Brief introduction to machine learning

Pavlo Mozharovskyi[1]

(with contributions of Laurent Rouviere[2] and Valentin Patilea[3])

[1]LTCI, Télécom Paris, Institut Polytechnique de Paris
[2]Université Rennes 2
[3]Ensai, CREST

Machine learning

Paris, March 12, 2022

Today

The task of classification and Bayes classifier

Linear discriminant analysis

k-nearest neighbors and the curse of dimension

Outlook

Literature

Learning materials include but are not limited to:

- Hastie, T., Tibshirani, R., and Friedman, J. (2009).
  The Elements of Statistics Learning: Data Mining, Inference, and Prediction (Second Edition).
  Springer.
  - Chapter 2.
  - Section 4.3.

- Slides of the lecture.

Contents

Binary supervised classification

Notation:

- **Given:** for the random pair (X, Y) in $\mathbb{R}^d \times \{0, 1\}$ consisting of a random observation X and its random binary label Y (class), a sample of n i.i.d.: $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$.

Binary supervised classification

Notation:

- **Given:** for the random pair (X, Y) in $\mathbb{R}^d \times \{0, 1\}$ consisting of a random observation X and its random binary label Y (class), a sample of n i.i.d.: $(\mathbf{x}_1, y_1), ..., (\mathbf{x}_n, y_n)$.
- **Goal:** predict the label of the new (unseen before) observation $\mathbf{x}$.

Binary supervised classification

Notation:

- **Given:** for the random pair (X, Y) in $\mathbb{R}^d \times \{0, 1\}$ consisting of a random observation X and its random binary label Y (class), a sample of n i.i.d.: $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$.
- **Goal:** predict the label of the new (unseen before) observation $\mathbf{x}$.
- **Method:** construct a classification rule:

$$g : \mathbb{R}^d \to \{0, 1\}, \mathbf{x} \mapsto g(\mathbf{x}),$$

so $g(\mathbf{x})$ is the prediction of the label for observation $\mathbf{x}$.

Binary supervised classification

Notation:

- **Given:** for the random pair (X, Y) in $\mathbb{R}^d \times \{0, 1\}$ consisting of a random observation X and its random binary label Y (class), a sample of n i.i.d.: $(\mathbf{x}_1, y_1), ..., (\mathbf{x}_n, y_n)$.
- **Goal:** predict the label of the new (unseen before) observation $\mathbf{x}$.
- **Method:** construct a classification rule:

$$g : \mathbb{R}^d \to \{0, 1\}\,, \, \mathbf{x} \mapsto g(\mathbf{x})\,,$$

  so $g(\mathbf{x})$ is the prediction of the label for observation $\mathbf{x}$.
- **Criterion:** of the performance of g is the **error probability**:

$$R(g) = \mathbb{P}[g(X) \neq Y]$$

Binary supervised classification

Notation:

- **Given:** for the random pair (X, Y) in $\mathbb{R}^d \times \{0,1\}$ consisting of a random observation X and its random binary label Y (class), a sample of n i.i.d.: $(\mathbf{x}_1, y_1), ..., (\mathbf{x}_n, y_n)$.
- **Goal:** predict the label of the new (unseen before) observation $\mathbf{x}$.
- **Method:** construct a classification rule:

$$g : \mathbb{R}^d \to \{0,1\}, \mathbf{x} \mapsto g(\mathbf{x}),$$

  so $g(\mathbf{x})$ is the prediction of the label for observation $\mathbf{x}$.
- **Criterion:** of the performance of g is the **error probability**:

$$R(g) = \mathbb{P}[g(X) \neq Y] = \mathbb{E}[1(g(X) \neq Y)].$$

Binary supervised classification

Notation:

- **Given:** for the random pair (X, Y) in $\mathbb{R}^d \times \{0, 1\}$ consisting of a random observation X and its random binary label Y (class), a sample of n i.i.d.: $(\mathbf{x}_1, y_1), ..., (\mathbf{x}_n, y_n)$.
- **Goal:** predict the label of the new (unseen before) observation $\mathbf{x}$.
- **Method:** construct a classification rule:

$$g \,:\, \mathbb{R}^d \to \{0, 1\}\,,\, \mathbf{x} \mapsto g(\mathbf{x})\,,$$

  so $g(\mathbf{x})$ is the prediction of the label for observation $\mathbf{x}$.
- **Criterion:** of the performance of g is the **error probability**:

$$R(g) = \mathbb{P}[g(X) \neq Y] = \mathbb{E}[1\big(g(X) \neq Y\big)]\,.$$

- In practice the error probability will be replaced by the *empirical error*:

$$R_n(g) = \frac{1}{n}\sum_{i=1}^{n} 1\big(g(\mathbf{x}_i) \neq y_i\big)\,.$$

The Bayes classifier

- **The 'best' situation:** is to know

$$\eta(\mathbf{x}) = \mathbb{E}[Y|X = \mathbf{x}] = \mathbb{P}(Y = 1 \mid X = \mathbf{x}) \,.$$

The Bayes classifier

- **The 'best' situation:** is to know

$$\eta(\mathbf{x}) = \mathbb{E}[Y|X = \mathbf{x}] = \mathbb{P}(Y = 1 \mid X = \mathbf{x})\,.$$

- The *Bayes classifier* is the rule

$$g^*(\mathbf{x}) = \begin{cases} 1 & \text{if} \qquad \eta(\mathbf{x}) > 1/2\,, \\ 0 & \text{otherwise}\,. \end{cases}$$

Bayes classification rule

Bayes formula for the probability of event A conditioned on event B:

$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)}\,.$$

Bayes classification rule

Bayes formula for the probability of event A conditioned on event B:

$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)}\,.$$

In the context of binary supervised classification:

$$P(Y=0|X=\mathbf{x}) = \frac{P(X=\mathbf{x}|Y=0)\,P(Y=0)}{P(X=\mathbf{x})}$$

Bayes classification rule

Bayes formula for the probability of event A conditioned on event B:

$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)}\,.$$

In the context of binary supervised classification:

$$P(Y=0|X=\mathbf{x}) = \frac{P(X=\mathbf{x}|Y=0)\,P(Y=0)}{P(X=\mathbf{x})}$$

and

$$P(Y=1|X=\mathbf{x}) = \frac{P(X=\mathbf{x}|Y=1)\,P(Y=1)}{P(X=\mathbf{x})}\,.$$

Bayes classification rule

Bayes formula for the probability of event A conditioned on event B:

