6CCS3ML1 (Machine Learning)

Tutorial 03 — Answers

(Version 1.1)

1. (a) For the priors, we look for the proportion of cases in which we have $Tennis = yes$ and $Tennis = no$:

$$\begin{aligned} p(Tennis = yes) &= \frac{\text{Number of cases where } Tennis = yes}{\text{Number of cases}} \\ &= \frac{9}{14} \\ &= 0.642 \end{aligned}$$

(b) Now, for $p(Outlook = rain|Tennis = yes)$ we look for situations in which $Tennis = yes?$ and count the number where $Outlook = rain$:

$$\begin{aligned} p(Outlook = rain|Tennis = yes) &= \frac{\text{Cases where } Outlook = rain \text{ and } Tennis = yes}{\text{Cases where} Tennis = yes} \\ &= \frac{3}{9} \\ &= 0.33 \end{aligned}$$

2. We need to compute the probability of playing tennis when:

$$\mathcal{D} = \{Outlook = Rain, Temp = Hot, Humidity = Normal, Wind = Weak\}$$

so we want:

$$p(Tennis = yes|\mathcal{D})$$

and from bayes rule we know that (writing $Tennis = yes$ as $tennis$):

$$p(tennis|\mathcal{D}) = \frac{p(\mathcal{D}|tennis)p(tennis)}{p(\mathcal{D})}$$

and now, writing $Outlook = Rain$ as $rain$, $Temp = Hot$ has hot, $Humidity = Normal$ as $normal$ and $Wind = Weak$ as $weak$, we have:

$$p(tennis|rain, hot, normal, weak) = \frac{p(rain, hot, normal, weak|tennis)p(tennis)}{p(rain, hot, normal, weak)}$$

Because we are using a Naive Bayes model, we assume that:

$$p(rain, hot, normal, weak|tennis) = p(rain|tennis).p(hot|tennis).p(normal|tennis)p(weak|tennis)$$

and from the data we can compute that:

$$\begin{aligned} p(rain|tennis) &= \frac{3}{9} \\ p(hot|tennis) &= \frac{2}{9} \\ p(normal|tennis) &= \frac{6}{9} \\ p(weak|tennis) &= \frac{6}{9} \end{aligned}$$

So:

$$\begin{aligned}
p(rain, hot, normal, weak|tennis) &= p(rain|tennis)\, p(hot|tennis)\, p(normal|tennis)\, p(weak|tennis) \\
&= \frac{3}{9} \cdot \frac{2}{9} \cdot \frac{6}{9} \cdot \frac{6}{9} \\
&= 0.0329
\end{aligned}$$

and:

$$\begin{aligned}
p(tennis|rain, hot, normal, weak) &= p(tennis) \cdot 0.0329 \\
&= \frac{9}{14} \cdot 0.0329 \\
&= 0.021
\end{aligned}$$

Similarly, Bayes rule tells us that:

$$p(\neg tennis|rain, hot, normal, weak) = \frac{p(rain, hot, normal, weak|\neg tennis)p(\neg tennis)}{p(rain, hot, normal, weak)}$$

where we write $Tennis = no$ as $\neg tennis$. Again we use Naive Bayes to get:

$$\begin{aligned}
p(rain, hot, normal, weak|\neg tennis) &= p(rain|\neg tennis) \cdot p(hot|\neg tennis) \cdot \\
&\quad p(normal|\neg tennis) \cdot p(weak|\neg tennis) \\
&= \frac{2}{5} \cdot \frac{2}{5} \cdot \frac{1}{5} \cdot \frac{2}{5} \\
&= 0.0128
\end{aligned}$$

and so:

$$\begin{aligned}
p(\neg tennis|rain, hot, normal, weak) &= p(\neg tennis) \cdot 0.0128 \\
&= \frac{5}{14} \cdot 0.0128 \\
&= 0.00457
\end{aligned}$$

Now, because a probability distribution has to sum to 1, we know that:

$$p(tennis|rain, hot, normal, weak) + p(\neg tennis|rain, hot, normal, weak)$$

and so we compute:

$$\begin{aligned}
p(tennis|rain, hot, normal, weak) &= 0.021 \cdot \frac{1}{0.021 + 0.00457} \\
&= 0.021 \cdot 39 \\
&= 0.821
\end{aligned}$$

3. In this question we are asked a variation of the previous question, which is what v_{NB} is under the conditions:

$$\mathcal{D} = \{Outlook = Rain, Temp = Hot, Humidity = Normal, Wind = Weak\}$$

From the slides, we know that this is:

$$v_{NB} = \arg \max_{Tennis=tennis,\neg tennis} p(Tennis) \cdot p(rain|\neg Tennis) \cdot p(hot|\neg Tennis) \cdot \\ p(normal|\neg Tennis) \cdot p(weak|\neg Tennis)$$

In other words, which value of $Tennis$ maximises:

$$p(Tennis) \cdot p(rain|Tennis) \cdot p(hot|Tennis) \cdot p(normal|Tennis) \cdot p(weak|Tennis)$$

Since we already computed:

$$p(tennis) \cdot p(rain|tennis) \cdot p(hot|tennis) \cdot p(normal|tennis) \cdot p(weak|tennis) = 0.021$$
$$p(\neg tennis) \cdot p(rain|\neg tennis) \cdot p(hot|\neg tennis) \cdot p(normal|\neg tennis) \cdot p(weak|\neg tennis) = 0.00457$$

it is clear that v_{NB} is $Tennis = yes$, meaning that the naive Bayes classifier predicts that tennis will happen.

4. For this question we need to compute v_{NB} (again) because this is the prediction made by the model. As the slides tell us, the calculation we need to do is:

$$v_{NB} = \arg \max_{v_i \in \{tennis,\neg tennis\}} p(v_i) \prod_j p(condition_j|v_i)$$

where

$$condition_j \in \{cloud, hot, normal, weak\}$$

which means we look at:

$$p(tennis) \prod_{condition_j \in \{cloud,hot,normal,weak\}} p(condition|tennis)$$

and

$$p(\neg tennis) \prod_{condition_j \in \{cloud,hot,normal,weak\}} p(condition|\neg tennis)$$

and pick which of $tennis$ and $\neg tennis$ gives the larger number.

The previous question gives us most of the conditional probabilities we need for these calculations, but we don't have the ones for $cloud$. Now, for $p(cloud|tennis)$ we look for situations in which $Tennis = yes?$ and count the number where $Outlook = cloud$:

$$\begin{aligned} p(Outlook = cloud|Tennis = yes) &= \frac{4}{9} \\ &= 0.44 \end{aligned}$$

When we do the same for $p(Outlook = cloud|Tennis = no)$, we find:

$$\begin{aligned} p(cloud|\neg tennis) &= \frac{0}{5} \\ &= 0 \end{aligned}$$

This, of course, is a case of our maximum likelihood estimates overfitting. So we need a better solution. One possibility would be to the m-estimate approach. Here, we compute:

$$p(A = a|B = b) = \frac{n_c + m \cdot p}{n + m}$$

where:

- n is the total number of cases where $B = b$
- n_c is the number of those cases where $A = a$.
- m is the equivalent sample size
- p is the prior estimate.

There is no right way to apply this, but reasonable instantiations of m and p are as follows:

- $m = 5$, since this is the same as the number of samples we have. Bigger would put more weight on the prior.
- $p = \frac{1}{3}$ since we have three values for $Outlook$ it assumes that without any information, all the values of $Outlook$ are equally likely for $\neg tennis$

With these numbers we have:

$$\begin{aligned} p(cloud|\neg tennis) &= \frac{0 + \frac{1}{3} \cdot 5}{5 + 5} \\ &= 0.166 \end{aligned}$$

We then do the calculations:

$$\begin{aligned} v_{nb}(tennis) &= p(tennis) \prod_{condition_j \in \{cloud, hot, normal, weak\}} p(condition|tennis) \\ &= p(tennis) \cdot p(cloud|tennis) \cdot p(hot|tennis) \cdot p(normal|tennis) \cdot p(weak|tennis) \\ &= \frac{9}{14} \cdot \frac{4}{9} \cdot \frac{2}{9} \cdot \frac{6}{9} \cdot \frac{6}{9} \\ &= 0.282 \end{aligned}$$

and

$$\begin{aligned} v_{nb}(\neg tennis) &= p(\neg tennis) \prod_{condition_j \in \{cloud, hot, normal, weak\}} p(condition|\neg tennis) \\ &= p(\neg tennis) \cdot p(cloud|\neg tennis) \cdot p(hot|\neg tennis) \cdot p(normal|\neg tennis) \cdot p(weak|\neg tennis) \\ &= \frac{5}{14} \cdot 0.166 \cdot \frac{2}{5} \cdot \frac{1}{5} \cdot \frac{2}{5} \\ &= 0.00189 \end{aligned}$$

and v_{NB} is $tennis$.

as sanity check on the value we get for $v_{nb}(\neg tennis)$, let's compre it with two things. First, with what we would have got without the m-estimate. In this case $v_{nb}(\neg tennis)$ would have been zero, because $p(cloud|\neg tennis)$ would have been zero, and so would been smaller than what we calculated. Second, let's compare it with the value of $v_{nb}(\neg tennis)$ under condition $Outlook = Rain$, which was the case in Q2. Here $v_{nb}(\neg tennis)$ had value 0.00457, larger than what we computed. These two values bracket what we got, which is what we would expect. Taking $p(cloud|\neg tennis)$ to be zero is probably an underestimate, so should give us a result that is less than what we computed using the m-estimate. $Outlook = Rain$ has more examples in the dataset, and so we would expect it give us a somewhat larger value for $v_{nb}(\neg tennis)$ than we got for $Oulook = cloud$, and that is the case.

