Classification

3.1 k nearest neighbor

The table below is the training data set with $n = 6$ observations of a 3-dimensional categorical input $\mathbf{x} = [x_1\ x_2\ x_3]^\mathsf{T}$ and 1 categorical output y (the color green or red).

i	$\mathbf{x}$			y
1	0	3	0	Red
2	2	0	0	Red
3	0	1	3	Red
4	0	1	2	Green
5	−1	0	1	Green
6	1	1	1	Red

(a) Compute the Euclidean distance between each observation in the training data, and the test point $\mathbf{x}_\star = [0\ 0\ 0]^\mathsf{T}$.

(b) What is our prediction for the test point $\mathbf{x}_\star$, if we use k-NN with $k = 1$?

(c) What is our prediction for the test point $\mathbf{x}_\star$, if we use k-NN with $k = 3$?

3.2 Logistic regression

Suppose we collect data from a group of students in a Machine learning class with variables x_1 = hours studied, x_2 = grade point average, and y = a binary output if that student received grade 5 ($y = 1$) or not ($y = -1$). We learn a logistic regression model

$$p(y = 1 \,|\, \mathbf{x}) = \frac{e^{\theta_0 + \theta_1 x_1 + \theta_2 x_2}}{1 + e^{\theta_0 + \theta_1 x_1 + \theta_2 x_2}} \tag{3.1}$$

with parameters $\widehat{\theta}_0 = -6$, $\widehat{\theta}_1 = 0.05$, $\widehat{\theta}_2 = 1$.

(a) Estimate the probability according to the logistic regression model that a student who studies for 40 h and has the grade point average of 3.5 gets a 5 in the Machine learning class.

(b) According to the logistic regression model, how many hours would the student in part (a) need to study to have 50% chance of getting a 5 in the class?

3.3 Difference between LDA and QDA

We now examine the differences between LDA and QDA.

(a) If the optimal boundary is linear, do we expect LDA or QDA to perform better on the training set? What do we expect on the test set?

(b) If the optimal decision boundary is nonlinear, do we expect LDA or QDA to perform better on the training set? What do we expect on the test set?

(c) In general, as the sample size n increases, do we expect the test error rate of QDA relative to LDA to increase, decrease or be unchanged? Why?

(d) True or false: Even if the optimal decision boundary for a given problem is linear, we will probably achieve a smaller test error rate using QDA rather than LDA because QDA is flexible enough to model a linear decision boundary. Justify your answer.

3.4 The optimal classifier

Suppose you work at a clinic in a field mission, and want to predict whether a patient has a particular (potentially deadly) disease or not. You do have a limited supply of an effective drug, which however has severe side effects. Due to unfortunate circumstances, the only diagnosis tool you have access to is a clinical thermometer, with which you can measure the body temperature of the patient. From previous studies made on the disease, you know the following:

- the distribution of body temperatures in infected patients is approximately Gaussian distributed with mean 38.5 °C and standard deviation 1 °C.
- the distribution of body temperatures in patients not infected by the disease (either healthy or infected by other diseases) is approximately Gaussian with mean 37.5 °C and standard deviation $\sqrt{0.5}$ °C.
- the prevalence of the disease is 5% (i.e., 5% of the population is infected).

The body temperatures of three patients are as follows: patient A 38.5 °C, patient B 39.2 °C, and patient C 40.1 °C.

(a) What is the probability that patient A, B and C are infected by the disease, respectively?
Hint: Use

$$p(y \mid \mathbf{x}) = \frac{p(\mathbf{x} \mid y)p(y)}{\sum_{m=1}^{M} p(\mathbf{x} \mid m)p(m)},$$

where $p(y)$ is the prior probability of class y, and $p(\mathbf{x} \mid y)$ the probability density of $\mathbf{x}$ for an observation from class y. (Bayes' theorem)

(b) Which prediction should you make for each patient, in order to make on average as few misclassifications as possible? (Hint: The most probable one. (Bayes' classifier))

(c) Argue why another performance metric other than the standard accuracy ('on average as few misclassifications as possible') should be considered for this problem. How would that affect your decisions in (b)?