$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)}\,.$$

In the context of binary supervised classification:

$$P(Y=0|X=\mathbf{x}) = \frac{P(X=\mathbf{x}|Y=0)\,P(Y=0)}{P(X=\mathbf{x})}$$

and

$$P(Y=1|X=\mathbf{x}) = \frac{P(X=\mathbf{x}|Y=1)\,P(Y=1)}{P(X=\mathbf{x})}\,.$$

When deciding which class to assign $\mathbf{x}$ we choose "1" if

$$P(Y=1|X=\mathbf{x}) > P(Y=0|X=\mathbf{x}) \quad \text{or} \quad \frac{P(Y=1|X=\mathbf{x})}{P(Y=0|X=\mathbf{x})} > 1\,.$$

Bayes classification rule

Bayes formula for the probability of event A conditioned on event B:

$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)}\,.$$

In the context of binary supervised classification:

$$P(Y=0|X=\mathbf{x}) = \frac{P(X=\mathbf{x}|Y=0)\,P(Y=0)}{P(X=\mathbf{x})}$$

and

$$P(Y=1|X=\mathbf{x}) = \frac{P(X=\mathbf{x}|Y=1)\,P(Y=1)}{P(X=\mathbf{x})}\,.$$

When deciding which class to assign $\mathbf{x}$ we choose "1" if

$$P(Y=1|X=\mathbf{x}) > P(Y=0|X=\mathbf{x}) \quad \text{or} \quad \frac{P(Y=1|X=\mathbf{x})}{P(Y=0|X=\mathbf{x})} > 1\,.$$

So choose "1" if $\dfrac{P(X=\mathbf{x}|Y=1)\,P(Y=1)}{P(X=\mathbf{x}|Y=0)\,P(Y=0)} = \dfrac{f_1(\mathbf{x})\pi_1}{f_0(\mathbf{x})\pi_0} > 1$

Bayes classification rule

Bayes formula for the probability of event A conditioned on event B:

$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)}\,.$$

In the context of binary supervised classification:

$$P(Y=0|X=\mathbf{x}) = \frac{P(X=\mathbf{x}|Y=0)\,P(Y=0)}{P(X=\mathbf{x})}$$

and

$$P(Y=1|X=\mathbf{x}) = \frac{P(X=\mathbf{x}|Y=1)\,P(Y=1)}{P(X=\mathbf{x})}\,.$$

When deciding which class to assign $\mathbf{x}$ we choose "1" if

$$P(Y=1|X=\mathbf{x}) > P(Y=0|X=\mathbf{x}) \quad \text{or} \quad \frac{P(Y=1|X=\mathbf{x})}{P(Y=0|X=\mathbf{x})} > 1\,.$$

So choose "1" if $\dfrac{P(X=\mathbf{x}|Y=1)\,P(Y=1)}{P(X=\mathbf{x}|Y=0)\,P(Y=0)} = \dfrac{f_1(\mathbf{x})\pi_1}{f_0(\mathbf{x})\pi_0} > 1$ and "0" if not.

Contents

Iris data

Fisher's iris data:

Iris data

Fisher's iris data: is this the same flower?

Iris data

Fisher's iris data: is this the same flower?

Iris **setosa**

Iris **versicolor**

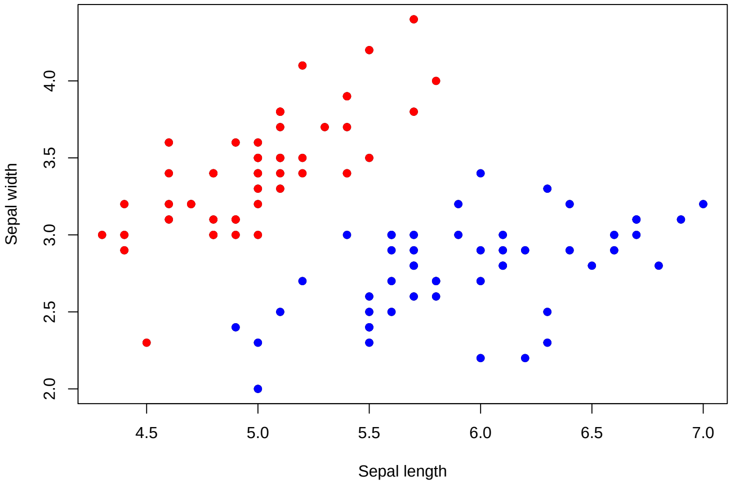
Iris data – description

- Three species of Iris (Iris *setosa*, Iris *virginica* and Iris *versicolor*) have been sampled.
- Four features were measured from each sample: the length and the width of the sepals and petals, in centimeters.
- The scatterplot indicates Iris *setosa* having features different from Iris *virginica* and Iris *versicolor* which appear to be quite similar

Iris data

Iris **setosa**		Iris **versicolor**	
Sepal length (cm)	Sepal width (cm)	Sepal length (cm)	Sepal width (cm)
5.1	3.5	7	3.2
4.9	3	6.4	3.2
4.7	3.2	6.9	3.1
4.6	3.1	5.5	2.3
5	3.6	6.5	2.8
5.4	3.9	5.7	2.8
4.6	3.4	6.3	3.3
5	3.4	4.9	2.4
4.4	2.9	6.6	2.9
...	...	...	...
...	...	...	...
...	...	...	...
4.6	3.2	6.2	2.9
5.3	3.7	5.1	2.5
5	3.3	5.7	2.8

Iris data



Linear discriminant analysis

- **Assumptions:**
  - X given Y admits a density
  - Both classes are normally distributed with the same covariance matrix, *i.e.* $X|Y=j \sim N(\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)$, $j=0,1$ or

$$f_j(\mathbf{x}) = \frac{1}{\sqrt{(2\pi)^d \det(\boldsymbol{\Sigma}_j)}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_j)^T \boldsymbol{\Sigma}_j^{-1}(\mathbf{x}-\boldsymbol{\mu}_j)} , \quad \text{for } j=0,1$$

$$\text{and} \quad \boldsymbol{\Sigma}_0 = \boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma} .$$

Linear discriminant analysis

- **Assumptions:**
  - X given Y admits a density
  - Both classes are normally distributed with the same covariance matrix, *i.e.* $X|Y = j \sim N(\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)$, $j = 0, 1$ or

$$f_j(\mathbf{x}) = \frac{1}{\sqrt{(2\pi)^d \det(\boldsymbol{\Sigma}_j)}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_j)^T \boldsymbol{\Sigma}_j^{-1}(\mathbf{x}-\boldsymbol{\mu}_j)}, \quad \text{for } j = 0, 1$$

$$\text{and} \quad \boldsymbol{\Sigma}_0 = \boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}\,.$$

- **Plug-in into Bayes:**

$$g(\mathbf{x}) = \begin{cases} 1 & \text{if } \frac{P(Y=1|X=\mathbf{x})}{P(Y=0|X=\mathbf{x})} > 1\,, \\ 0 & \text{else}\,; \end{cases}$$

Linear discriminant analysis

- **Assumptions:**
  - X given Y admits a density
  - Both classes are normally distributed with the same covariance matrix, *i.e.* $X|Y=j \sim N(\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)$, $j=0,1$ or

$$f_j(\mathbf{x}) = \frac{1}{\sqrt{(2\pi)^d \det(\boldsymbol{\Sigma}_j)}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_j)^T \boldsymbol{\Sigma}_j^{-1}(\mathbf{x}-\boldsymbol{\mu}_j)} , \quad \text{for } j=0,1$$

$$\text{and} \quad \boldsymbol{\Sigma}_0 = \boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma} .$$

- **Plug-in into Bayes:**

$$g(\mathbf{x}) = \begin{cases} 1 & \text{if } \frac{P(Y=1|X=\mathbf{x})}{P(Y=0|X=\mathbf{x})} > 1 , \\ 0 & \text{else} ; \end{cases}$$

$$\text{or} \quad g(\mathbf{x}) = 1\left(\log \frac{\pi_1 f_1(\mathbf{x})}{\pi_0 f_0(\mathbf{x})} > 0\right) .$$

Linear discriminant analysis

$$\log \frac{\pi_1 f_1(\mathbf{x})}{\pi_0 f_0(\mathbf{x})} \quad =$$

Linear discriminant analysis

$$\log \frac{\pi_1 f_1(\mathbf{x})}{\pi_0 f_0(\mathbf{x})} = \log \frac{\pi_1}{\pi_0} + \log \frac{\frac{1}{\sqrt{(2\pi)^d \det(\boldsymbol{\Sigma}_1)}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_1)^T \boldsymbol{\Sigma}_1^{-1}(\mathbf{x}-\boldsymbol{\mu}_1)}}{\frac{1}{\sqrt{(2\pi)^d \det(\boldsymbol{\Sigma}_0)}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_0)^T \boldsymbol{\Sigma}_0^{-1}(\mathbf{x}-\boldsymbol{\mu}_0)}}$$

Linear discriminant analysis

$$\log \frac{\pi_1 f_1(\mathbf{x})}{\pi_0 f_0(\mathbf{x})} = \log \frac{\pi_1}{\pi_0} + \log \frac{\frac{1}{\sqrt{(2\pi)^d \det(\boldsymbol{\Sigma}_1)}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_1)^T \boldsymbol{\Sigma}_1^{-1}(\mathbf{x}-\boldsymbol{\mu}_1)}}{\frac{1}{\sqrt{(2\pi)^d \det(\boldsymbol{\Sigma}_0)}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_0)^T \boldsymbol{\Sigma}_0^{-1}(\mathbf{x}-\boldsymbol{\mu}_0)}}$$

$$= \log \frac{\pi_1}{\pi_0} + \log \frac{\sqrt{\det(\boldsymbol{\Sigma}_0)}}{\sqrt{\det(\boldsymbol{\Sigma}_1)}}$$

$$+ \frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_0)^T \boldsymbol{\Sigma}_0^{-1}(\mathbf{x} - \boldsymbol{\mu}_0) - \frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_1)^T \boldsymbol{\Sigma}_1^{-1}(\mathbf{x} - \boldsymbol{\mu}_1)$$

Linear discriminant analysis

$$
\begin{aligned}
\log \frac{\pi_1 f_1(\mathbf{x})}{\pi_0 f_0(\mathbf{x})} \quad &= \quad \log \frac{\pi_1}{\pi_0} + \log \frac{\frac{1}{\sqrt{(2\pi)^d \det(\boldsymbol{\Sigma}_1)}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_1)^T \boldsymbol{\Sigma}_1^{-1} (\mathbf{x}-\boldsymbol{\mu}_1)}}{\frac{1}{\sqrt{(2\pi)^d \det(\boldsymbol{\Sigma}_0)}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_0)^T \boldsymbol{\Sigma}_0^{-1} (\mathbf{x}-\boldsymbol{\mu}_0)}} \\
&= \quad \log \frac{\pi_1}{\pi_0} + \log \frac{\sqrt{\det(\boldsymbol{\Sigma}_0)}}{\sqrt{\det(\boldsymbol{\Sigma}_1)}} \\
&+ \quad \frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_0)^T \boldsymbol{\Sigma}_0^{-1} (\mathbf{x} - \boldsymbol{\mu}_0) - \frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_1)^T \boldsymbol{\Sigma}_1^{-1} (\mathbf{x} - \boldsymbol{\mu}_1) \\
&= \quad \log \frac{\pi_1}{\pi_0} + \log \frac{\sqrt{\det(\boldsymbol{\Sigma}_0)}}{\sqrt{\det(\boldsymbol{\Sigma}_1)}} \\
&+ \quad \frac{1}{2}\left(\mathbf{x}^T \boldsymbol{\Sigma}_0^{-1} \mathbf{x} - \mathbf{x}^T \boldsymbol{\Sigma}_0^{-1} \boldsymbol{\mu}_0 - \boldsymbol{\mu}_0^T \boldsymbol{\Sigma}_0^{-1} \mathbf{x} + \boldsymbol{\mu}_0^T \boldsymbol{\Sigma}_0^{-1} \boldsymbol{\mu}_0\right) \\
&- \quad \frac{1}{2}\left(\mathbf{x}^T \boldsymbol{\Sigma}_1^{-1} \mathbf{x} - \mathbf{x}^T \boldsymbol{\Sigma}_1^{-1} \boldsymbol{\mu}_1 - \boldsymbol{\mu}_1^T \boldsymbol{\Sigma}_1^{-1} \mathbf{x} + \boldsymbol{\mu}_1^T \boldsymbol{\Sigma}_1^{-1} \boldsymbol{\mu}_1\right)
\end{aligned}
$$

