

Foundations of Data Analysis - WS21

Pen and paper assignment

Supervised learning

Due date: 9:45 am on 27.10.2021

Description and instructions

The maximum number of points achievable in this assignment is 100. Kindly follow the submission instructions carefully as failing to do so will result in a penalty.

- Calculations and derivations must be shown in detail along with sufficient explanatory text.
- You are allowed to work on this assignment either individually or in groups. If you work in a group, be sure to include the names of all those who you have worked in your report. Your report must always be prepared individually!
- Remember to cite every external source that you use. Calculators and plotting tools are not required for this assignment and hence their use is discouraged. However, if you must use them, please mention this in the relevant sections.
- Any act of plagiarism will be taken very seriously and handled according to university guidelines.
- Upload your submission to Moodle as a single PDF comprising photos of your (legibly) hand-written work or a document that has been typeset in L^AT_EX.
- This PDF should be named `<last_name>.pdf`, replacing `<last_name>` with your last name(s).

Do not hesitate to email me (Akshey Kumar) at akshey.kumar@univie.ac.at or post on the discussion forum on Moodle with any questions you may have.

Introduction

In the lectures you have learnt how to construct different kinds of binary classifiers. Let's say that you are given a dataset that contains m samples of n -dimensional feature vectors. In the binary classification setting, each feature vector $\mathbf{x} \in \mathbb{R}^n$ is assigned to one of two classes which is denoted by a label $y \in \{0, 1\}$. However, we need not be restricted to datasets that contain only two class labels. In general, y could take on multiple values, so we have $y \in \mathcal{Y} = \{1, 2, \dots, C\}$. We shall call this the *multi-class learning paradigm* where our aim is to learn a hypothesis $h : \mathbb{R}^n \rightarrow \mathcal{Y}$.

In this assignment you will learn how to construct classifiers for supervised learning when there are more than two classes. By constructing a multi-class LDA classifier you will learn to build multi-class classifiers by utilising binary classifiers in two different ways: the one-vs-all classifier and the pairwise classifier.

1 Mean and covariance

Given below is a dataset $\mathcal{S} = \{(\mathbf{x}_i, y_i)\}_{i \in \{1, \dots, n\}}$ with $\mathbf{x}_i = (x_i^1, x_i^2) \in \mathbb{R}^2$ and $y_i \in \{1, 2, 3\}$.

$$\mathcal{S} = \{((1, 2)^\top, 1), ((1, 3)^\top, 1), ((5, 3)^\top, 1), ((4, 2)^\top, 2), ((6, 1)^\top, 2), ((8, 2)^\top, 2), ((3, 5)^\top, 3), ((7, 5)^\top, 3), ((7, 4)^\top, 3)\}$$

- (a) (5 points) Plot the given samples in a scatter plot, which shows x^1 and x^2 on the axes of the plot and y as differently coloured markers. Carefully label all axes!
- (b) (10 points) Calculate the sample mean vector $\boldsymbol{\mu}$, the sample covariance matrix $\boldsymbol{\Sigma}$, as well as the class sample mean vectors $\boldsymbol{\mu}_1$, $\boldsymbol{\mu}_2$ and $\boldsymbol{\mu}_3$. Add $\boldsymbol{\mu}$, $\boldsymbol{\mu}_1$, $\boldsymbol{\mu}_2$ and $\boldsymbol{\mu}_3$ to your plot.

Hints:

$$\boldsymbol{\mu} = \frac{1}{n} \sum_{i=0}^n \mathbf{x}_i \quad (\boldsymbol{\mu} \in \mathbb{R}^2)$$

$$\Sigma_{jk} = \frac{1}{n-1} \sum_{i=0}^n (x_i^j - \mu^j)(x_i^k - \mu^k) \quad (\boldsymbol{\Sigma} \in \mathbb{R}^{2 \times 2}, j, k \in \{1, 2\})$$

$$\boldsymbol{\mu}_j = \frac{1}{|\{i | y_i = j\}|} \sum_{\{i | y_i = j\}} \mathbf{x}_i \quad (\boldsymbol{\mu}_j \in \mathbb{R}^2, j \in \{1, 2, 3\})$$

2 Constructing multi-class classifiers

Recall that the hyperplane defined by the equation $\mathbf{w}^\top \mathbf{x} + b - c = 0$ represents the decision boundary of the two-class LDA classifier, where $\mathbf{w} \in \mathbb{R}^d$, $b \in \mathbb{R}$ and $c \in \mathbb{R}$. The corresponding hypothesis function is

$$h_{\text{LDA}}(\mathbf{x}) = \mathbf{1}_{\{\mathbf{w}^\top \mathbf{x} + b - c \geq 0\}}, \quad (1)$$

where

$$\mathbf{1}_A(x) = \begin{cases} 1 & \text{if } x \in A \\ 0 & \text{otherwise} \end{cases}$$

is the indicator function. Note that this is a slightly different notation than in the lecture, where the sign function was used (which assigns -1 and 1). Both definitions are equivalent, but the indicator function definition is a little more convenient for the following exercises. Using this binary classifier as the basis, we shall consider two different types of classifiers, one-vs-one (OVO) and one-vs-all (OVA).

