Maschinelles Lernen 1 - Assignment 2

Technische Universität Berlin
WS 2013/2014

Christoph Conrads (315565)
Benjamin Pietrowicz (332542)
Antje Relitz (327289)
Mitja Richter (324680)

25th October 2013

1 Exercise

$$
\begin{aligned}
f &= \text{a house will be flooded} & \neg f &= \text{a house won't be flooded} \\
x &= \text{house stands in a high risk area} & \neg x &= \text{house stands in a low risk area} \\
P(f) &= 0.0005 & P(\neg f) &= 0.9995 \\
P(x) &= 0.04 & P(\neg x) &= 0.96 \\
P(x|f) &= 0.8 & P(\neg x|f) &= 0.2 \\
\alpha_1 &= \text{buying insurance} & \alpha_2 &= \text{not buying insurance}
\end{aligned}
$$

We estimate the houses value is 210,000 €, since the text states that 100,000 € is less than half the value.

$$
\begin{aligned}
\lambda(\alpha_1|f) &= 111100€ & \lambda(\alpha_1|\neg f) &= 1100€ \\
\lambda(\alpha_2|f) &= 210000€ & \lambda(\alpha_1|\neg f) &= 0€
\end{aligned}
$$

(a) Bayes Rule:

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

Compute $P(f|x)$:

$$P(f|x) = \frac{P(x|f)P(f)}{P(x)} = \frac{0.8 \cdot 0.0005}{0.04} = 0.01$$

(b) Compute $R(\alpha_1|x)$ and $R(\alpha_2|x)$:

$$
\begin{aligned}
&P(\neg f|x) = 1 - P(f|x) = 0.99 \\
&R(\alpha_1|x) = \lambda(\alpha_1|f)P(f|x) + \lambda(\alpha_1|\neg f)P(\neg f|x) = 111100€ * 0.01 + 1100€ * 0.99 = 2200€ \\
&R(\alpha_2|x) = \lambda(\alpha_2|f)P(f|x) + \lambda(\alpha_2|\neg f)P(\neg f|x) = 210000€ * 0.01 + 0€ * 0.99 = 2100€
\end{aligned}
$$

Since $R(\alpha_2|x) < R(\alpha_1|x)$ not buying an insurance would be more viable.

2 Exercise

see attached documents

3 Exercise

(a) All classifiers can be used to fit the data perfectly, when we use the correct parameters. For example classifier A divides the data with parameter a = 0.5 perfectly. The problem is to generalize correctly and not to overfit the data. Since we have only a few datapoints we want to be as general as possible. That is why we have chosen classifier B to be the optimal classifier.

(b) see attached documents

(c) Computational aspects: Since the learn algorithm uses the linear classifier with a different parameter set, which is also linear, the learn algortihm itself is linear. The time it consumes is minimal because it stops as soon as the first solution has been found.

Model selection: In the case that more than one parameter perfectly classifies the data we would choose the parameter, for which the sum of the smallest distances of the datapoints of one class to the segregation line, is closest to the other classes smallest distance of the datapoints. Since we're using the classifier from 1.(b), which linearly segregates the two classes of datapoints, there is no danger of overfitting the data. Our algorithm however just chooses the first parameter that fits.