Linear discriminant analysis

$$
\begin{aligned}
\log \frac{\pi_1 f_1(\mathbf{x})}{\pi_0 f_0(\mathbf{x})} &= \log \frac{\pi_1}{\pi_0} + \log \frac{\frac{1}{\sqrt{(2\pi)^d \det(\boldsymbol{\Sigma}_1)}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_1)^T \boldsymbol{\Sigma}_1^{-1}(\mathbf{x}-\boldsymbol{\mu}_1)}}{\frac{1}{\sqrt{(2\pi)^d \det(\boldsymbol{\Sigma}_0)}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_0)^T \boldsymbol{\Sigma}_0^{-1}(\mathbf{x}-\boldsymbol{\mu}_0)}} \\
&= \log \frac{\pi_1}{\pi_0} + \log \frac{\sqrt{\det(\boldsymbol{\Sigma}_0)}}{\sqrt{\det(\boldsymbol{\Sigma}_1)}} \\
&+ \frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_0)^T \boldsymbol{\Sigma}_0^{-1}(\mathbf{x}-\boldsymbol{\mu}_0) - \frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_1)^T \boldsymbol{\Sigma}_1^{-1}(\mathbf{x}-\boldsymbol{\mu}_1) \\
&= \log \frac{\pi_1}{\pi_0} + \log \frac{\sqrt{\det(\boldsymbol{\Sigma}_0)}}{\sqrt{\det(\boldsymbol{\Sigma}_1)}} \\
&+ \frac{1}{2}\left(\mathbf{x}^T \boldsymbol{\Sigma}_0^{-1}\mathbf{x} - \mathbf{x}^T \boldsymbol{\Sigma}_0^{-1}\boldsymbol{\mu}_0 - \boldsymbol{\mu}_0^T \boldsymbol{\Sigma}_0^{-1}\mathbf{x} + \boldsymbol{\mu}_0^T \boldsymbol{\Sigma}_0^{-1}\boldsymbol{\mu}_0\right) \\
&- \frac{1}{2}\left(\mathbf{x}^T \boldsymbol{\Sigma}_1^{-1}\mathbf{x} - \mathbf{x}^T \boldsymbol{\Sigma}_1^{-1}\boldsymbol{\mu}_1 - \boldsymbol{\mu}_1^T \boldsymbol{\Sigma}_1^{-1}\mathbf{x} + \boldsymbol{\mu}_1^T \boldsymbol{\Sigma}_1^{-1}\boldsymbol{\mu}_1\right) \\
&= \ldots
\end{aligned}
$$

Exploit $\boldsymbol{\Sigma}_0 = \boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}$ to simplify.

Linear discriminant analysis

$$
\begin{aligned}
\log \frac{\pi_1 f_1(\mathbf{x})}{\pi_0 f_0(\mathbf{x})} \quad = \quad & \log \frac{\pi_1}{\pi_0} + \log \frac{\sqrt{\det(\boldsymbol{\Sigma})}}{\sqrt{\det(\boldsymbol{\Sigma})}} \\
+ \quad & \frac{1}{2}\left(\mathbf{x}^T \boldsymbol{\Sigma}^{-1}\mathbf{x} - \mathbf{x}^T \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_0 - \boldsymbol{\mu}_0^T \boldsymbol{\Sigma}^{-1}\mathbf{x} + \boldsymbol{\mu}_0^T \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_0\right) \\
- \quad & \frac{1}{2}\left(\mathbf{x}^T \boldsymbol{\Sigma}^{-1}\mathbf{x} - \mathbf{x}^T \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_1 - \boldsymbol{\mu}_1^T \boldsymbol{\Sigma}^{-1}\mathbf{x} + \boldsymbol{\mu}_1^T \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_1\right)
\end{aligned}
$$

Linear discriminant analysis

$$\begin{aligned}
\log \frac{\pi_1 f_1(\mathbf{x})}{\pi_0 f_0(\mathbf{x})} &= \log \frac{\pi_1}{\pi_0} + \log \frac{\sqrt{\det(\mathbf{\Sigma})}}{\sqrt{\det(\mathbf{\Sigma})}} \\
&+ \frac{1}{2}\left(\mathbf{x}^T \mathbf{\Sigma}^{-1}\mathbf{x} - \mathbf{x}^T \mathbf{\Sigma}^{-1}\boldsymbol{\mu}_0 - \boldsymbol{\mu}_0^T \mathbf{\Sigma}^{-1}\mathbf{x} + \boldsymbol{\mu}_0^T \mathbf{\Sigma}^{-1}\boldsymbol{\mu}_0\right) \\
&- \frac{1}{2}\left(\mathbf{x}^T \mathbf{\Sigma}^{-1}\mathbf{x} - \mathbf{x}^T \mathbf{\Sigma}^{-1}\boldsymbol{\mu}_1 - \boldsymbol{\mu}_1^T \mathbf{\Sigma}^{-1}\mathbf{x} + \boldsymbol{\mu}_1^T \mathbf{\Sigma}^{-1}\boldsymbol{\mu}_1\right) \\
&= \log \frac{\pi_1}{\pi_0} + \frac{1}{2}\boldsymbol{\mu}_0^T \mathbf{\Sigma}^{-1}\boldsymbol{\mu}_0 - \frac{1}{2}\boldsymbol{\mu}_1^T \mathbf{\Sigma}^{-1}\boldsymbol{\mu}_1 \\
&+ \frac{1}{2}\mathbf{x}^T \mathbf{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0) + \frac{1}{2}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)^T \mathbf{\Sigma}^{-1}\mathbf{x}
\end{aligned}$$

Linear discriminant analysis

$$
\begin{aligned}
\log \frac{\pi_1 f_1(\mathbf{x})}{\pi_0 f_0(\mathbf{x})} &= \log \frac{\pi_1}{\pi_0} + \log \frac{\sqrt{\det(\boldsymbol{\Sigma})}}{\sqrt{\det(\boldsymbol{\Sigma})}} \\
&+ \frac{1}{2}\left(\mathbf{x}^T \boldsymbol{\Sigma}^{-1}\mathbf{x} - \mathbf{x}^T \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_0 - \boldsymbol{\mu}_0^T \boldsymbol{\Sigma}^{-1}\mathbf{x} + \boldsymbol{\mu}_0^T \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_0\right) \\
&- \frac{1}{2}\left(\mathbf{x}^T \boldsymbol{\Sigma}^{-1}\mathbf{x} - \mathbf{x}^T \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_1 - \boldsymbol{\mu}_1^T \boldsymbol{\Sigma}^{-1}\mathbf{x} + \boldsymbol{\mu}_1^T \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_1\right) \\
&= \log \frac{\pi_1}{\pi_0} + \frac{1}{2}\boldsymbol{\mu}_0^T \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_0 - \frac{1}{2}\boldsymbol{\mu}_1^T \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_1 \\
&+ \frac{1}{2}\mathbf{x}^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0) + \frac{1}{2}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)^T \boldsymbol{\Sigma}^{-1}\mathbf{x} \\
&= \log \frac{\pi_1}{\pi_0} - \frac{1}{2}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0) \\
&+ \mathbf{x}^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)\,.
\end{aligned}
$$

Linear discriminant analysis (algorithm)

- **Learning:**

Linear discriminant analysis (algorithm)

- **Learning:**

  Let

  - $I_0 = \{i \, : \, y_i = 0 \, , \, i = 1, ..., n\}$ $(n_0 = \# I_0)$;
  - $I_1 = \{i \, : \, y_i = 1 \, , \, i = 1, ..., n\}$ $(n_1 = \# I_1)$.

