© N. Krasowski, U. Köthe
ullrich.koethe@iwr.uni-heidelberg.de

# Exercise 3

**Deadline: 22.11.2017, 10:00**

## Regulations

Please hand-in your solution for task 1 as a PDF file `LDA.pdf`, created with LaTeX or jupyter (or another tool of your liking), or scanned in from a hand-written solution on paper. The solution for task 2 shall be a jupyter notebook `naive-bayes.ipynb`, accompanied with exported PDF `naive-bayes.pdf`. Zip all files into a single archive with naming convention

`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 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, $\mu_1$ and $\mu_{-1}$ 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

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

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

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

$$\hat{w} = \tau \; S_W^{-1} (\mu_1 - \mu_{-1}) \tag{12}$$

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

$$a \cdot (b^T \cdot c) = (a \cdot c^T) \cdot b \tag{13}$$

## 2 Naive Bayes Classifier (16 points)

We will once again work with the digits dataset, this time using the naive Bayes classifier. If we assume that all classes are equally likely, i.e. the priors are $p(y = k) = 1/C$ with $C$ the number of classes, the decision rule is defined by

$$\hat{y} = \operatorname{argmax}_k \left( \prod_{j=1} p_j(x_j | y = k) \right) \tag{14}$$

where $p_j(x_j | y = k)$ are 1-dimensional histograms (or other suitable 1-dimensional probability estimators) for each feature $j$ and class $k$. Since the product of many small probabilities is a tiny number prone to numerical inaccuracy, it is better to rewrite this as a sum of logarithms:

$$\hat{y} = \operatorname{argmax}_k \left( \sum_{j=1} \log p_j(x_j | y = k) \right) \tag{15}$$

Implement training of the naive Bayes classifier as a function

`histograms = fit_naive_bayes(features, labels, bincount)`

where `histograms` is the $C \times D \times B$ array of histograms ($D$ is the number of feature dimensions, $B$

the number of bins). If `bincount=0`, the number of bins shall be determined automatically according to the Freedman-Diaconis rule. Likewise, implement prediction as a function

```
predicted_labels = predict_naive_bayes(test_features, histograms)
```

Filter the trainig set to use only digits "3" and "9", train the classifier with

- the 2-dimensional feature space you constructed in exercise 2 (with `bincount=0`), and
- the full 64-dimensional feature space (one histogram per pixel with `bincount=8`)

and determine the two confusion matrices on the test set. Visualize the decision boundary for the 2-D case, overlayed with the scatterplot of the test data.