3 Classification

Exercise 3.1

Which of the following is an example of *qualitative variable*?

1. Height
2. Age
3. Speed
4. Color

Provide a method to convert the qualitative ones into quantitative one, without introducing further structure over the data.

Exercise 3.2

Consider the following code lines in `Python`:

```
url = "https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.
    data"
dataset = pd.read_csv(url, names=names)

x = dataset[['sepal-length']].values
t = dataset['class'].values == 'Iris-setosa'
phi = zscore(np.hstack((x, x ** 2)))

lin_model = sm.OLS(t, x).fit()
qua_model = sm.OLS(t, phi).fit()

if qua_model.rsquared_adj > lin_model.rsquared_adj:
  y = lin_model.predict(x)
else:
  y = qua_model.predict(phi)
```

1. Describe the process and purpose of what is implemented in this snippet.
2. Tell if the method is sound or if it is necessary to modify the procedure to follow the classic ML guideline regarding this method.

Exercise 3.3

Suppose we collect data for a group of workers with variables hours spent working x_1, number of completed projects x_2 and receive a bonus t. We fit a logistic regression and produce estimated coefficients: $w_0 = -6$, $w_1 = 0.05$ and $w_2 = 1$.

Estimate the probability that a worker who worked for 40h and completed 3.5 projects gets an bonus.

How many hours would that worker need to spend working to have a 50% chance of getting an bonus?

Do you think that values of z in $\sigma(z)$ lower than -6 make sense in this problem? Why?

Exercise 3.4

Suppose you have trained a logistic regression classifier l on a dataset $Z = \{(\mathbf{x}_n, t_n)\}_n$ and the output corresponding to observation $\mathbf{x}_n$ is $\hat{y}_n$. Currently, you predict class 1 if $\hat{y}_n > \tau$, and predict 0 if $\hat{y}_n < \tau$, with $\tau = 0.5$. Suppose you increase the threshold to $\tau = 0.8$ getting a new classifier l_{new}. Which of the following are true? Check all that apply and provide a motivation.

1. The number of samples x_n from Z classified as positive instance will decrease if we use l_{new} instead of l.
2. The number of samples x_n from a test dataset Z_{test} classified as positive instance will decrease if we use l_{new} instead of l.
3. The classifier l_{new} is likely to have a higher accuracy.
4. The classification error over Z might decrease by using l_{new} instead of l.

* Exercise 3.5

Derive for logistic regression, the gradient descent update for a batch of K samples.

Do we have assurance about converge to the optimum?

Exercise 3.6

Tell if the following statement about the perceptron algorithm for classification are true or false.

1. Shuffling the initial data influences the perceptron optimization procedure;
2. We are guaranteed that, during the learning phase, the perceptron loss function

is decreasing over time;

3. There exists a unique solution to the minimization of the perceptron loss;
4. The choice of a proper learning rate α might speed up the learning process;
5. The solution of the Logistic regression and the one of the perceptron always coincide.

Motivate your answer.

Exercise 3.7

Consider a classification problem having more than two classes. Propose a method to deal with multiple classes in each one of the following methods:

1. K-Nearest Neighbors;
2. Näive Bayes;
3. Linear regression;
4. Logistic regression;
5. Perceptron.

Exercise 3.8

Tell if the following statements are true or false and motivate your answers.

1. The relationship between the input $\mathbf{x}$ and the estimated output class $y(\mathbf{x})$ induced by a generalized linear model used for classification is linear;
2. The solution of a classification problem using *discriminant function* provides a probability distribution of a generic input $\mathbf{x}$ to belong to a class C_k;
3. Both the Logistic regression and the perceptron use the same loss function to learn the boundary between the classes;
4. Both the Logistic regression and the perceptron use the same updating rule to learn the boundary between the classes.