Linear discriminant analysis (algorithm)

- **Learning:**

  Let

  - $I_0 = \{i \,:\, y_i = 0\,,\, i = 1, ..., n\}$ $(n_0 = \# I_0)$;
  - $I_1 = \{i \,:\, y_i = 1\,,\, i = 1, ..., n\}$ $(n_1 = \# I_1)$.

  Estimate

  - Priors: $p_0 = \frac{n_0}{n}$, $p_1 = \frac{n_1}{n}$;
  - Means: $\bar{\mathbf{x}}_0 = \frac{1}{n_0} \sum_{i \in I_0} \mathbf{x}_i$, $\bar{\mathbf{x}}_1 = \frac{1}{n_1} \sum_{i \in I_1} \mathbf{x}_i$, $(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_0)$;
  - Common covariance matrix:

    $$\boldsymbol{S} = \frac{1}{n-2}\left(\sum_{i \in I_0} (\mathbf{x}_i - \bar{\mathbf{x}}_0)(\mathbf{x}_i - \bar{\mathbf{x}}_0)^T + \sum_{i \in I_1} (\mathbf{x}_i - \bar{\mathbf{x}}_1)(\mathbf{x}_i - \bar{\mathbf{x}}_1)^T\right) .$$

Linear discriminant analysis (algorithm)

- **Learning:**

  Let

  - $I_0 = \{i \ : \ y_i = 0\,,\ i = 1, ..., n\}$ $(n_0 = \# I_0)$;
  - $I_1 = \{i \ : \ y_i = 1\,,\ i = 1, ..., n\}$ $(n_1 = \# I_1)$.

  Estimate

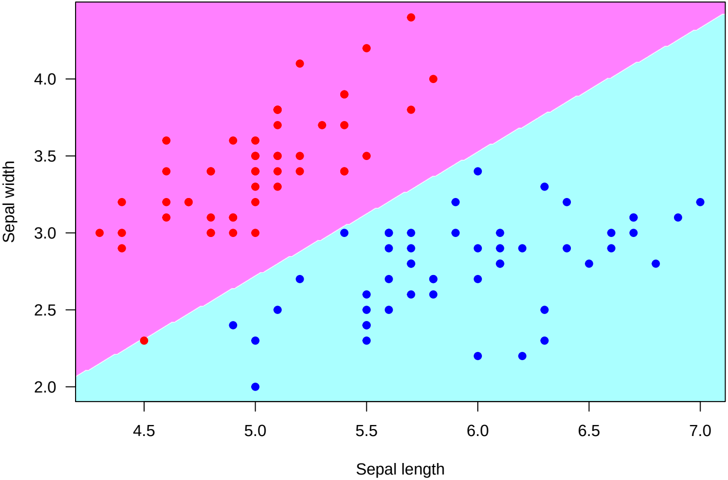
  - Priors: $p_0 = \frac{n_0}{n}$, $p_1 = \frac{n_1}{n}$;
  - Means: $\bar{\mathbf{x}}_0 = \frac{1}{n_0} \sum_{i \in I_0} \mathbf{x}_i$, $\bar{\mathbf{x}}_1 = \frac{1}{n_1} \sum_{i \in I_1} \mathbf{x}_i$, $(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_0)$;
  - Common covariance matrix:

    $$\boldsymbol{S} = \frac{1}{n-2}\left(\sum_{i \in I_0} (\mathbf{x}_i - \bar{\mathbf{x}}_0)(\mathbf{x}_i - \bar{\mathbf{x}}_0)^T + \sum_{i \in I_1} (\mathbf{x}_i - \bar{\mathbf{x}}_1)(\mathbf{x}_i - \bar{\mathbf{x}}_1)^T\right) .$$

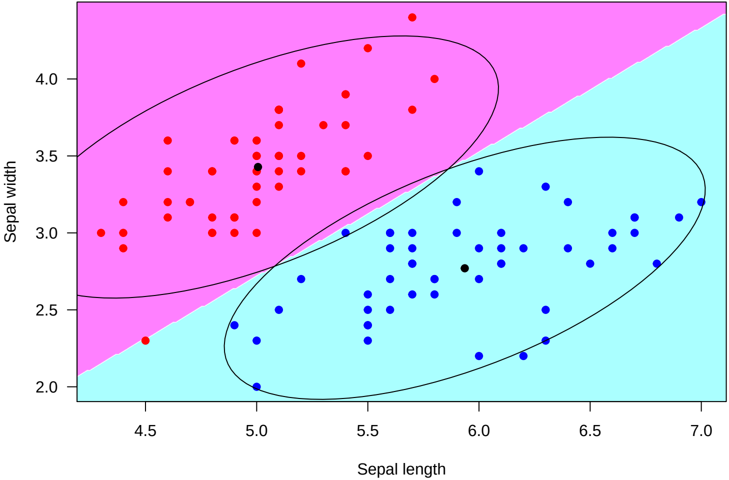
- **Classification:** For a new observation $\mathbf{x}$

$$g(\mathbf{x}) = \begin{cases} 1 & \text{if } \log \frac{p_1}{p_0} - \frac{1}{2}(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_0)^T \boldsymbol{S}^{-1} (\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_0) \\ & \quad + \mathbf{x}^T \boldsymbol{S}^{-1} (\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_0) > 0\,, \\ 0 & \text{otherwise}\,. \end{cases}$$

Linear discriminant analysis (iris data)



Linear discriminant analysis (iris data)



Linear discriminant analysis (closer look)

Assume $\pi_0 = \pi_1 = 0.5$:

- Bias-corrected discrimination function

$$T(\mathbf{x}) = (\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_0)^T \boldsymbol{S}^{-1}\left(\mathbf{x} - \frac{1}{2}(\bar{\mathbf{x}}_1 + \bar{\mathbf{x}}_0)\right) - \frac{n(n_1 - n_0)d}{2(n-d-1)n_0 n_1}\,.$$

Linear discriminant analysis (closer look)

Assume $\pi_0 = \pi_1 = 0.5$:

- Bias-corrected discrimination function

$$T(\mathbf{x}) = (\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_0)^T \boldsymbol{S}^{-1}\left(\mathbf{x} - \frac{1}{2}(\bar{\mathbf{x}}_1 + \bar{\mathbf{x}}_0)\right) - \frac{n(n_1 - n_0)d}{2(n - d - 1)n_0 n_1}\,.$$

- Let

$$u = (\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_0)^T \boldsymbol{S}^{-1}(\bar{\mathbf{x}}_1 - \boldsymbol{\mu}_1) - \frac{(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_0)^T \boldsymbol{S}^{-1}(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_0)}{2} + \frac{n(n_1 - n_0)d}{2(n - d - 1)n_0 n_1}\,,$$

$$v = (\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_0)^T \boldsymbol{S}^{-1}\boldsymbol{\Sigma}\boldsymbol{S}^{-1}(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_0)\,.$$