For the following exercises, consider the dataset shown in Figure 1. It consists of two features x^1 and x^2 and three classes denoted by the label $y \in \{1, 2, 3\}$. There are 500 samples in each class. For your ease of calculation, the mean and covariance matrix of each class has been already estimated and are as follows,

$$\begin{aligned} \boldsymbol{\mu}_1 &= (0, 0)^\top \\ \boldsymbol{\mu}_2 &= (0, 6)^\top \\ \boldsymbol{\mu}_3 &= (6, 0)^\top \\ \boldsymbol{\Sigma}_1 &= \boldsymbol{\Sigma}_2 = \boldsymbol{\Sigma}_3 = \begin{pmatrix} 1 & 0.5 \\ 0.5 & 1 \end{pmatrix} \end{aligned}$$

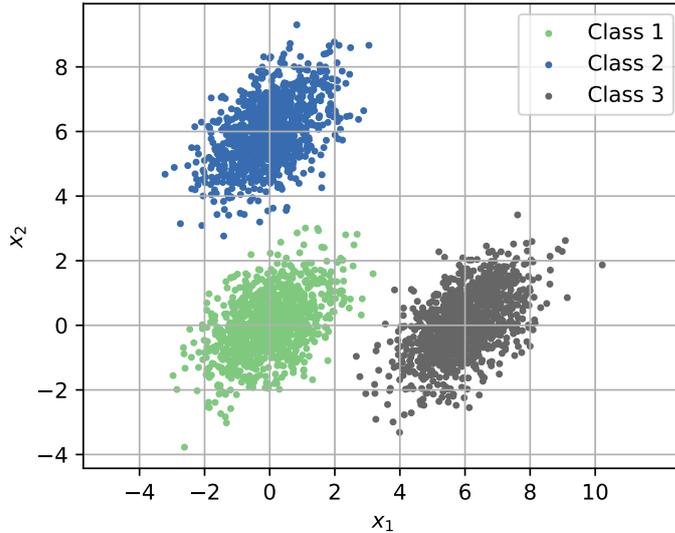


Figure 1: Example data

2.1 One versus one classifier (OVO)

To construct an OVO classifier (also known as the pairwise classifier) we start by building binary classifiers between every pair of classes. This results in a total number of $C(C - 1)/2$ classifiers, where C is the number of classes in the dataset. In the case of LDA, the binary classifier between class i and j is

$$h_{ij}(\mathbf{x}) = i \cdot \mathbf{1}_{\{\mathbf{w}_{ij}^\top \mathbf{x} + b_{ij} - c_{ij} < 0\}} + j \cdot \mathbf{1}_{\{\mathbf{w}_{ij}^\top \mathbf{x} + b_{ij} - c_{ij} \geq 0\}}. \quad (2)$$

To classify a given data point \mathbf{x}_i , we let all the classifiers “vote” and choose the class which gets the highest number of votes. For example, in case of a three-class classifier, if we have

$$h_{12}(\mathbf{x}_i) = 2, \quad h_{23}(\mathbf{x}_i) = 2, \quad h_{13}(\mathbf{x}_i) = 1$$

we would choose $y_i = 2$ for the given data point \mathbf{x}_i . *Note:* In the case where more than one class gets the maximum numbers of votes, we can randomly choose among these classes to break such “ties”.