Exercise 3.9

Given the following dataset:

$$\mathbf{x}_1 = (2,3,4)^\top,\ y_1 = 1 \qquad \mathbf{x}_2 = (0,1,2)^\top,\ y_2 = 0$$
$$\mathbf{x}_3 = (1,2,5)^\top,\ y_2 = 1 \qquad \mathbf{x}_4 = (1,4,3)^\top,\ y_4 = 0$$
$$\mathbf{x}_5 = (0,3,1)^\top,\ y_5 = 0 \qquad \mathbf{x}_6 = (1,2,2)^\top,\ y_6 = 0$$
$$\mathbf{x}_7 = (3,1,4)^\top,\ y_7 = 1 \qquad \mathbf{x}_8 = (4,2,5)^\top,\ y_8 = 1$$
$$\mathbf{x}_9 = (1,3,3)^\top,\ y_9 = 0 \qquad \mathbf{x}_{10} = (1,2,4)^\top,\ y_{10} = 1$$

- Classify the point $\mathbf{x}_{11} = (0,1,2)^\top$ according to a KNN classifier trained on the given dataset with $K = 3$;
- What happens if we use $K = 10$ instead? Do you think it is a good idea to choose such a parameter (hint: two pros or two cons);
- Suggest a technique to set the parameter K.

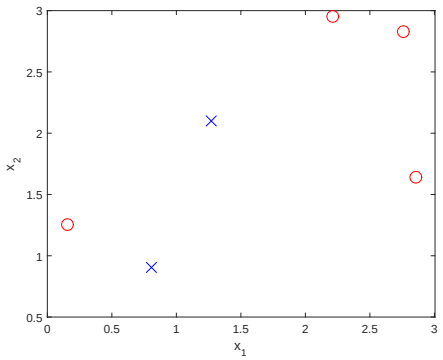
Exercise 3.10

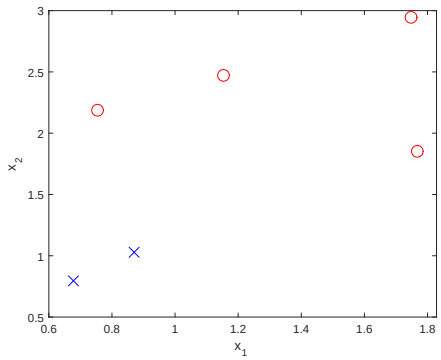
Consider a binary classifier trained on a dataset made of $N = 100$ samples.

1. Suppose that the Precision = 0.25 and the F1 = 0.4, compute the Recall.
2. Knowing, in addition, that the Accuracy = 0.85, compute the full confusion matrix.
3. In which circumstances the Accuracy is not a reliable index to assess the quality of the trained model?

Exercise 3.11

Consider the following datasets:





and consider the online stochastic gradient descend algorithm to train a perceptron.

Does the learning procedure terminates? If so, how many steps we require to reach convergence? Provide motivations for your answers.

What about the Logistic regression?

Exercise 3.12

Starting from the formula of the softmax classifier for k classes:

$$y_k(\mathbf{x}) = \frac{\exp(\mathbf{w}_k^T \mathbf{x})}{\sum_{j=1}^{K} \exp(\mathbf{w}_j^T \mathbf{x})}$$

derive the formula for the sigmoid logistic regression parameter $\mathbf{w}$ for the two-class problem.

Assume that the estimated parameter is $\mathbf{w} = (4, 2, 1)^\top$ and the input vector is of the form $\mathbf{x} = (x_1, x_2, 1)^\top$. Draw the boundary of the logistic regression in the input space and in the parameter space.

Exercise 3.13

Consider one at a time the following characteristics for an ML problem:

1. Large dataset (big data scenario);
2. Embedded system;
3. Prior information on data distribution;
4. Learning in a Real-time scenario;
5. Reduced computational capabilities.

Provide motivations for the use of either a parametric or non-parametric method in the above situations.