Linear discriminant analysis (closer look)

Assume $\pi_0 = \pi_1 = 0.5$:

- Bias-corrected discrimination function

$$T(\mathbf{x}) = (\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_0)^T \boldsymbol{S}^{-1}(\mathbf{x} - \frac{1}{2}(\bar{\mathbf{x}}_1 + \bar{\mathbf{x}}_0)) - \frac{n(n_1 - n_0)d}{2(n-d-1)n_0 n_1}\,.$$

- Let

$$u = (\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_0)^T \boldsymbol{S}^{-1}(\bar{\mathbf{x}}_1 - \boldsymbol{\mu}_1) - \frac{(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_0)^T \boldsymbol{S}^{-1}(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_0)}{2} + \frac{n(n_1 - n_0)d}{2(n-d-1)n_0 n_1}\,,$$

$$v = (\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_0)^T \boldsymbol{S}^{-1} \boldsymbol{\Sigma} \boldsymbol{S}^{-1}(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_0)\,.$$

- Discrimination function conditioned on data is distributed as

$$T(\mathbf{x}) | \bar{\mathbf{x}}_0, \bar{\mathbf{x}}_1, \boldsymbol{S} \sim N(-u, v)\,.$$

Linear discriminant analysis (closer look)

Assume $\pi_0 = \pi_1 = 0.5$:

- Bias-corrected discrimination function

$$T(\mathbf{x}) = (\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_0)^T \boldsymbol{S}^{-1}(\mathbf{x} - \frac{1}{2}(\bar{\mathbf{x}}_1 + \bar{\mathbf{x}}_0)) - \frac{n(n_1 - n_0)d}{2(n-d-1)n_0 n_1}\,.$$

- Let

$$u = (\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_0)^T \boldsymbol{S}^{-1}(\bar{\mathbf{x}}_1 - \boldsymbol{\mu}_1) - \frac{(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_0)^T \boldsymbol{S}^{-1}(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_0)}{2} + \frac{n(n_1 - n_0)d}{2(n-d-1)n_0 n_1}\,,$$

$$v = (\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_0)^T \boldsymbol{S}^{-1} \boldsymbol{\Sigma} \boldsymbol{S}^{-1}(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_0)\,.$$

- Discrimination function conditioned on data is distributed as

$$T(\mathbf{x})|\bar{\mathbf{x}}_0, \bar{\mathbf{x}}_1, \boldsymbol{S} \sim N(-u, v)\,.$$

- Error probability (for class "1")

$$R_1 = \mathbb{E}\big[\mathbb{P}\big(T(\mathbf{x}) \le 0|\mathbf{x}, y = 1\big)\big] = \mathbb{E}[\Phi(\frac{u}{\sqrt{v}})]\,.$$

Linear discriminant analysis (closer look)

Error probability R_1 can be consistently estimated:

$$\hat{R}_1 = \Phi\Big(\frac{\hat{u}_0}{\sqrt{\hat{v}_0}}\Big) ,$$

where

$$\begin{aligned}
\hat{u}_0 &= -\frac{\hat{\Delta}^2}{2(1-\frac{d}{n})} , \\
\hat{v}_0 &= \frac{1}{(1-\frac{d}{n})^3}\Big(\hat{\Delta}^2 + \frac{d}{n\pi_0\pi_1}\Big) , \\
\hat{\Delta}^2 &= \frac{n-d-1}{n}(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_0)^T \boldsymbol{S}^{-1}(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_0) - \frac{(n+2)d}{n_0 n_1} .
\end{aligned}$$

Linear discriminant analysis (closer look)

Error probability R_1 can be consistently estimated:

$$\hat{R}_1 = \Phi\Big(\frac{\hat{u}_0}{\sqrt{\hat{v}_0}}\Big) \, ,$$

where

$$\begin{aligned}
\hat{u}_0 &= -\frac{\hat{\Delta}^2}{2(1-\frac{d}{n})} \, , \\
\hat{v}_0 &= \frac{1}{(1-\frac{d}{n})^3}\Big(\hat{\Delta}^2 + \frac{d}{n\pi_0\pi_1}\Big) \, , \\
\hat{\Delta}^2 &= \frac{n-d-1}{n}(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_0)^T \boldsymbol{S}^{-1}(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_0) - \frac{(n+2)d}{n_0 n_1} \, .
\end{aligned}$$

Corollary

Under certain asymptotic framework it holds that

$$\hat{R}_1 \xrightarrow{p} R_1 \, .$$

Contents

k-nearest neighbors (algorithm)

For $\mathbf{x} \in \mathbb{R}^d$ and some integer $0 < k < n$, let a set $I_k(\mathbf{x})$ index the k-nearest neighbors of the point $\mathbf{x}$:

$$I_k(\mathbf{x}) = \{i(1), ..., i(k)\},$$

where $\|\mathbf{x} - \mathbf{x}_{i(1)}\| \leq \|\mathbf{x} - \mathbf{x}_{i(2)}\| \leq ... \leq \|\mathbf{x} - \mathbf{x}_{i(n)}\|$ is an ascending order.

k-nearest neighbors (algorithm)

For $\mathbf{x} \in \mathbb{R}^d$ and some integer $0 < k < n$, let a set $I_k(\mathbf{x})$ index the k-nearest neighbors of the point $\mathbf{x}$:

$$I_k(\mathbf{x}) = \{i(1), ..., i(k)\},$$

where $\|\mathbf{x} - \mathbf{x}_{i(1)}\| \leq \|\mathbf{x} - \mathbf{x}_{i(2)}\| \leq ... \leq \|\mathbf{x} - \mathbf{x}_{i(n)}\|$ is an ascending order.
k is to be set, *e.g.* chosen by the means of cross-validation.

k-nearest neighbors (algorithm)

For $\mathbf{x} \in \mathbb{R}^d$ and some integer $0 < k < n$, let a set $I_k(\mathbf{x})$ index the k-nearest neighbors of the point $\mathbf{x}$:

$$I_k(\mathbf{x}) = \{i(1), ..., i(k)\},$$

where $\|\mathbf{x} - \mathbf{x}_{i(1)}\| \leq \|\mathbf{x} - \mathbf{x}_{i(2)}\| \leq ... \leq \|\mathbf{x} - \mathbf{x}_{i(n)}\|$ is an ascending order. k is to be set, *e.g.* chosen by the means of cross-validation.