3.5 Error rates

Suppose that we take a data set, divide it into equally-sized training and test sets, and then try out two different classification procedures. First we use logistic regression and get an error rate of 20% on the training data and 30% on the test data. Next, we use 1-nearest neighbors (i.e. k-NN with $k = 1$) and get an average error rate (averaged over both test and training data sets) of 18%. Based on these results, which method should we prefer to use for classification of new observations? Why?

3.6 Quadratic Discriminant Analysis

Consider a classification problem with the input $\mathbf{x} \in \mathbb{R}^p$ and output $y \in \{1, \dots, K\}$. Consider also Bayes' classifier

$$\widehat{y} = \underset{k=\{1,\dots,K\}}{\arg\max}\ p(y = k \mid \mathbf{x}), \qquad \text{where} \qquad p(y \mid \mathbf{x}) = \frac{p(\mathbf{x} \mid y)p(y)}{\sum_{k=1}^{K} p(\mathbf{x} \mid k)p(k)}. \tag{3.2}$$

In Quadratic Discriminant Analysis (QDA), we assume that $p(\mathbf{x} \mid y)$ is a multivariate Gaussian density with mean $\boldsymbol{\mu}_k$ and covariance $\boldsymbol{\Sigma}_k$

$$p(\mathbf{x} \mid y = k) = \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \frac{1}{(2\pi)^{p/2}|\boldsymbol{\Sigma}_k|^{1/2}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1}(\mathbf{x}-\boldsymbol{\mu}_k)} \tag{3.3a}$$

$$p(y = k) = \pi_k \tag{3.3b}$$

(each with a different $\boldsymbol{\mu}_k$, $\boldsymbol{\Sigma}_k$ and π_k).

(a) Show that if one makes the assumption (3.3), then Bayes' classifier becomes

$$\widehat{y} = \underset{k=\{1,\dots,K\}}{\arg\max}\ \delta_k(\mathbf{x}), \quad \text{where} \quad \delta_k(\mathbf{x}) = -\frac{1}{2}\mathbf{x}^\top \boldsymbol{\Sigma}_k^{-1}\mathbf{x} + \mathbf{x}^\top \boldsymbol{\Sigma}_k^{-1}\boldsymbol{\mu}_k - \frac{1}{2}\boldsymbol{\mu}_k^\top \boldsymbol{\Sigma}_k^{-1}\boldsymbol{\mu}_k - \frac{1}{2}\log|\boldsymbol{\Sigma}_k| + \log \pi_k. \tag{3.4}$$

This is QDA, and $\delta_k(\mathbf{x})$ is called the discriminant function.
Hint: In lecture 4, an equivalent derivation for LDA was made, assuming $\boldsymbol{\Sigma}_k = \boldsymbol{\Sigma}$ is constant for all k. Look on that one and extend it accordingly for QDA by relaxing this assumption.

(b) Consider two classes k and l. Show that the decision boundary between these two classes is given by a quadratic function.

3.7 Curse of dimensionality

For large number of inputs p, some methods, such as the nonparametric k-NN, may perform bad due to the large dimensionality p of the input space. The problem is that the concept of 'near' or 'close' is very much depending on the number of dimensions p, and is commonly referred to as *the curse of dimensionality*. To investigate this, we will now consider an alternative version of the k-NN method, considering all neighbors within a fixed hypercube (instead of the k nearest) for making the decision.

(a) Suppose that $p = 1$, and that the inputs $\mathbf{x}$ are uniformly distributed on $[0, 1]$. We decide to consider all observations with an input within a ± 0.05 interval (as an alternative to using the k nearest observed inputs in the k-NN method) when making predictions. We now want to predict a test observation with input $X = 0.3$. On average, what fraction of all training observations will be used in making the prediction?

(b) Now consider the corresponding situation for $p = 2$: The inputs are uniformly distributed on $[0, 1] \times [0, 1]$, and for making predictions we use all training observations within ± 0.05 in each dimension. On average, what fraction of all training observations will we use when making a prediction for a test observation with input $\mathbf{x}_\star = [0.3\ 0.6]^\mathsf{T}$?

