Exercise 3

Deadline: 21.11.2018, 15:00

Regulations

Provide your comments for last week's homework in the files `qda-lda-commented.ipynb` and `qda-lda-commented.html`. Hand-in your LDA derivation as a PDF file `LDA.pdf`, created with LaTeX or another tool of your liking, or scanned-in from a (readable!) hand-written solution on paper. The solution to task 3 goes into `qda-generation.ipynb` and `qda-generation.html`. Zip all files into a single archive with naming convention (sorted alphabetically by last names)

`lastname1-firstname1_lastname2-firstname2_exercise03.zip`

or (if you work in a team of three)

`lastname1-firstname1_lastname2-firstname2_lastname3-firstname3_exercise03.zip`

and upload it to Moodle before the given deadline.

Starting this exercise, we will give zero points if your zip-file does not conform to this naming convention.

1 Comment on your solution to exercise 2

Study the sample solution `ex02-solution.ipynb` provided on Moodle and use it to comment on your own solution to this exercise. Specifically, copy your original notebook `qda-lda.ipynb` to `qda-lda-commented.ipynb` and insert comments as markdown cells starting with

```
<span style="color:green;font-weight:bold">Comment</span>
```

so that we can clearly distinguish your comments from the other cell types. Insert comment cells at the appropriate places according to the following rules:

- If your code is incorrect, identify the bugs and make brief suggestions for possible fixes (don't include a full corrected solution).
- If your solution is slow, identify inefficient code sections (e.g. Python loops) and suggest possible improvements.
- If your code is correct, but differs from the sample solution, briefly explain why your solution as a valid alternative and where either solution is more elegant.
- If your code is essentially equal to the sample solution, explicitly say so.

Export the commented notebook to `qda-lda-commented.html` and hand in both files.

Note: If you fail to hand in `qda-lda-commented.html`, we will deduct 50% of the points from your solution to exercise 2.

2 LDA-Derivation from the Least Squares Error (24 points)

The goal of this exercise is to derive closed-form expressions for the optimal parameters $\hat{w}$ and $\hat{b}$ in Linear Discriminant Analysis, given some training set with two classes. Remember that the decision boundary in LDA is given by a $D-1$ dimensional hyperplane (where D is the dimension of the feature space) that we parametrize via

$$\hat{w}^T x + \hat{b} = 0. \tag{1}$$

$\hat{w}$ is the hyperplane's normal vector and $\hat{b}$ a scalar fixing its position in the D-dimensional space. **Note that w and x are column vectors in this exercise.** The decision rule for our two classes at query point x is then

$$\hat{y} = \text{sign}(\hat{w}^T x + \hat{b}) = \begin{cases} 1, & \text{if } \hat{w}^T x + \hat{b} > 0 \\ -1, & \text{if } \hat{w}^T x + \hat{b} < 0 \end{cases} \tag{2}$$

In the training phase we are given N datapoints $\{x_i\}_{i\in 1,\ldots,N}$ with $x_i \in \mathbb{R}^D$ and their respective labels $\{y_i\}_{i\in 1,\ldots,N}$ with $y_i \in \{-1, 1\}$. We assume that the training set is balanced, i.e.

$$N_1 = N_{-1} = \frac{N}{2} \tag{3}$$

with N_k denoting the number of instances in either class. The optimal parameters $\hat{w}$ and $\hat{b}$ are now the ones minimizing the least squares error criterion:

$$\hat{w}, \hat{b} = \text{argmin}_{w,b} \sum_{i=1}^{N} \left(w^T x_i + b - y_i\right)^2 . \tag{4}$$

You shall solve this problem in three steps: First (4 points), compute $\hat{b}$ from

$$\frac{\partial}{\partial b} \sum_{i=1}^{N} \left(w^T x_i + b - y_i\right)^2 = 0 . \tag{5}$$

Second (16 points), use this result to reshuffle

$$\frac{\partial}{\partial w} \sum_{i=1}^{N} \left(w^T x_i + \hat{b} - y_i\right)^2 = 0 . \tag{6}$$

into the intermediate equation

$$\left(S_W + \frac{1}{4} S_B\right) \hat{w} = (\mu_1 - \mu_{-1}) . \tag{7}$$

Here, μ_1 and μ_{-1} are the class means

$$\mu_{-1} = \frac{1}{N_{-1}} \sum_{i:\, y_i = -1} x_i \tag{8}$$

$$\mu_1 = \frac{1}{N_1} \sum_{i:\, y_i = 1} x_i \tag{9}$$

and S_B and S_W are the between-class and within-class covariance matrices

$$S_B = (\mu_1 - \mu_{-1})(\mu_1 - \mu_{-1})^T \tag{10}$$

$$S_W = \frac{1}{N} \sum_{i=1}^{N} (x_i - \mu_{y_i})(x_i - \mu_{y_i})^T . \tag{11}$$

The notation μ_{y_i} means

$$\mu_{y_i} = \begin{cases} \mu_{-1} & \text{if } y_i = -1 \\ \mu_1 & \text{if } y_i = 1 \end{cases} \tag{12}$$

Finally (4 points), transform equation (7) into

$$\hat{w} = \tau \; S_W^{-1} (\mu_1 - \mu_{-1}) \tag{13}$$

where τ is an arbitrary positive constant, expressing the fact that $\text{sign}(\tau(\hat{w}^T x + \hat{b}))$ is the same decision rule as $\text{sign}(\hat{w}^T x + \hat{b})$. During your calculations you may find the following relation for general vectors a, b and c useful:

$$a \cdot (b^T \cdot c) = (a \cdot c^T) \cdot b \tag{14}$$

3 Data Generation with QDA (8 points)

We learned in the lecture that QDA is a *generative* model, i.e. it can be used to create new data instances that look similar to the training set. We want to try this with the digits dataset again. Call the function:

```
mu, covmat, p = fit_qda(training_features, training_labels)
```

from exercise 2 with the full 81-dimensional feature vectors and determine the means and covariance matrices for two digit classes of your choice. Now pass means and covariances to

```
numpy.random.multivariate_normal()
```

to generate 8 new instances of either class and plot them as 9x9 images. Comment on the quality of the results and possible shortcomings of the method.