- (5 points) Does the given dataset (Figure 1) fulfill the assumptions of LDA, if we were to construct a binary classifier between every pair of classes? Justify your answer!
- (20 points) Perform LDA on every pair of classes. Estimate the parameters w_{ij} , b_{ij} and c_{ij} of the decision boundary between class i and j . Write down $h_{ij}(\mathbf{x})$ in the form of equation (2).
- (10 points) Plot the separating hyperplanes on the figure. Note that the pairwise classifiers divide the feature space into 6 regions. Label these regions i, ii, ..., vi in your plot and fill in the table below according to what class each pairwise classifier would assign them. Finally use the voting rule for the multi-class classifier and thus estimate h_{OVO} . Shade the regions of the feature space according to what class label they would be assigned.

	i	ii	iii	iv	v	vi
h_{12}						
h_{23}						
h_{13}						
h_{OVO}						

- (d) (10 points) Use the h_{OVO} of the classifier that you just built to classify each of the following data points:

$$\mathbf{x}_1 = (10, 0)^\top, \mathbf{x}_2 = (-2, 6)^\top, \mathbf{x}_3 = (-2, -6)^\top, \mathbf{x}_4 = (9, 8)^\top, \mathbf{x}_5 = (9, 11)^\top$$

2.2 One versus all classifier (OVA)

A OVA classifier involves training a binary classifier for each class that discriminates between one class and all the remaining classes. Thus in total there will be C classifiers, each with their own set of parameters.

- (a) (5 points) Does the dataset (Figure 1) fulfill the assumptions of LDA, if we were to construct a binary classifier between each class and all other classes? Justify your answer!
- (b) (10 points) Suppose we were to construct h_{OVA} in a similar way as the h_{OVO} classifier presented above, i.e. we try to classify samples via majority vote. With the help of the example in Figure 1 and a rough sketch, explain in your own words, why this approach does not work.

Hint: Try separating the feature space into different regions, like in exercise 2.1 (c).

As we have seen in the previous exercise, we need more information than simply the class assigned by each of the C classifiers, in order to uniquely classify each possible input. Instead of only looking at which side of the hyperplane an input \mathbf{x}_i is on, we are now interested in how *confident* the classifier is in its decision. We train a binary classifier for each class $j \in \mathcal{Y}$ against all other classes

$$h_j(\mathbf{x}) = \mathbf{1}_{\{\mathbf{w}_j^\top \mathbf{x} + b_j - c_j \geq 0\}}.$$

We then obtain the probability of \mathbf{x}_i belonging to class j via

$$P(y = j | \mathbf{x} = \mathbf{x}_i) = \sigma(\mathbf{w}_j^\top \mathbf{x}_i + b_j - c_j), \quad (3)$$

where $\sigma(t) = \frac{1}{1+e^{-t}}$ is the sigmoid function. We can interpret this probability as the confidence of the binary classifier for each class. We thus define the multi-class OVA-LDA classifier as

$$h_{\text{OVA}}(x) = \operatorname{argmax}_{j \in \mathcal{Y}} [\sigma(\mathbf{w}_j^\top \mathbf{x} + b_j - c_j)] \quad (4)$$

where \mathbf{w}_j , b_j and c_j are parameters of classifier h_j .

- (c) (10 points) From the definition of the binary LDA classifier in the lecture notes, derive equation (3).
- (d) (10 points) Make a sketch of the dataset in Figure 2 and shade the regions of the feature space according to the class that equation (4) would assign them to.
- (e) (5 points) Sketch an example of a dataset in which the OVA method cannot provide an optimal classifier, but the OVO method can. Justify your answer with a brief explanation!

Hints:

- Compare your answers to exercises 2.1 (a) and 2.2 (a). Can you find a distribution of classes, where the OVA classifier violates the LDA assumptions to an extreme extent?
- You need not perform calculations, but instead use your intuition to make rough sketches.

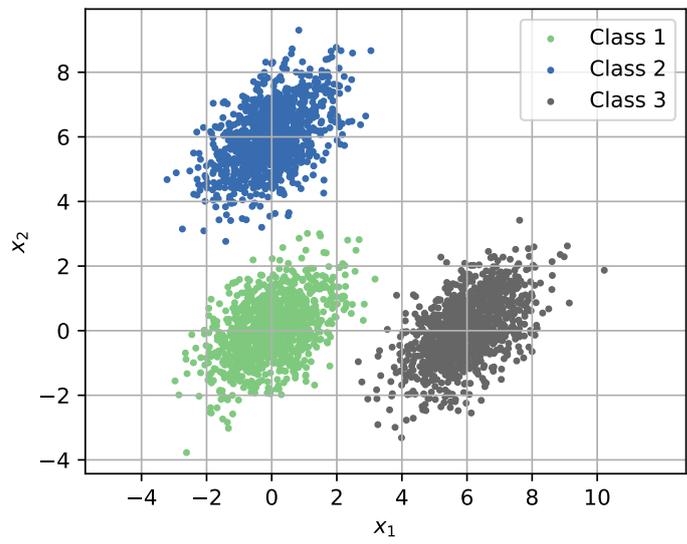


Figure 2: Example data