(c) In general, what fraction of all training observations will be used in predictions if there are p dimensions? As before, all inputs are uniformly distributed on $[0, 1]^p$ and for prediction we consider the training observations within ± 0.05 for each dimension. You may ignore the boundary effects if the test input is within 0.05 from the borders 0 or 1.

(d) Based on your answers to (a)-(c), argue why the prediction performance of k-NN might deteriorate for large p.

(e) If the inputs are distributed as in (c), and we want to make predictions using 10% of the training data inputs, which length should the side of a symmetric hypercube have, that covers on average 10% of the inputs?

Solutions

3.1 (a) The Euclidean distances are in the rightmost column below

i	$\mathbf{x}$			y	distance $\|\mathbf{x}-\mathbf{x}_\star\|$
1	0	3	0	Red	3
2	2	0	0	Red	2
3	0	1	3	Red	$\sqrt{10}\approx 3.2$
4	0	1	2	Green	$\sqrt{5}\approx 2.2$
5	-1	0	1	Green	$\sqrt{2}\approx 1.4$
6	1	1	1	Red	$\sqrt{3}\approx 1.7$

(b) k-NN with $k=1$, the closest observation, i.e., observation 5, is selected as prediction. Thus, the prediction is green.

(c) The 3 closest observation are observation 5, 6, and 2. Thus, the prediction is red.

3.2 (a) The probability of getting a 5 using the parameters $\widehat{\theta}_0=-6$, $\widehat{\theta}_1=0.05$ is

$$p(y=1\,|\,\mathbf{x})=\frac{e^{\widehat{\theta}_0+\widehat{\theta}_1x_1+\widehat{\theta}_2x_2}}{1+e^{\widehat{\theta}_0+\widehat{\theta}_1x_1+\widehat{\theta}_2x_2}}=\frac{e^{-6+0.051x_1+x_2}}{1+e^{-6+0.05x_1+1x_2}} \tag{3.5}$$

Now, with $x_1=40$ and $x_2=3.5$,

$$p(y=1\,|\,\mathbf{x})=\frac{e^{-6+0.05\,\cdot\,40+1\,\cdot\,3.5}}{1+e^{-6+0.05\,\cdot\,40+1\,\cdot\,3.5}}=\frac{e^{-0.5}}{1+e^{-0.5}} \tag{3.6}$$

$$=\frac{1}{1+e^{0.5}}\approx 38\%. \tag{3.7}$$

(b) Set $p(y=1\,|\,\mathbf{x})=0.5$ and $x_2=3.5$. This gives

$$0.5=\frac{e^{-6+0.05x_1+3.5}}{1+e^{-6+0.05x_1+3.5}}=\frac{1}{e^{2.5-0.05x_1}+1}\Rightarrow \tag{3.8}$$

$$0.5(1+e^{2.5-0.05x_1})=1\Rightarrow \tag{3.9}$$

$$e^{2.5-0.05x_1}=\frac{1}{0.5}-1=1\Rightarrow \tag{3.10}$$

$$2.5-0.05x_1=\log(1)=0\Rightarrow \tag{3.11}$$

$$x_1=\frac{2.5}{0.05}=50\text{ h}. \tag{3.12}$$

3.3 (a) We can always expect QDA to perform better than LDA on the *training* set because it is more flexible and is capable of fitting the training data better. If the optimal decision boundary is linear, we expect LDA to perform better on *test* data because it does not overfit.

(b) If the optimal decision boundary is nonlinear, we expect QDA to be able to perform better also on the test sets.

(c) In general we expect the test error rate to improve with QDA relative to LDA as the sample size n increases, since QDA is more flexible and will therefore be able to be closer to the optimal decision boundary. (However, for small n, it may overfit to the training data.)

(d) False. With few data points n, the QDA is likely to overfit (yielding a higher test error rate than LDA), which the LDA cannot if the Bayes decision boundary also is linear.