Exercise 3.14

Consider the following dataset to implement a spam filter function:

"pills"	"fee"	"kittens"	Url Presence	"PoliMi"	spam
0	1	0	0	1	0
0	0	1	1	0	0
0	0	1	0	0	0
0	0	1	0	1	0
0	0	0	0	0	0
1	1	0	0	1	1
0	1	0	1	0	1
1	0	0	1	0	1

where we enumerate the presence of specific word or of an URL in 8 different e-mails and the corresponding inclusion in the spam or non-spam class.

1. Estimate a Naïve Bayes classifier, choosing the proper distributions for the classes priors and the feature posteriors.

2. Predict the probability of the following samples to belong to the spam and no-spam classes.

"pills"	"fee"	"kittens"	Url Presence	"PoliMi"
1	1	0	1	0
0	1	1	0	1

Exercise 3.15

Consider the following snippet of code:

```
1  X = zscore(dataset[['sepal-length', 'sepal-width']].values)
2  t = dataset['class'].values == 'Iris-setosa'
3  X, t = shuffle(X, t, random_state=0)
4
5  w = np.ones(3)
6  for i, (x_i, t_i) in enumerate(zip(X, t)):
7    ext_x = np.concatenate([np.ones(1), x_i.flatten()])
8    if np.sign(w.dot(ext_x)) != t_i:
9        w = w + ext_x * t_i
```

1. Describe the procedure presented above. What is the purpose of such a procedure? Which problem it solves?

2. Tell if the procedure above is correct and, in the case it is not, propose a modification to fix it.

3. Do you think that Line 3 is fundamental for this procedure? Can you describe an example where it can be removed and motivate why it is not useful in such a case?

Exercise 3.16

Consider a binary perceptron classifier defined by parameters $w = [2, 1, 1]^\top$ with features vector $\phi([a, b]) = [a, b, ab]^\top$). Answer to the following questions related to the perceptron algorithm. Provide full calculations and clear motivations.

1. Given a new data point $(x, t) = ([a, b], t)$, explain the procedure you would follow to decide whether the classifier should be retrained.
2. Consider the data point $(x_1, t_1) = ([1, 2], +1)$. Update the classifier with the perceptron algorithm ($\alpha = 1$).
3. Consider the data point $(x_2, t_2) = ([-1, -2], +1)$. Update the classifier with the perceptron algorithm ($\alpha = 1$).
4. After the previous updates to the classifier, can we say the retrain procedure is completed?

Answers

Answer of exercise 3.1

1. Quantitative variable: we are able to order heights and they usually are in a bounded continuous set (e.g., in $[50, 230]$ cm);
2. Quantitative variable: they are ordered values, all of them being natural numbers;
3. Quantitative variable: this variable takes values in the real numbers;
4. Qualitative variable: since we do not have an ordering over the colors (in general cases), we need to convert the set of the available colors into binary variables.

Given the set $\mathcal{C} = \{c_1, \dots, c_p\}$ of all possible colors, a solution that does not enforce further structure is *one-hot encoding*. We create a new binary variable $z_i \in \{0, 1\}$ for each color c_i, that gets value 1 when the color is actually c_i. This way, a color c_i is fully determined by a binary vector $\mathbf{z}_i = (z_1, \dots, z_p)^\top$, where $z_i = 1$ and all other $z_j = 0$ for $j \neq i$ (i.e., $\mathbf{z}_i$ is the i-th vector of the canonical basis of $\mathbb{R}^p$). Clearly, this procedure requires creating p new variables. It is worth noting that solution does not introduce further structure in the data since any two vectors $\mathbf{z}_i \neq \mathbf{z}_j$ are equally distant for any reasonable choice of metric (e.g., Euclidean).

It is worth noting that associating to each color c_i the quantitative variable i would introduce further structure in the data and should be avoided in general, although requiring just one variable instead of p. Indeed, this solution enforces an ordering among the colors. Furthermore, color c_i would result closer to color c_{i+1} than to color c_{i+2} w.r.t. the Euclidean distance.

