Statistical Machine Learning: Exercise 3

Linear Regression, Linear Classification and Principal Component Analysis

Prof. Dr. Kristian Kersting, Karl Stelzner, Claas Voelcker

TECHNISCHE UNIVERSITÄT DARMSTADT

Summer Term 2020

Due date: Jun 21th, 23:59 PM

Task 1: Bayesian Decision Theory (20 Points)

In this exercise, we consider data generated by a mixture of two Gaussian distributions with parameters $\{\mu_1, \sigma_1\}$ and $\{\mu_2, \sigma_2\}$. Each Gaussian represents a class labeled C_1 and C_2, respectively.

1a) Optimal Boundary (4 Points)

Explain in one short sentence what Bayesian Decision Theory is. What is its goal? Consider the case of two classes C_1 and C_2. Which condition holds at the optimal decision boundary? When do we decide for class C_1 over C_2?

1b) Decision Boundaries (8 Points)

If both classes have equal prior probabilities $p(C_1) = p(C_2)$ and the same variance $\sigma_1 = \sigma_2$, derive the decision boundary x^* analytically as a function of the two means μ_1 and μ_2.

1c) Different Misclassification Costs (8 Points)

Assume $\mu_1 > 0$, $\mu_1 = 2\mu_2$, $\sigma_1 = \sigma_2$ and $p(C_1) = p(C_2)$. If misclassifying sample $x \in C_2$ as class C_1 is four times more expensive than the opposite, how does the decision boundary change? Derive the boundary analytically. (There is no cost for correctly classifying samples.)

Task 2: Density Estimation (45 Points)

In this exercise, you will use the datasets `densEst1.txt` and `densEst2.txt`. The datasets contain 2D data belonging to two classes, C_1 and C_2.

2a) Gaussian Maximization Likelihood Estimate (10 Points)

Derive the ML estimate for the mean and covariance of the **multivariate** Gaussian distribution. Start your derivations with the function you optimize. Assume that you can collect i.i.d data. (Hint: you can find many matrix identities on the Matrix Cookbook and at `http://en.wikipedia.org/wiki/Matrix_calculus`.)

2b) Prior Probabilities (2 Points)

Compute the prior probability of each class from the dataset.

2c) Biased ML Estimate (5 Points)

Define the bias of an estimator and write how we can compute it. Then calculate the biased and unbiased estimates of the conditional distribution $p(x|C_i)$, assuming that each class can be modeled with a Gaussian distribution. Which parameters have to be calculated? Show the final result and attach a snippet of your code. Do not use existing functions, but rather implement the computations by yourself!

2d) Class Density (5 Points)

Using the unbiased estimates from the previous question, fit a Gaussian distribution to the data of each class. Generate a single plot showing the data points and the probability densities of each class. (Hint: use the contour function for plotting the Gaussians.)

2e) Posterior (8 Points)

In a single graph, plot the posterior distribution of each class $p(C_i|x)$ and show the decision boundary.

2f) [Bonus] Bayesian Estimation (15 Points)

State the generic case of Bayesian linear regression with data $< \vec{X}, \vec{Y} >$ and parameters $\vec{\theta}$. What do we assume about the data, the model and the parameters?
Formulate the posterior distribution for your model parameters given the data, i.e., $p(\vec{\theta}|\vec{X}, \vec{Y})$, and derive its mean and covariance, assuming that the model of the output variable is a Gaussian distribution with a fixed variance.
What do we do when we want to predict a new point?
Which are the advantages of being Bayesian?

Task 3: Non-parametric Density Estimation (20 Points)

In this exercise, you will use the datasets `nonParamTrain.txt` for training and `nonParamTest.txt` for evaluating the performance of your model.

3a) Histogram (4 Points)

Compute and plot the histograms using 0.02, 0.5, 2.0 size bins. Intuitively, indicate which bin size performs the best and explain why. Knowing only these three trials, would you be sure that the one you picked is truly the best one? Attach the plot of the histograms and a snippet of your code.

3b) Kernel Density Estimate (6 Points)

Compute the probability density estimate using a Gaussian kernel with $\sigma = 0.03$, $\sigma = 0.2$ and $\sigma = 0.8$. Compute the log-likelihood of the data for each case, compare them and show which parameter performs the best. Generate a single plot with the different density estimates and attach it to your solutions. Plot the densities in the interval $x \in [-4, 8]$, attach a snippet of your code and comment the results.

3c) K-Nearest Neighbors (6 Points)

Estimate the probability density with the K-nearest neighbors method with $K = 2, K = 8, K = 35$. Generate a single plot with the different density estimates and attach it to your solutions. Plot the densities in the interval $x \in [-4, 8]$, attach a snippet of your code and comment the results.

3d) Comparison of the Non-Parametric Methods (4 Points)

Estimate the log-likelihood of the testing data using the KDE estimators and the K-NN estimators. Why do we need to test them on a different data set? Compare the log-likelihoods of the estimators w.r.t. both the training and testing sets in a table. Which estimator would you choose?

Task 4: Expectation Maximization (20 Points)

In this exercise, you will use the datasets `gmm.txt`. It contains data from a Gaussian Mixture Model with four 2-dimensional Gaussian distributions.

4a) Gaussian Mixture Update Rules (2 Points)

Define the model parameters and the update rules for your model. Specify the E- and M-steps of the algorithm.

4b) EM (18 Points)

Implement the Expectation Maximization algorithm for Gaussian Mixture Models. Initialize your model uniformly. Generate plots at different iterations $t_i \in [1, 3, 5, 10, 30]$, showing the data and the mixture components, and plot the log-likelihood for every iteration $t_i = 1 : 30$. Attach a snippet of your code.