Naive Bayes Classifier in Python

Do they play tennis tomorrow?

Day	Outlook	Temperature	Humidity	Wind	Play Tennis
Day1	Sunny	Hot	High	Weak	*No*
Day2	Sunny	Hot	High	Strong	*No*
Day3	Overcast	Hot	High	Weak	Yes
Day4	Rain	Mild	High	Weak	Yes
Day5	Rain	Cool	Normal	Weak	Yes
Day6	Rain	Cool	Normal	Strong	*No*
Day7	Overcast	Cool	Normal	Strong	Yes
Day8	Sunny	Mild	High	Weak	*No*
Day9	Sunny	Cool	Normal	Weak	Yes
Day10	Rain	Mild	Normal	Weak	Yes
Day11	Sunny	Mild	Normal	Strong	Yes
Day12	Overcast	Mild	High	Strong	Yes
Day13	Overcast	Hot	Normal	Weak	Yes
Day14	Rain	Mild	High	Strong	*No*

For tomorrow, Day15 <sunny, cool, high, strong>, do they play tennis?

Bayes Theorem

- Given a hypothesis *h* and data *D* which bears on the hypothesis:

$$P(h \mid D) = \frac{P(D \mid h)P(h)}{P(D)}$$

- *P(h)*: independent probability of *h*: *prior probability*
- *P(D)*: independent probability of *D*
- *P(D|h)*: conditional probability of *D* given h: *likelihood*
- *P(h|D)*: conditional probability of *h* given *D*: *posterior probability*

Maximum A Posterior

- Based on Bayes Theorem, we can compute the *Maximum A Posterior* (MAP) hypothesis for the data
- We are interested in the best hypothesis for some space H given observed training data D.

$$
\begin{aligned}
h_{MAP} &\equiv \underset{h \in H}{\operatorname{argmax}}\, P(h \mid D) \\
&= \underset{h \in H}{\operatorname{argmax}}\, \frac{P(D \mid h) P(h)}{P(D)} \\
&= \underset{h \in H}{\operatorname{argmax}}\, P(D \mid h) P(h)
\end{aligned}
$$

H: set of all hypothesis.

Note that we can drop *P(D)* as the probability of the data is constant (and independent of the hypothesis).

Bayes Classifiers

Assumption: training set consists of instances of different classes described c_j as conjunctions of attributes values

Task: Classify a new instance *d* based on a tuple of attribute values into one of the classes $c_j \in C$

Key idea: assign the most probable class c_{MAP} using Bayes Theorem.

$$c_{MAP} = \underset{c_j \in C}{\operatorname{argmax}}\, P(c_j \mid x_1, x_2, \ldots, x_n)$$

$$= \underset{c_j \in C}{\operatorname{argmax}}\, \frac{P(x_1, x_2, \ldots, x_n \mid c_j) P(c_j)}{P(x_1, x_2, \ldots, x_n)}$$

$$= \underset{c_j \in C}{\operatorname{argmax}}\, P(x_1, x_2, \ldots, x_n \mid c_j) P(c_j)$$

Parameters estimation

- $P(c_j)$
    - Can be estimated from the frequency of classes in the training examples.
- $P(x_1, x_2, \ldots, x_n | c_j)$
    - $O(|X|^n \bullet |C|)$ parameters
    - Could only be estimated if a very, very large number of training examples was available.
- Independence Assumption: attribute values are conditionally independent given the target value: ***naïve*** *Bayes*.

$$P(x_1, x_2, \ldots, x_n \mid c_j) = \prod_i P(x_i \mid c_j)$$

$$c_{NB} = \operatorname*{arg\,max}_{c_j \in C} P(c_j) \prod_i P(x_i \mid c_j)$$

Do they play tennis tomorrow?

Day	Outlook	Temperature	Humidity	Wind	Play Tennis
Day1	Sunny	Hot	High	Weak	*No*
Day2	Sunny	Hot	High	Strong	*No*
Day3	Overcast	Hot	High	Weak	Yes
Day4	Rain	Mild	High	Weak	Yes
Day5	Rain	Cool	Normal	Weak	Yes
Day6	Rain	Cool	Normal	Strong	*No*
Day7	Overcast	Cool	Normal	Strong	Yes
Day8	Sunny	Mild	High	Weak	*No*
Day9	Sunny	Cool	Normal	Weak	Yes
Day10	Rain	Mild	Normal	Weak	Yes
Day11	Sunny	Mild	Normal	Strong	Yes
Day12	Overcast	Mild	High	Strong	Yes
Day13	Overcast	Hot	Normal	Weak	Yes
Day14	Rain	Mild	High	Strong	*No*

For tomorrow, Day15 <sunny, cool, high, strong>, do they play tennis?

Based on the examples in the table, classify the following datum ***x***:

x=(Outl=Sunny, Temp=Cool, Hum=High, Wind=strong)

- That means: Play tennis or not?

$$h_{NB} = \arg\max_{h\in[yes,no]} P(h)P(\mathbf{x}\mid h) = \arg\max_{h\in[yes,no]} P(h)\prod_t P(a_t \mid h)$$

$$= \arg\max_{h\in[yes,no]} P(h)P(Outlook = sunny \mid h)P(Temp = cool \mid h)P(Humidity = high \mid h)P(Wind = strong \mid h)$$

- Working:

$$P(PlayTennis = yes) = 9/14 = 0.64$$

$$P(PlayTennis = no) = 5/14 = 0.36$$

$$P(Wind = strong \mid PlayTennis = yes) = 3/9 = 0.33$$

$$P(Wind = strong \mid PlayTennis = no) = 3/5 = 0.60$$

$$etc.$$

$$P(yes)P(sunny \mid yes)P(cool \mid yes)P(high \mid yes)P(strong \mid yes) = 0.0053$$

$$P(no)P(sunny \mid no)P(cool \mid no)P(high \mid no)P(strong \mid no) = \mathbf{0.0206}$$

$$\Rightarrow answer : PlayTennis(x) = no$$

Underflow Prevention

- Multiplying lots of probabilities, which are between 0 and 1 by definition, can result in floating-point underflow.
- Since *log(xy) = log(x) + log(y),* it is better to perform all computations by summing logs of probabilities rather than multiplying probabilities.
- Class with highest final un-normalized log probability score is still the most probable.

$$c_{NB} = \underset{c_j \in C}{\operatorname{argmax}} \log P(c_j) + \sum_{i \in positions} \log P(x_i \mid c_j)$$