Elements of Machine Learning, WS 2023/2024
Jilles Vreeken and Krikamol Muandet
Exercise Sheet #2: *Classification*

Deadline: Thursday, November 30, 2023, 15:00

Before solving the exercises, read the instructions on the course website.

- For each theoretical problem, submit a single `pdf` file that contains your answer to the respective problem. This file may be a scan of your (legible) handwriting.
- For each practical problem, submit a single `zip` file that contains
  - the completed jupyter notebook (`.ipynb`) file,
  - any necessary files required to reproduce your results, and
  - a `pdf` report generated from the jupyter notebook that shows all your results.
- For the bonus question, submit a single `zip file` that contains
  - a `pdf` file that includes your answers to the theoretical part,
  - the completed jupyter notebook (`.ipynb`) file for the practical component,
  - any necessary files required to reproduce your results, and
  - a `pdf` report generated from the jupyter notebook that shows your results.
- **Every team member** has to submit a signed Code of Conduct.
- **IMPORTANT** You must make the team on CMS *before* you upload the solutions. If you upload the solutions first and create the team after it, the solution will not show for the new team member!

Problem 1 (T, 8 Points). **Logistic regression.** When answering the following questions be precise ,e.g. use equations, and keep your answers short.

1. [2*pts*] What are two problems of linear regression for classification?
2. [1*pts*] What is the mathematical relationship between the input variables and predicted probabilities? What does logistic regression model?
3. [1*pts*] What are odds and how are they computed? Explain the relationship between odds and logistic regression?
4. [2*pts*] In the case of multivariate logistic regression, how do the odds change for an input $x \in \mathbb{R}^d$, if x is perturbed as $\tilde{x} = x + \epsilon e_j$, for $\epsilon \neq 0$ What can you say about the change depending on ϵ?
5. [2*pts*] Let X be a scalar random variable. Prove that

$$p(X) = \frac{e^{\beta_0+\beta_1 X}}{1+e^{\beta_0+\beta_1 X}} \iff \log\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + \beta_1 X \ .$$

   What is the relationship between the logistic and the logit function? Why is this information about the relationship important? Explain.

Problem 2 (T, 12 Points). **Linear Discriminate Analysis.**

1. [2*pts*] In the derivation LDA we assume that input feature $x \in \mathbb{R}$ is continous. Here, we consider $x \in \mathbb{N}_0$ to be a natural number and x follows a Poisson distribution. So, $p_k(x)$ is given by

$$p_k(x) = \Pr(Y = k \mid X = x) = \frac{\pi_k f_k(x)}{\sum_{\ell=1}^{K} \pi_\ell f_\ell(x)} \ , \text{ where } f_k(x) = \frac{\lambda_k^x e^{-\lambda_k}}{x!} \ .$$

   Per class k, there is seperate distribution parameter $\lambda_k > 0$. Derive the decision boundary for two classes k and l with respect to x.

2. [4*pts*] In the lecture we dealt with finitely many classes $k \in [K]$. In the following assume there are *infinitely many classes*[1] $k \in \mathbb{N}_0$ (here classes start 0) and k is Poisson-distributed $\pi(k) = \frac{\lambda^k e^{-\lambda}}{k!}$. Moreover, assume $x \in \mathbb{R}$ and $f_k(x) \sim \mathcal{N}(\mu_k, \sigma^2)$. Derive the decision boundary for two classes k and l with respect to x.
   Hint: Perform similar steps as in (1.), except that $\pi(k)$ follows a poisson distribution .

3. [3*pts*] Next, we consider the multivarite setting $x = (x_1, ..., x_m)^T$ with K classes i.e. $p(k) = \pi_k$. Furthermore, we assume that all x_i are mutually independent, that is $f(x) = \prod_{i=1}^{m} f_i(x_i)$, even when conditioned on a certain class. We assume that the variance is the same across classes, while the means differ per class. Thus, per class each feature is distributed as $f_{ki}(x_i) \sim \mathcal{N}(\mu_{ki}, \sigma_i^2)$. Derive the density $f_k(x)$. State (not derive) the discrimant δ_x.

4. [2*pts*] What are the assumptions of LDA regarding the data distribution and decision boundary. How do they compare to the assumptions of the other models presented in the lecture ($k-$NN, Logistic regression, QDA)?

5. [1*pts*] For the multivariate setting, give a heuristic depending on covariance matrix Σ, when it is better to use LDA or QDA. Why can it be preferable to use LDA instead of QDA in that case?

Problem 3 (P, 15 Points). **Speech Recognition.** We will now consider LDA and QDA for a real-world speech recognition task. The data we consider contains digitized pronunciation of five phonemes: `sh` as in "she", `dcl` as in "dark", `iy` as the vowel in "she", `aa` as the vowel in "dark", and `ao` as the first vowel in "water". These phonemes correspond to responses/classes (column name `g`). The dataset contains 256 predictors (log-periodograms, which is a common way of representing voice recordings in speech recognition).
Please rename the file to include the matriculation numbers of all team members (e.g. `7010000_2567890_A1.ipynb`)
Use `Practical_Problem_1.ipynb` found in the `a1_programming` file from the course website.