3.4 (a) We have the output y describing the patient status as $\{\text{infected}, \text{healthy}\}$, and the input $\mathbf{x}$ being the body temperature:

- $p(\mathbf{x} \,|\, y = \text{infected}) = \mathcal{N}(x \,|\, 38.5, 1)$
- $p(\mathbf{x} \,|\, y = \text{healthy}) = \mathcal{N}(x \,|\, 37.5, 0.5)$
- $p(\text{infected}) = 0.05$, and hence $p(\text{healthy}) = 0.95$

Inserting the expressions and patients temperatures into Bayes' theorem, we get

$$\begin{aligned} p(\text{Patient A is infected}) &= p(y = \text{infected} \,|\, \mathbf{x} = 38.5) = 0.09, \\ p(\text{patient B is infected}) &= p(y = \text{infected} \,|\, \mathbf{x} = 39.2) = 0.34, \\ p(\text{patient C is infected}) &= p(y = \text{infected} \,|\, \mathbf{x} = 40.1) = 0.90. \end{aligned}$$

(b) Predicting the most likely class minimizes the average number of misclassifications. For this problem, it gives the following predictions: patient A healthy, patient B healthy, patient C infected.

(c) Since the disease is deadly, there is an asymmetry in the problem. The consequences of falsely classifying an infected patient as healthy is probably worse than falsely classifying a healthy patient as infected (despite the side effects of the drug). A classifier designed with this asymmetry in mind would probably also predict patient B, and perhaps also A, as infected.

A useful tool for such a design could be the confusion matrix.

3.5 Logistic regression has a training error rate of $P_{\text{training}} = 20\%$ and test error rate of $P_{\text{test}} = 30\%$. k-NN ($k = 1$): average error rate of $\frac{P_{\text{training}} + P_{\text{test}}}{2} = 18\%$.

However, for k-NN with k=1, the training error rate is $P_{\text{training}} = 0\%$ because for any training observation, its nearest neighbor will be the response itself. So, k-NN has a test error rate of $P_{\text{training}} = 36\%$. I would choose logistic regression because of its lower test error rate of 30%.

3.6 (a) Since the denominator in (3.2) does not depend on k, we get

$$\begin{aligned} \widehat{y} &= \underset{k=\{1,\ldots,K\}}{\arg\max}\ p(y = k|\mathbf{x}) \\ &= \underset{k=\{1,\ldots,K\}}{\arg\max}\ p(\mathbf{x} \,|\, k)p(k) \\ &= \underset{k=\{1,\ldots,K\}}{\arg\max}\ \log\Big(p(\mathbf{x} \,|\, k)p(k)\Big) \end{aligned}$$

Further, we see that

$$\begin{aligned} \log\Big(p(\mathbf{x} \,|\, k)p(k)\Big) &= \log\Big(p(k)\Big) + \log\Big(p(\mathbf{x} \,|\, k)\Big) \\ &= -\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_k)^\mathsf{T}\boldsymbol{\Sigma}_k^{-1}(\mathbf{x} - \boldsymbol{\mu}_k) - \frac{1}{2}\log|\boldsymbol{\Sigma}_k| + \log(\pi_k) + \underbrace{\text{const.}}_{\text{independent of } k} \\ &= \underbrace{-\frac{1}{2}\mathbf{x}^\mathsf{T}\boldsymbol{\Sigma}_k^{-1}\mathbf{x} + \mathbf{x}^\mathsf{T}\boldsymbol{\Sigma}_k^{-1}\boldsymbol{\mu}_k - \frac{1}{2}\boldsymbol{\mu}_k^\mathsf{T}\boldsymbol{\Sigma}_k^{-1}\boldsymbol{\mu}_k - \frac{1}{2}\log|\boldsymbol{\Sigma}_k| + \log(\pi_k)}_{=\delta_k(\mathbf{x})} + \underbrace{\text{const.}}_{\text{independent of } k} \end{aligned}$$

where the classification problem can be written as

$$\widehat{y} = \underset{k=\{1,\dots,K\}}{\arg\max}\ \delta_k(\mathbf{x}). \tag{3.13}$$

(b) Compare two classes $y = k$ and $y = l$. The decision boundary between these two classes is given by

$$p(y = k|\mathbf{x}) = p(y = l|\mathbf{x}) \Rightarrow \delta_k(\mathbf{x}) - \delta_l(\mathbf{x}) = 0,$$

i.e., where the predicted probability for each of the two classes are equally high.

