Q & A: Nearest Neighbor Methods

Lecture series „Machine Learning“

Niels Landwehr

Research Group „Data Science“
Institute of Computer Science
University of Hildesheim

Quiz: Bayesian Linear Regression

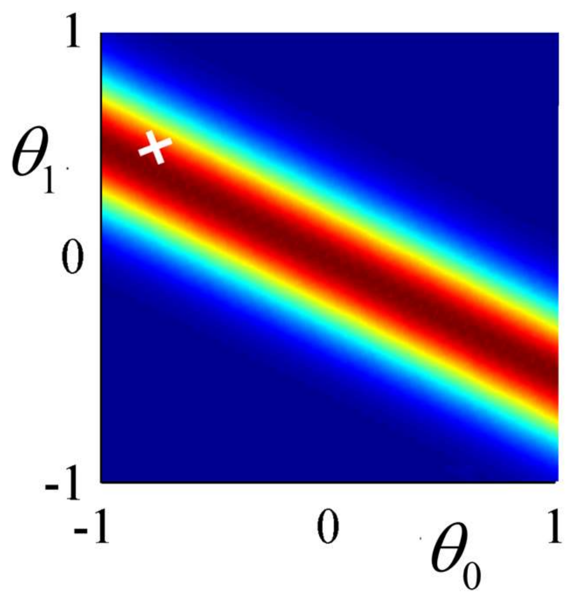
- We study a univariate Bayesian linear regression model

$$f_{\boldsymbol{\theta}}(x) = \theta_0 + \theta_1 x = \mathbf{x}^{\mathrm{T}}\boldsymbol{\theta} \qquad \boldsymbol{\theta} = (\theta_0, \theta_1)^{\mathrm{T}}$$

  with the usual predictive distribution

$$p(y \mid x, \boldsymbol{\theta}) = \mathcal{N}(y \mid f_{\boldsymbol{\theta}}(x), \sigma^2)$$

- **Question**: Which data point (x_1, y_1) will result in the following likelihood function:
  - $(x_1, y_1) = (1,1)$
  - $(x_1, y_1) = (0,-2)$
  - $(x_1, y_1) = (-1,1)$
  - $(x_1, y_1) = (2,0)$



$p(y_1 \mid x_1, \boldsymbol{\theta})$
plotted as a function of $\boldsymbol{\theta}$

Quiz: Bayesian Linear Regression

- **Solution:**
- The likelihood function is

$$p(y_1 \mid x_1, \boldsymbol{\theta}) = \mathcal{N}(y_1 \mid \theta_0 + \theta_1 x_1, \sigma^2)$$

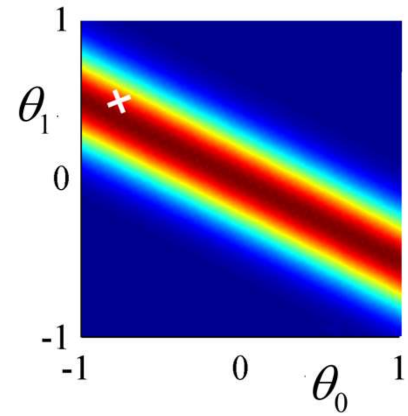
which implies

$$y_1 = \theta_0 + \theta_1 x_1 + \epsilon_1 \qquad \text{with } \epsilon_1 \sim \mathcal{N}(\epsilon_1 \mid 0, \sigma^2)$$

$$\theta_1 = -\frac{1}{x_1}\theta_0 + \frac{y_1}{x_1} - \frac{1}{x_1}\epsilon_1$$

- From the plot, we can observe that approximately

$$\theta_1 = -0.5 \cdot \theta_0 + 0$$



- Therefore $(x_1, y_1) = (2, 0)$ is the correct answer

Quiz: Levenshtein Distance

- We consider a modified version of the Levenshtein distance, which defines $d((x_1,\ldots,x_L),(y_1,\ldots,y_K))$ as the minimum cost required to transform $x_1,\ldots,x_L$ into $y_1,\ldots,y_K$ where
  - Insertions have a cost of one,
  - Deletions have a cost of one,
  - Substitutions have a cost of two.
- **Question**: which of the following modified recursion schemes computes this distance:

A) $$D(i,j)=\begin{cases} 2\cdot D(i-1,j-1): & \text{if } x_i=y_j \\ 1+\min\begin{cases} D(i-1,j) \\ D(i,j-1) \\ D(i-1,j-1) \end{cases} & \text{otherwise} \end{cases}$$