Answer of exercise 3.2

1. This snippet performs classification on the Iris dataset. The script evaluates two different models, one linear in the original input and one with the addition of a quadratic feature, by looking at the adjusted R^2 of the fitted model.
2. There are three issues about the model we trained:
   a) A model with larger adjusted R^2 is better, therefore the `if` conditions should be the other way round.
   b) It is not a good idea to solve a classification problem with a regression one. This solution is not robust to outliers.
   c) The `zscore` normalization is performed for the `qua_model` only.

Answer of exercise 3.3

The logistic model provides as output the probability of getting a bonus, thus:

$$P(t = 1|\mathbf{x}) = \sigma(w_0 + w_1 x_1 + w_2 x_2)$$

where $x_1 = 40$ and $x_2 = 3.5$

$$P(t = 1|\mathbf{x}) = \sigma(-6 + 0.05 \cdot 40 + 1 \cdot 3.5) = \sigma(-0.5) = 0.3775$$

To have a $\alpha\%$ chance of having a bonus we need to invert the sigmoidal function, while in this case we know that we have 50% chance when the argument of the sigmoid is equal to zero, thus:

$$\begin{aligned} w_0 + w_1 \hat{x} + w_2 x_2 &= 0 \\ -6 + 0.05 \cdot \hat{x} + 3.5 &= 0 \\ \hat{x} &= \frac{2.5}{0.05} = 50 \end{aligned}$$

Since all the considered variables are positive definite, it makes only sense to consider values greater than -6 as predictions.

Answer of exercise 3.4

1. TRUE: we are shrinking the positive region, thus it might happen that some points which were classified as positive will be classified as negative with l_{new}.
2. TRUE: same reason as in 1.
3. TRUE/FALSE: the amount of true positive decreases and the amount of true negative increases, thus it might happen that the accuracy increases or decreases.
4. FALSE: if we properly trained the logistic regression the classification error should increase when we use a sub-optimal decision boundary.

Answer of exercise 3.6

1. TRUE: the learning procedure is influenced by both the initial parameter we consider and the order we present the data to it.
2. FALSE: we are guaranteed that the error (loss) on the currently considered datum does not increase.
3. FALSE: if the data are linearly separable, there is an infinite number of linear boundaries able to provide the same loss performance, which are all equivalent solutions for the perceptron.

4. FALSE: the parameter vector norm does not influence the result of the discrimination, thus the use of a generic parameter $\alpha > 0$ would work.

5. FALSE: even if the two methods use the same optimization algorithm, they have different loss functions and, therefore, their solution might be different.

Answer of exercise 3.7

1. KNN is naturally able to deal with multi-class classification problems by using majority voting to decide the class. However, we need to carefully choose the way we are breaking ties since this might be crucial in the case of many classes.

2. Näive Bayes is able to deal with multi-class classification problems by using a categorical distribution for the class distribution prior $p(C_k)$ and estimating the posterior $P(\mathbf{x}|C_k)$ for each class.

3. Linear regression is a regression methods and, consequently, it is not suited for classification problems.

4. The standard definition of a Logistic regression classifier deals with binary classification problems. It can be extended to multi-class classification, using the softmax transformation.

5. The perceptron deals with binary classification problems. It can be employed for K multi-class classification training K one-versus-the-rest classifiers.

Answer of exercise 3.8

1. FALSE. A generalized linear model applies a non-linear transformation g to the linear model $\mathbf{w}^\top \mathbf{x}$. Thus, the overall relationship is non-linear, unless g is linear.

2. FALSE. A discriminant function f directly provides the predicted class $f(\mathbf{x}) = C_k$, not the predicted probability.

3. FALSE. The logistic regression uses the log likelihood, while the preceptron the distance from the decision boundary of the missclassified points.