This gives

$$\begin{aligned}\delta_k(\mathbf{x}) - \delta(_l(\mathbf{x})) &= -\frac{1}{2}\mathbf{x}^\mathsf{T}\boldsymbol{\Sigma}_k^{-1}\mathbf{x} + \mathbf{x}^\mathsf{T}\boldsymbol{\Sigma}_k^{-1}\boldsymbol{\mu}_k - \frac{1}{2}\boldsymbol{\mu}_k^\mathsf{T}\boldsymbol{\Sigma}_k^{-1}\boldsymbol{\mu}_k - \frac{1}{2}\log|\boldsymbol{\Sigma}_k| + \log(\pi_k) \\ &\quad - \left(-\frac{1}{2}\mathbf{x}^\mathsf{T}\boldsymbol{\Sigma}_l^{-1}\mathbf{x} + \mathbf{x}^\mathsf{T}\boldsymbol{\Sigma}_l^{-1}\boldsymbol{\mu}_l - \frac{1}{2}\boldsymbol{\mu}_l^\mathsf{T}\boldsymbol{\Sigma}_l^{-1}\boldsymbol{\mu}_l - \frac{1}{2}\log|\boldsymbol{\Sigma}_l| + \log(\pi_l)\right) \\ &= -\frac{1}{2}\mathbf{x}^\mathsf{T}(\boldsymbol{\Sigma}_k^{-1} - \boldsymbol{\Sigma}_l^{-1})\mathbf{x} + \mathbf{x}^\mathsf{T}(\boldsymbol{\Sigma}_k^{-1}\boldsymbol{\mu}_k - \boldsymbol{\Sigma}_l^{-1}\boldsymbol{\mu}_l) \\ &\quad - \frac{1}{2}\boldsymbol{\mu}_k^\mathsf{T}\boldsymbol{\Sigma}_k^{-1}\boldsymbol{\mu}_k - \frac{1}{2}\log|\boldsymbol{\Sigma}_k| + \log(\pi_k) + \frac{1}{2}\boldsymbol{\mu}_l^\mathsf{T}\boldsymbol{\Sigma}_l^{-1}\boldsymbol{\mu}_l + \frac{1}{2}\log|\boldsymbol{\Sigma}_l| - \log(\pi_l),\end{aligned}$$

which is a quadratic function in $\mathbf{x}$ as long as $\boldsymbol{\Sigma}_k \neq \boldsymbol{\Sigma}_l$.

3.7 (a) All training observations with inputs on the interval $[0.25, 0.35]$ will be used for making the prediction. Since the inputs are uniformly distributed on the interval $[0, 1]$, will 10% on the average be in $[0.25, 0.35]$, and hence be used for the prediction.

(b) In this case, we will use all training observations with inputs in the square $[0.25, 0.35] \times [0.55, 0.65]$. The square covers 1% of $[0, 1] \times [0, 1]$, and hence will on average only 1% of the training observations be used in the predictions.

(c) The probability of an input to be inside the hypercube with dimensions $0.1 \times \cdots \times 0.1$ is 0.1 per dimension, and thus 0.1^p for all dimensions.

(d) If p is large, the nearest neighbor to a test input might still be quite far away from the test input: when $p = 1$ in the situation in (a), about 10% of the training data could be expected to be within ± 0.05, and we can thus expect to find an observation 'similar' to the test case among the training data, yielding a hopefully useful prediction. If $p = 10$ in (c), only 0.000001% of the training data could be expected to be within the hypercube ± 0.05 for each dimension around the test input, and it seems less likely that we have an observation among the training data that is 'similar' to the test case, and the prediction performance may therefore deteriorate.

(e)
- For $p = 1$, the side needs to be 0.1.
- For $p = 2$, the side needs to be $0.1^{1/2} = 0.316$.
- For $p = 3$, the side needs to be $0.1^{1/3} = 0.464$.
- ...
- For p, the side needs to be $0.1^{1/p}$.

Thus, if the number of input is high, let us say $p = 100$, the side of the cube needs to be 0.977. This means that if we want on average to use 10% of the training data for a prediction, we would need to include almost the entire range of each input dimension in order to achieve that, because of the large number of dimensions.