B) $$D(i,j)=\begin{cases} D(i-1,j-1): & \text{if } x_i=y_j \\ 1+\min\begin{cases} D(i-1,j) \\ D(i,j-1) \\ 1+D(i-1,j-1) \end{cases} & \text{otherwise} \end{cases}$$

C) $$D(i,j)=\begin{cases} D(i-1,j-1): & \text{if } x_i=y_j \\ 2+\min\begin{cases} D(i-1,j) \\ D(i,j-1) \\ D(i-1,j-1) \end{cases} & \text{otherwise} \end{cases}$$

D) $$D(i,j)=\begin{cases} D(i-1,j-1): & \text{if } x_i=y_j \\ 1+\min\begin{cases} 1+D(i-1,j) \\ D(i,j-1) \\ D(i-1,j-1) \end{cases} & \text{otherwise} \end{cases}$$

Quiz: Levenshtein Distance

- **Solution**: the correct recursion scheme is B)

- The standard recusion scheme says that to transform $x_1,...,x_i$ into $y_1,...,y_j$ if the last characters do not match we have three options:

$$D(i,j) = \begin{cases} D(i-1, j-1): & \text{if } x_i = y_j \\ 1 + \min \begin{cases} D(i-1, j) \\ D(i, j-1) \\ D(i-1, j-1) \end{cases} & \text{ot} \end{cases}$$

Delete last character from source string and then transform $x_1,...,x_{i-1}$ into $y_1,...,y_j$: cost stays at one

Insert a character at end of source string that matches and transform $x_1,...,x_i$ into $y_1,...,y_{j-1}$: cost stays at one

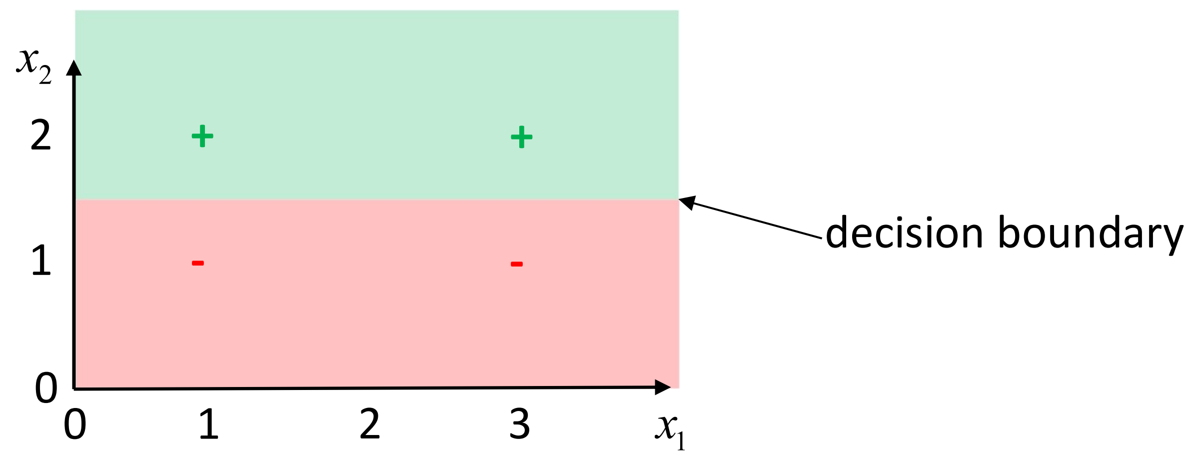
Substitute last character in source string such that it matches and then transform $x_1,...,x_{i-1}$ into $y_1,...,y_{j-1}$: here cost should be two

- In recursion scheme B), the cost for the last option is indeed two

Quiz: Nearest Neighbor

- We study a binary one-nearest-neighbor classifier $f_{\mathcal{D}} : \mathbb{R}^2 \to \{0,1\}$ on the data set

$$\mathcal{D} = \{((1,1),0),((3,1),0),((1,2),1),((3,2),1)\}$$



- **Question**: If we perform a leave-one-out cross-validation for estimating the error of our classifier on this data set, what is the error estimate?
  A) error estimate is 0% error
  B) error estimate is 25% error
  C) error estimate is 50% error
  D) error estimate is 75% error
  E) error estimate is 100% error

Quiz: Nearest Neighbor

- **Solution**: the error estimate is 100% error.

- In leave-one-out cross-validation, we always use three of the four instances for the training set, and one of the four instances as the test instance
- The test instance will always be misclassified, because the nearest instance to any positive instance is a negative instance and vice versa: