for Machine Learning*. Cambridge University Press, 2020.

[3] Ronald Aylmer Fisher. The Use of Multiple Measurements in Taxonomic Problems. *Annals of Eugenics*, 7(2):179–188, 1936.