5. From the definition of a mixture model, we have:

$$p(x) = 0.4\mathcal{N}_1(x) + 0.6\mathcal{N}_2(x)$$

Now:

$$\begin{aligned}\mathcal{N}_1(x) &= e^{\frac{-1}{2\times 5}(6-3)^2}\\ &= e^{\frac{-1}{10}3^2}\\ &= e^{\frac{-9}{10}}\\ &= 0.406\end{aligned}$$

and

$$\begin{aligned}\mathcal{N}_2(x) &= e^{\frac{-1}{2\times 2}(6-4)^2}\\ &= e^{\frac{-1}{4}2^2}\\ &= e^{\frac{-4}{4}}\\ &= 0.368\end{aligned}$$

Thus:

$$\begin{aligned}p(x) &= 0.4 \times 0.406 + 0.6 \times 0.368\\ &= 0.383\end{aligned}$$

6. With the starting cluster centres $C_1 = (7, 5)$, $C_2 = (9, 7)$ and $C_3 = (9, 1)$, we can compute the distance to each centre:

Instance	Attributes		Distance to C_1	Distance to C_2	Distance to C_3	Closest
	x_1	x_2				
X_1	5	8	5	5	11	C_1
X_2	6	7	3	3	9	C_2
X_3	6	4	2	6	6	C_1
X_4	5	7	4	4	10	C_2
X_5	5	5	2	6	8	C_1
X_6	6	5	1	5	7	C_1
X_7	1	7	8	8	14	C_2
X_8	7	5	0	4	6	C_1
X_9	6	5	1	5	7	C_1
X_{10}	6	7	3	3	9	C_2

Where two points are the same distance from a cluster centre, ties are broken randomly.

Since C_3 has no points assigned to it, a new centre is chosen randomly, $(6, 4)$, and the other cluster centres are reset by averging the points in the cluster: $C_1 = (5.83, 5.33)$ and $C_2 = (4.5, 7)$.

We then repeat the process, giving:

Instance	Attributes		Distance to C_1	Distance to C_2	Distance to C_3	Closest
	x_1	x_2				
X_1	5	8	3.5	1.5	5	C_2
X_2	6	7	1.83	1.5	3	C_2
X_3	6	4	1.5	4.5	0	C_3
X_4	5	7	2.5	0.5	4	C_2
X_5	5	5	1.16	2.5	2	C_1
X_6	6	5	0.5	3.5	1	C_1
X_7	1	7	6.5	3.5	8	C_2
X_8	7	5	1.5	4.5	2	C_1
X_9	6	5	0.5	3.5	1	C_1
X_{10}	6	7	1.83	1.5	3	C_2

Aside from the point claimed by C_3, only one other point changed cluster, suggesting the algorithm is nearly done.

We have the new cluster centres $C_1 = (6, 5)$ and $C_2 = (4.6, 7.2)$ while C_3 does not change since it coincides with the only point in the cluster.

Another iteration gives:

Instance	Attributes		Distance to C_1	Distance to C_2	Distance to C_3	Closest
	x_1	x_2				
X_1	5	8	4	1.2	5	C_2
X_2	6	7	2	1.6	3	C_2
X_3	6	4	1	4.6	0	C_3
X_4	5	7	3	0.6	4	C_2
X_5	5	5	1	2.6	2	C_1
X_6	6	5	0	3.6	1	C_1
X_7	1	7	7	3.8	8	C_2
X_8	7	5	1	4.6	2	C_1
X_9	6	5	0	3.6	1	C_1
X_{10}	6	7	2	1.6	3	C_2

No points change cluster, so the cluster centres remain the same, and the clusters have converged.

7. One way to view the K-means algorithm as applied in the previous question is as a version of the EM algorithm applied to clustering, where the algorithm determines the mean values of two dimensional Gaussians. In a mixture of Gaussians, the likelihood that a given data point was generated by a specific Gaussian is (as we saw in Question 5) related to the distance of the point from the mean of the distribution — a point that is closer to mean A than mean B is more likely to come from that distribution. In exactly the same way, the points associated with a given cluster at the end of a run of K-means are the ones that are closest to the centre of that cluster (the ones that would be most likely to come from the Gaussian whose mean is the cluster centre).

Version list

- Version 1.0, January 18th 2020.
- Version 1.1, January 11th 2021.