4. TRUE. They both use online gradient descent.

Answer of exercise 3.9

1. The distance of $\mathbf{x}_{11}$ from the training points are:

$$d_1 = \sqrt{2^2 + 2^2 + 2^2} = \sqrt{12} \qquad d_2 = \sqrt{0^2 + 0^2 + 0^2} = 0$$
$$d_3 = \sqrt{1^2 + 1^2 + 3^2} = \sqrt{11} \qquad d_4 = \sqrt{1^2 + 3^2 + 1^2} = \sqrt{11}$$
$$d_5 = \sqrt{0^2 + 2^2 + 1^2} = \sqrt{5} \qquad d_6 = \sqrt{1^2 + 1^2 + 0^2} = \sqrt{2}$$
$$d_7 = \sqrt{3^2 + 0^2 + 2^2} = \sqrt{13} \qquad d_8 = \sqrt{4^2 + 1^2 + 3^2} = \sqrt{26}$$
$$d_9 = \sqrt{1^2 + 2^2 + 1^2} = \sqrt{6} \qquad d_10 = \sqrt{1^2 + 1^2 + 2^2} = \sqrt{6}$$

   The 3 closest points to $\mathbf{x}_{11}$ according to the Euclidean distance are $\mathbf{x}_2$, $\mathbf{x}_6$, and $\mathbf{5}$. Therefore, the class predicted by the KNN is the negative class (0).

2. If we use $K = 10$, all points belong to the set of the 10 closest points to $\mathbf{x}_{11}$. Since among the 10 points there are 5 positive class points and 5 negative class points, the KNN is unable to provide a prediction, unless a the breaking rule is provided.

3. K can be regarded as a regularization parameter of the model. Thus, a sound way to select it is cross-validation.

Answer of exercise 3.10

1. First of all, we compute the Recall from the F1 definition:

$$F1 = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \implies \text{Recall} = \frac{F1 \cdot \text{Precision}}{2 \cdot \text{Precision} - F1} = \frac{0.4 \cdot 0.25}{2 \cdot 0.25 - 0.4} = \frac{0.1}{0.1} = 1.$$

2. To get the confusion matrix, we simply apply the definition of Accuracy, Precision, and Recall, keeping in mind that the total number of samples is $N = 100$:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} = \frac{TP + TN}{100} = 0.85 \implies TP + TN = 85$$
$$\text{Recall} = \frac{TP}{TP + FN} = 1 \implies FN = 0$$
$$\text{Precision} = \frac{TP}{TP + FP} = 0.25 \implies (1 - 0.25)TP = 0.25FP \implies 3TP = FP$$

   Now, using $TP+TN+FP+FN = 100$, recalling that $TP+TN = 85$ and $FN = 0$, we get $FP = 15$. Using $3TP = FP$, we get $TP = 5$. Finally, using $TP+TN = 85$, we get $TN = 80$. Thus, the confusion matrix is given by:

	Actual Class: 1	Actual Class: 0
Predicted Class: 1	5	15
Predicted Class: 0	0	80

3. Accuracy is not a reliable index of the quality of the trained model mainly in two scenarios: (i) when the dataset is unbalanced; (ii) when the importance of wrongly predicting positive-class samples is different from wrongly predicting negative-class samples.

Answer of exercise 3.11

The perceptron learning algorithm is guaranteed to converge in the case it exists a linear separation hyperplane. In this case, we are able to reduce the classification error to zero, otherwise the optimization procedure does not stop. We do not have any assurance on the convergence rate, since it depends on the starting point for the parameter, and on the ordering of the points we consider for training. Nevertheless the convergence occurs in a *finite* number of steps.

In the first case (left), we are sure it does not converge, while in the second case (right) the online stochastic gradient descend will eventually converge in a finite number of steps.

Since the loss function for the logistic regression is convex, the online learning procedure converges to the global optimum (asymptotically), no matter what dataset is given in input.

Answer of exercise 3.12

Since we are considering only two classes we have that summation is only over two parameter vectors $\mathbf{w}_1$ and $\mathbf{w}_2$. if we consider class C_1 we may write:

$$\begin{aligned} y_1(x) &= \frac{\exp(\mathbf{w}_1^T\mathbf{x})}{\exp(\mathbf{w}_1^T\mathbf{x}) + \exp(\mathbf{w}_2^T\mathbf{x})} \\ &= \frac{\frac{\exp(\mathbf{w}_1^T\mathbf{x})}{\exp(\mathbf{w}_1^T\mathbf{x})}}{\frac{\exp(\mathbf{w}_1^T\mathbf{x})+\exp(\mathbf{w}_2^T\mathbf{x})}{\exp(\mathbf{w}_1^T\mathbf{x})}} \\ &= \frac{1}{1+\exp[(\mathbf{w}_2-\mathbf{w}_1)^T\mathbf{x}]} \end{aligned}$$

Similarly for class C_2 we have the same formula with $(\mathbf{w}_1 - \mathbf{w}_2)^T$. Since we are considering a probability distribution it is not necessary to consider both $y_1(\mathbf{x})$ and $y_2(\mathbf{x})$. Indeed, $y_2(\mathbf{x}) = 1 - y_1(\mathbf{x})$, which is why we just need to store a single parameter vector $\mathbf{w} = \mathbf{w}_2 - \mathbf{w}_1$ and the formula for the two class classifier has a single parameter vector:

$$y_1(x) = \frac{1}{1+\exp(\mathbf{w}^T\mathbf{x})}.$$

The boundary in the parameter space is the point $[4\ 2\ 1]$ in a 3D space, while in the input space it is the line $\mathbf{w}^T\mathbf{x} = 0 \rightarrow x_2 = -2x_1 - \frac{1}{2}$.

Answer of exercise 3.13

1. PARAMETRIC: in the case we have a large dataset it is better to have a model which is able to capture the characteristics of the problem than by basing on the entire dataset to provide a prediction. Indeed, a non-parametric method would require storing the whole dataset and perform queries on it to perform predictions.

2. PARAMETRIC/NON-PARAMETRIC: some of the algorithm we considered in machine learning requires computationally expensive training phases. If the entire system has to work on an embedded system, we should either perform the training phase on a different device or use non parametric methods, which does not require a learning phase. On the other hand, non-parametric methods require storing the whole dataset which might be infeasible on an embedded systems due to possible memory constraints.

3. PARAMETRIC: an easy way for introducing a-priori information on the dataset we have in a learning methods is to include them in the model (e.g., with a Bayesian approach). Since non-parametric only are based on data, it is not trivial to include prior knowledge in them.

4. PARAMETRIC/NON-PARAMETRIC: if we want a fast way of performing tasks a non-parametric method is probably a good idea, since it does not require to have a training phase. On the contrary, if we are able to provide an online method for training a parametric method, we could also consider parametric methodologies. Recall, that a non-parametric method, although it does not require training, the prediction requires querying the dataset.

5. PARAMETRIC/NON-PARAMETRIC: if we are able to perform the training phase on a different device, a parametric model would be a good choice in the case we have reduced computational capabilities. If we do not have this possibility a non–parametric method would be a good choice since it does not require any training, but might be memory demanding.

Answer of exercise 3.14

1. The proper models in this case uses Bernoulli variables both as prior distributions and for the posteriors. More specifically we estimate the probabilities with the

MLE, which is the common empirical expected value:

$$
\begin{aligned}
P(C_1) &= \frac{5}{8} & P(C_2) &= \frac{3}{8} \\
P(x_1 = 0 \mid C_1) &= \frac{5}{5} & P(x_1 = 1 \mid C_1) &= \frac{0}{5} \\
P(x_2 = 0 \mid C_1) &= \frac{4}{5} & P(x_2 = 1 \mid C_1) &= \frac{1}{5} \\
P(x_3 = 0 \mid C_1) &= \frac{2}{5} & P(x_3 = 1 \mid C_1) &= \frac{3}{5} \\
P(x_4 = 0 \mid C_1) &= \frac{4}{5} & P(x_4 = 1 \mid C_1) &= \frac{1}{5} \\
P(x_5 = 0 \mid C_1) &= \frac{3}{5} & P(x_5 = 1 \mid C_1) &= \frac{2}{5} \\
P(x_1 = 0 \mid C_2) &= \frac{1}{3} & P(x_1 = 1 \mid C_2) &= \frac{2}{3} \\
P(x_2 = 0 \mid C_2) &= \frac{1}{3} & P(x_2 = 1 \mid C_2) &= \frac{2}{3} \\
P(x_3 = 0 \mid C_2) &= \frac{3}{3} & P(x_3 = 1 \mid C_2) &= \frac{0}{3} \\
P(x_4 = 0 \mid C_2) &= \frac{1}{3} & P(x_4 = 1 \mid C_2) &= \frac{2}{3} \\
P(x_5 = 0 \mid C_2) &= \frac{2}{3} & P(x_5 = 1 \mid C_2) &= \frac{1}{3}
\end{aligned}
$$

2. The probability is the product of the priors and the posteriors of each feature:

   - First sample: $P(C_1|x) \propto P(C_1)P(x_1 = 1 \mid C_1)P(x_2 = 1 \mid C_1)P(x_3 = 0 \mid C_1)P(x_4 = 1 \mid C_1)P(x_5 = 0 \mid C_1) = \frac{5}{8} \cdot 0 \cdot \dots = 0$, therefore it belongs to C_2 for the trained NB model.

   - Second sample: $P(C_2|x) \propto P(C_2)P(x_1 = 0 \mid C_2)P(x_2 = 1 \mid C_2)P(x_3 = 1 \mid C_2)P(x_4 = 0 \mid C_2)P(x_5 = 1 \mid C_2) = \frac{3}{8} \cdot \frac{1}{3} \cdot \frac{2}{3} \cdot 0 \ldots = 0$, therefore it belongs to C_1 for the trained NB model.

Answer of exercise 3.15

1. The snippet is applying the perceptron online learning algorithm to the iris dataset. At first, it standardize the data and shuffle the samples. Then it initialize the weight vector $w = [111]$ and apply the gradient descent update over each of the samples in the training set. It solves a binary classification problem.

2. The procedure should be repeated multiple times over the dataset. In this case, a single iteration over the dataset does not guarantee that the algorithm converges

to a meaningful solution, even if the dataset is linearly separable. We should add a for loop to perform the procedure multiple times over the dataset, until the solution w does not change for an entire epoch (an iteration over the entire dataset) or a maximum number of iterations is reached. In addition, the encoding of the target variable (line 2) is not correct. The target variable should be encoded as -1 and 1 instead of 0 and 1.

3. Here it is important to shuffle the data since it might occur that the samples are ordered (e.g., the ones of the positive class are all in the first positions of the dataset), which might slow down the optimization procedure. The same holds if we want to apply a train/validation approach. Instead, if we are using a closed form solution, like the one for linear regression, the shuffling procedure is not mandatory.

Answer of exercise 3.16

1. First, we should check if the new data point is correctly classified, i.e., whether

$$sign(w^\top \phi(x)) = sign([2,1,1] \cdot [a,b,ab]^\top) = t$$

If the answer is positive we do not need to retrain, otherwise we should retrain the model, i.e., use the stochastic gradient descent method until convergence.

2.

$$sign(w^\top \phi(x_1)) = sign([2,1,1] \cdot [1,2,2]^\top) = sign(6) = +1$$

The data point x_1 is correctly classified, so we do not need to update the model.

3.

$$sign(w^\top \phi(x_2)) = sign([2,1,1] \cdot [-1,-2,2]^\top) = sign(-2) = -1$$

The data point x_2 is misclassified, so we have to update the model:

$$w \leftarrow w + \alpha \phi(x_2) t_2 = [2,1,1]^\top + [-1,-2,2]^\top = [1,-1,3]^\top$$

4. No, since we have updated the classifier, we should check if all the other data points in the dataset are still correctly classified.