Then the k-**nearest neighbors** (kNN) algorithm **classifies** a new observation as follows:

k-nearest neighbors (algorithm)

For $\mathbf{x} \in \mathbb{R}^d$ and some integer $0 < k < n$, let a set $I_k(\mathbf{x})$ index the k-nearest neighbors of the point $\mathbf{x}$:

$$I_k(\mathbf{x}) = \{i(1), ..., i(k)\},$$

where $\|\mathbf{x} - \mathbf{x}_{i(1)}\| \leq \|\mathbf{x} - \mathbf{x}_{i(2)}\| \leq ... \leq \|\mathbf{x} - \mathbf{x}_{i(n)}\|$ is an ascending order. k is to be set, *e.g.* chosen by the means of cross-validation.

Then the k-**nearest neighbors** (kNN) algorithm **classifies** a new observation as follows:

- Calculate the ratio of classes' proportion in the k-neighborhood:

$$p_k(\mathbf{x}) = \frac{\sum_{i \in I_k(\mathbf{x})} 1(y_i = 1)}{\sum_{i \in I_k(\mathbf{x})} 1(y_i = 0)}.$$

k-nearest neighbors (algorithm)

For $\mathbf{x} \in \mathbb{R}^d$ and some integer $0 < k < n$, let a set $I_k(\mathbf{x})$ index the k-nearest neighbors of the point $\mathbf{x}$:

$$I_k(\mathbf{x}) = \{i(1), ..., i(k)\} \,,$$

where $\|\mathbf{x} - \mathbf{x}_{i(1)}\| \leq \|\mathbf{x} - \mathbf{x}_{i(2)}\| \leq ... \leq \|\mathbf{x} - \mathbf{x}_{i(n)}\|$ is an ascending order. k is to be set, *e.g.* chosen by the means of cross-validation.

Then the k-**nearest neighbors** (kNN) algorithm **classifies** a new observation as follows:

- Calculate the ratio of classes' proportion in the k-neighborhood:

$$p_k(\mathbf{x}) = \frac{\sum_{i \in I_k(\mathbf{x})} 1(y_i = 1)}{\sum_{i \in I_k(\mathbf{x})} 1(y_i = 0)} \,.$$

- Assign the class based on majority:

$$g(\mathbf{x}) = \begin{cases} 1 & \text{if } p_k(\mathbf{x}) > 1 \,, \\ 0 & \text{otherwise} \,. \end{cases}$$

k-nearest neighbors (algorithm)

For $\mathbf{x} \in \mathbb{R}^d$ and some integer $0 < k < n$, let a set $I_k(\mathbf{x})$ index the k-nearest neighbors of the point $\mathbf{x}$:

$$I_k(\mathbf{x}) = \{i(1), ..., i(k)\} ,$$

where $\|\mathbf{x} - \mathbf{x}_{i(1)}\| \leq \|\mathbf{x} - \mathbf{x}_{i(2)}\| \leq ... \leq \|\mathbf{x} - \mathbf{x}_{i(n)}\|$ is an ascending order. k is to be set, *e.g.* chosen by the means of cross-validation.

Then the k-**nearest neighbors** (kNN) algorithm **classifies** a new observation as follows:

- Calculate the ratio of classes' proportion in the k-neighborhood:

$$p_k(\mathbf{x}) = \frac{\sum_{i \in I_k(\mathbf{x})} 1(y_i = 1)}{\sum_{i \in I_k(\mathbf{x})} 1(y_i = 0)} .$$

- Assign the class based on majority:

$$g(\mathbf{x}) = \begin{cases} 1 & \text{if } p_k(\mathbf{x}) > 1 , \\ 0 & \text{otherwise} . \end{cases}$$

- Deal with ties, *e.g.* decide randomly, or choose odd ks only.

- Consider the kNN regression estimate of $\mathbb{P}(Y = 1 \mid X = \mathbf{x})$, (which, remember, here is equal to $\mathbb{E}(Y \mid X = \mathbf{x})$):

$$\widehat{\eta}(\mathbf{x}) = \widehat{\eta}_n(\mathbf{x}) = \sum_{i=1}^{n} w_{in}(x) y_i = \frac{1}{k} \sum_{i \in I_k(\mathbf{x})} y_i \,,$$

with

$$w_{in} = \frac{1(i \in I_k(\mathbf{x}))}{k} \,.$$

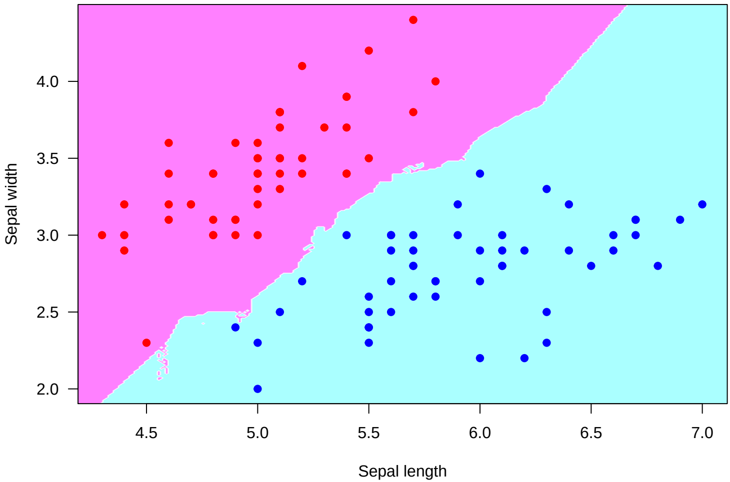
- **Remark:** the rule

$$g(\mathbf{x}) = 1(p_k(\mathbf{x}) > 1)$$

is equivalent to the rule

$$1(\widehat{\eta}(\mathbf{x}) > 1/2) \,.$$

k-nearest neighbors (iris data, k=9)



k-nearest neighbors classifier (universal consistency)

Under certain assumptions, kNN is universally consistent, *i.e.* approaches the classification error of the Bayes classifier with increasing length of the training sample n.

k-nearest neighbors classifier (universal consistency)

Under certain assumptions, kNN is universally consistent, *i.e.* approaches the classification error of the Bayes classifier with increasing length of the training sample n.

Theorem (Stone, 1977)

If $k \to \infty$ and $\frac{k}{n} \to 0$ then the kNN in $\mathbb{R}^d$ with Euclidean distance is universally consistent, i.e.

$$\lim_{n \to \infty} \mathbb{E}\Big[\int_X \big(\widehat{\eta}_n(\mathbf{x}) - \mathbb{E}[Y|X = \mathbf{x}]\big)^2 \mu_X(d\mathbf{x})\Big] = 0\,,$$

for any probability measure of (X, Y). Here, μ_X is the probability measure of X.

k-nearest neighbors classifier (universal consistency)

Under certain assumptions, kNN is universally consistent, *i.e.* approaches the classification error of the Bayes classifier with increasing length of the training sample n.