1. [2*pts*] Load the phoneme data set `phoneme.csv` and split the dataset into a training and test set according to the `speaker` column. Then exclude the `row.names`, `speaker` and response column `g` from the features.
   Useful functions: `sklearn.model_selection.StratifiedShuffleSplit`.

2. [2*pts*] Fit an LDA model to classify the response based on the predictors; then compute and report train and test error.
   Useful functions: `sklearn.discriminant_analysis.LinearDiscriminantAnalysis`.

3. [4*pts*] For every pair of phenomes select the corresponding data points. Fit an LDA model on all data sets and repeat the steps done in (2). Explain your findings.

4. [4*pts*] Repeat steps (2) and (4) using QDA and report your findings. What model do you prefer and why?

[1] This setting corresponds to CW Ex 1.3 e.g. how many cars cross an intersection.

5. [3*pts*] Generate confusion matrices for the LDA and QDA model for the combination of phenomes, which proved to be the hardest to classify. Which differences can you observe between the models?

Problem 4 (Bonus). **Logistic Regression.**
In the lecture we saw multivariate logistic regression (MLR) for the two class setting. For a weight vector $w = (w_1, ..., w_m)^T \in \mathbb{R}^m$ and bias $b \in \mathbb{R}$, the probability of a sample x has class $y = 1$ is given by:

$$p(y = 1 \mid x) = \frac{\exp(w^T x + b)}{1 + \exp(w^T x + b)} = \sigma(w^T x + b) = \sigma(z) \ ,$$

where $z = w^T x + b$. In practice, several tasks involve classifying more than two classes. In this exercise, we extend the multivariate logistic regression to the multiclass setting (MMLR). Instead of having one weight vector, MMLR uses K weight vectors i.e. one per class. The probability of a sample x_n belonging to class $y = k$ is modelled as

$$p(y = k \mid x_n) = \frac{\exp(w_k^T x_n + b_k)}{\sum_{j=1}^{K} \exp(w_j^T x_n + b_j)} =: \Lambda_\theta(k \mid z_n)$$

where $z_{nj} = w_j^T x_n + b_j$, for $j = 1...K$ and $\theta = \{w_i\}_{i=1}^m \cup \{b_i\}_{i=1}^m$. In the following, we omit the subscript θ if it is not required.

1. Show that for $K = 2$, MLR and MMLLR are equivalent by choosing appropriate parameters.
2. For an i.i.d. set of observations $(X, Y) = \{(x_n, y_n)\}_{n=1}^N$, where $x_i \in \mathbb{R}^m$ and $y_i \in \{1..K\}$, show that the negative log likelihood function for MLLR can be written as:

$$-\log p(Y \mid X) = -\sum_{n=1}^{N} \sum_{k=1}^{K} y_{nk}^o \log \hat{y}_{nk} \ . \tag{4.1}$$

   For ease of notation, let $\hat{y}_{nk} = \Lambda(k \mid z_n)$. Furthermore, $y_n^o \in \{0, 1\}^K$, denotes the one-hot encoding of y_n i.e. $y_{nk}^o = 1$ if $y_n = k$ else 0.
3. Show that the partial derivative of $\Lambda(k \mid z_n)$ wrt. z_{nj} can be written as

$$\frac{\partial \Lambda(k \mid z_n)}{\partial z_{nj}} = \Lambda(k \mid z_n)(\mathbb{1}_{kj} - \Lambda(j \mid z_n)) \ ,$$

   where $\mathbb{1}$ is the identity matrix.
4. Next, the goal is to find the set of parameters $\theta = \{w_i\}_{i=1}^m \cup \{b_i\}_{i=1}^m$ that minimize the negative log-likelihood in equation 4.1.

$$\arg\min_\theta \mathcal{L}(Y, \hat{Y}_\theta) = -\sum_{n=1}^{N} \sum_{k=1}^{K} y_{nk}^o \log \hat{y}_{nk}$$

   Note that $\hat{Y}_\theta$ depends on θ. In the lecture, we used the Newton-Rhapson method to optimize the weights, which involves computing the inverse of hessian with respect to the parameters. Depending on the dimensionlity of the data, this can be prohibitively expensive. Here, we resort to a simpler optimzation scheme, namely *gradient descent*. The update rule for a weight vector $w_j \in \theta$ and bias $b_j \in \theta$ is given by

$$w_j^{new} = w_j^{old} - \alpha \nabla_{w_j^{old}} \mathcal{L}(Y, \hat{Y}_{w_j^{old}})$$
$$b_j^{new} = b_j^{old} - \alpha \partial_{b_j^{old}} \mathcal{L}(Y, \hat{Y}_{b_j^{old}})$$

   This involves computing, the gradient ∇^2 and partial derivative with respect to parameters. The hyper-parameter α is called the learning rate and specifies the step size in each iteration.

[2] `https://en.wikipedia.org/wiki/Partial_derivative`

(a) Derive the gradient descent update in closed form.

(b) Implement the gradient update in the accompanying notebook. Do **not** use any for loops for the gradient update, instead use matrix and vector operations.

(c) Train your implentation of MMLR on the handwritten digit dataset. Choose the learning rate and number of iterations, such that the classifier achieves a test accuracy of $> 90\%$.