Theorem (Stone, 1977)

If $k \to \infty$ *and* $\frac{k}{n} \to 0$ *then the* k*NN in* $\mathbb{R}^d$ *with Euclidean distance is universally consistent, i.e.*

$$\lim_{n\to\infty} \mathbb{E}\Big[\int_X \big(\widehat{\eta}_n(\mathbf{x}) - \mathbb{E}[Y|X = \mathbf{x}]\big)^2 \mu_X(d\mathbf{x})\Big] = 0\,,$$

for any probability measure of (X, Y)*. Here,* μ_X *is the probability measure of* X*.*

In general for kernel-based methods with h being the bandwidth:

Theorem (Devroye-Krzyżak, 1989)

If $h \to 0$ *and* $nh^d \to +\infty$ *then the kernel-based classifier is universally consistent.*

Rate of convergence

Nonparametric methods suffer from the **curse of dimensionality**: if the number of exploratory variables is large, the spherical neighborhood is filled poorly, which reduces the rate of convergence.

Rate of convergence

Nonparametric methods suffer from the **curse of dimensionality**: if the number of exploratory variables is large, the spherical neighborhood is filled poorly, which reduces the rate of convergence.

Recall the kNN regression estimate:

$$\widehat{\eta}(\mathbf{x}) = \frac{1}{k} \sum_{i \in I_k(\mathbf{x})} y_i \,.$$

Rate of convergence

Nonparametric methods suffer from the **curse of dimensionality**: if the number of exploratory variables is large, the spherical neighborhood is filled poorly, which reduces the rate of convergence.

Recall the kNN regression estimate:

$$\widehat{\eta}(\mathbf{x}) = \frac{1}{k} \sum_{i \in I_k(\mathbf{x})} y_i \,.$$

Theorem (Györfi, Kohler, Krzyżak, Walk, 2002)

If the regression function is Lipschitz continuous then for the kNN estimator it holds

$$\mathbb{E}\Big[\int_X \big(\widehat{\eta}_n(\mathbf{x}) - \mathbb{E}[Y|X=\mathbf{x}]\big)^2 \mu_X(d\mathbf{x})\Big] = O(n^{-\frac{2}{d+2}})\,.$$

Rate of convergence

Nonparametric methods suffer from the **curse of dimensionality**: if the number of exploratory variables is large, the spherical neighborhood is filled poorly, which reduces the rate of convergence.

Recall the kNN regression estimate:

$$\widehat{\eta}(\mathbf{x}) = \frac{1}{k} \sum_{i \in I_k(\mathbf{x})} y_i \, .$$

Theorem (Györfi, Kohler, Krzyżak, Walk, 2002)

If the regression function is Lipschitz continuous then for the kNN estimator it holds

$$\mathbb{E}\Big[\int_X \big(\widehat{\eta}_n(\mathbf{x}) - \mathbb{E}[Y|X = \mathbf{x}]\big)^2 \mu_X(d\mathbf{x})\Big] = O(n^{-\frac{2}{d+2}}) \, .$$

In practice non-parametric estimators possess poor performance in high-dimensional spaces.

Contents

Possible solution: aggregation methods

Aggregation methods allow, to a certain extent, deal with

1. **curse of dimensionality**;

Possible solution: aggregation methods

Aggregation methods allow, to a certain extent, deal with

1. **curse of dimensionality**;
2. **sensibility** of the method w.r.t. the choice of parameters;

Possible solution: aggregation methods

Aggregation methods allow, to a certain extent, deal with

1. **curse of dimensionality**;
2. **sensibility** of the method w.r.t. the choice of parameters;
3. **preserve previous properties** while being computationally tractable.

Possible solution: aggregation methods

Aggregation methods allow, to a certain extent, deal with

1. **curse of dimensionality**;
2. **sensibility** of the method w.r.t. the choice of parameters;
3. **preserve previous properties** while being computationally tractable.

These proposed approaches are based on the **aggregation**, *i.e.*:

Possible solution: aggregation methods

Aggregation methods allow, to a certain extent, deal with

1. **curse of dimensionality**;
2. **sensibility** of the method w.r.t. the choice of parameters;
3. **preserve previous properties** while being computationally tractable.

These proposed approaches are based on the **aggregation**, *i.e.*:

1. construct an ensemble of $g_1, ..., g_B$ of **weak learning algorithms**;

Possible solution: aggregation methods

Aggregation methods allow, to a certain extent, deal with

1. **curse of dimensionality**;
2. **sensibility** of the method w.r.t. the choice of parameters;
3. **preserve previous properties** while being computationally tractable.

These proposed approaches are based on the **aggregation**, *i.e.*:

1. construct an ensemble of $g_1, ..., g_B$ of **weak learning algorithms**;
2. aggregate them into the **final classifier**

$$g(\mathbf{x}) = \frac{1}{B} \sum_{k=1}^{B} g_k(\mathbf{x}) .$$

Possible solution: aggregation methods

Aggregation methods allow, to a certain extent, deal with

1. **curse of dimensionality**;
2. **sensibility** of the method w.r.t. the choice of parameters;
3. **preserve previous properties** while being computationally tractable.

These proposed approaches are based on the **aggregation**, *i.e.*:

1. construct an ensemble of $g_1, ..., g_B$ of **weak learning algorithms**;
2. aggregate them into the **final classifier**

$$g(\mathbf{x}) = \frac{1}{B} \sum_{k=1}^{B} g_k(\mathbf{x}) \, .$$

The key concepts:

Possible solution: aggregation methods

Aggregation methods allow, to a certain extent, deal with

1. **curse of dimensionality**;
2. **sensibility** of the method w.r.t. the choice of parameters;
3. **preserve previous properties** while being computationally tractable.

These proposed approaches are based on the **aggregation**, *i.e.*:

1. construct an ensemble of $g_1, ..., g_B$ of **weak learning algorithms**;
2. aggregate them into the **final classifier**

$$g(\mathbf{x}) = \frac{1}{B} \sum_{k=1}^{B} g_k(\mathbf{x}) .$$

The key concepts:

- **bagging and random forests**;
- **boosting**.

Thank you for your attention!

And some more references

- Devroye, L., Györfi, L., Lugosi, G. (1996).
  A Probabilistic Theory of Pattern Recognition.
  Springer.
- Györfi, L., Kohler, M., Krzyżak, A., Walk, H. (2002).
  A distribution-free theory of nonparametric regression.
  Springer.
- Hastie, T., Tibshirani, R., and Friedman, J. (2009).
  The Elements of Statistics Learning: Data Mining, Inference, and Prediction (Second Edition).
  Springer.
- Stone, C.J. (1977).
  Consistent nonparametric regression.
  The Annals of Statistics, 55(4), 595–645.
- Vapnik, V. N. (1998).
  Statistical Learning Theory.
  John Wiley & Sons.