Chapter 9
Bayesian Learning

Bayesian learning [Tipping, 2004, Barber, 2012] is the name commonly used to identify a set of computational methods for supervised learning based on Bayes' Theorem. Broadly speaking, Bayes' Theorem deals with the modification of our perception of the probability of an event, as a consequence of the occurrence of one or more facts. For instance, what probability are you assigning to the event "somebody stole my car" at the moment? Of course, this can depend on many different factors, but on a normal day, one may argue that that probability is generally rather low. Now, imagine that you go looking for your car, and the car is not in the place where you remember that you parked it. What is now the probability of the event "somebody stole my car"? The fact that the car is not where it was parked clearly changes the probability that it was stolen. This property is general: the realization of some events can modify the probability of others. This property can be exploited to tackle Machine Learning tasks, for instance classification: data, interpreted as events, can be used to change the probability that a given observation belongs to a given class. Before studying this mechanism in detail, let us first present Bayes' Theorem and its most immediate use in Machine Learning.

9.1 Bayes' Theorem and Machine Learning

Let A and B be two events, and $P(A)$ and $P(B)$ their respective probabilities (sometimes called marginal probabilities). Also, let $P(A|B)$ be the conditional probability of event A, or likelihood of event A, knowing that B is true. Analogously, let $P(B|A)$ be the probability of event B being true given that event A is true. Bayes' Theorem can be enunciated as follows.

Theorem 9.1. *(Bayes' Theorem)*

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

L. Vanneschi and S. Silva, *Lectures on Intelligent Systems*, Natural Computing Series,
https://doi.org/10.1007/978-3-031-17922-8_9

Proof. Directly from the definition of conditional probability, we have:

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

where $P(A,B)$ is the probability of both events A and B happening (in other references sometimes denoted by $P(A \cap B)$). From the previous equation, we can write:

$$P(A,B) = P(A|B)\ P(B)$$

However, given that $P(A,B) = P(B,A)$, we have:

$$P(A|B)\ P(B) = P(B|A)\ P(A)$$

From which follows the thesis immediately.

□

The formula might look complicated, but it is actually quite user friendly: all we need to do is plug three ingredients into the formula, and this allows us to have an updated probability, based on new information. $P(A|B)$ is often called the *posterior probability*, in the sense that it quantifies the "new" probability of A, after we have received the information that B has happened.

As puzzling as it may seem at first sight, this simple theorem of probability calculus can be very important for Machine Learning. In fact, let us assume, for instance, that we want to tackle a classification task. In other terms, we have a dataset D, and we receive a new (unseen) observation, on which we need to predict the class label. Let us assume, for simplicity, that the task is binary classification, i.e., classification into two possible classes C_1 and C_2. Then, we have two different possible hypotheses:

- The new observation belongs to class C_1 (let us call this hypothesis h_1);
- The new observation belongs to class C_2 (let us call this hypothesis h_2).

We can use Bayes' Theorem to make this classification. Remember that we have some facts that we can observe, and it is based on those facts that we do the classification. Now, let us ask to ourselves, what is the information (i.e., the facts!) that we normally use to generate a classifier? The answer is straightforward: the data! All the information we need is in D, which we consider the training set. So, all we have to do is to calculate the probability that h_1 is true after having observed D, and the probability that h_2 is true after having observed D. If the former is larger than the latter, then the new observation will be categorized into class C_1, otherwise into class C_2. Applying Bayes' Theorem directly, we have:

$$P(h_1|D) = \frac{P(D|h_1) \cdot P(h_1)}{P(D)}, \qquad P(h_2|D) = \frac{P(D|h_2) \cdot P(h_2)}{P(D)}$$

Remark that we just have to compare $P(h_1|D)$ to $P(h_2|D)$. In order to understand which one of the two is larger, we do not have to calculate them exactly. To make

this comparison, we can observe that the quantity at the denominator, i.e., $P(D)$, is the same in both formulas, and thus it does not have any influence on the comparison. So, we can simply "ignore it". So, in practice, what we have to do is just to calculate $P(D|h_1) \cdot P(h_1)$ and $P(D|h_2) \cdot P(h_2)$. If the former is larger than the latter, then the new observation will be categorized into class C_1, otherwise into class C_2.

Let us give the name *maximum a posteriori* hypothesis h_{MAP} to the hypothesis that has the maximum probability, among h_1 and h_2, *after* having observed our data D (i.e., "*a posteriori*"), and let H be the space of all possible hypotheses (in our case: $H = \{h_1, h_2\}$). We have:

$$h_{MAP} = \underset{h \in H}{\textbf{argmax}}\ P(D|h) \cdot P(h)$$

Let us see how this result can be calculated in practice, using a simple example.

Example 9.1. Let us consider a medical diagnosis problem, in which there are only two alternative hypotheses; given a particular patient and a particular disease:

- the patient has the disease;
- the patient does not have the disease.

The available data is from a particular laboratory test, with two possible outcomes: *positive* or *negative*. We have prior knowledge that, over the entire population of people, only 0.8% have this disease. Furthermore, we know that the lab test is an imperfect indicator of the disease: the test returns a correct positive result in 98% of the cases in which the disease is actually present, and a correct negative result in 97% of the cases in which the disease is actually not present. We can summarize the situation like this:

$P(\text{disease}) = 0.008$	$P(\text{not disease}) = 0.992$
$P(\text{positive test} \mid \text{disease}) = 0.98$	$P(\text{negative test} \mid \text{disease}) = 0.02$
$P(\text{positive test} \mid \text{not disease}) = 0.03$	$P(\text{negative test} \mid \text{not disease}) = 0.97$

Suppose we now observe a new patient, for whom the lab test returns a positive result. Should we diagnose the patient as having the disease?

According to the previous discussion, we have to understand whether $P(\text{disease} \mid \text{positive test})$ is larger than $P(\text{not disease} \mid \text{positive test})$ or the other way around. Applying Bayes' Theorem, we have to respectively calculate $P(\text{positive test} \mid \text{disease}) \cdot P(\text{disease})$ and $P(\text{positive test} \mid \text{not disease}) \cdot P(\text{not disease})$ and see which one of the two is the largest. We have all the quantities we need, so we can do it:

$$\begin{aligned} P(\text{disease} \mid \text{positive test}) &= P(\text{positive test} \mid \text{disease}) \cdot P(\text{disease}) \\ &= 0.98 \cdot 0.008 = 0.0078 \end{aligned}$$

$$\begin{aligned} P(\text{not disease} \mid \text{positive test}) &= P(\text{positive test} \mid \text{not disease}) \cdot P(\text{not disease}) \\ &= 0.03 \cdot 0.992 = 0.0298 \end{aligned}$$

So, the maximum *a posteriori* hypothesis h_{MAP} is:

$$h_{MAP} = \text{not disease}$$

A bit surprisingly, we must conclude that, even though the lab test was positive, the most probable hypothesis is still that the patient does *not* have the disease. It is important to remark that in Bayesian inference, the hypotheses are not completely accepted or rejected, but rather become more or less probable as more data is observed.

9.2 Naïve Bayes

Naïve Bayes is the simplest classifier based on Bayesian inference. It uses a *very* restrictive (and often false!) hypothesis: that all the variables in the dataset are independent of each other. Despite its simplicity and this restrictive hypothesis, Naïve Bayes has been shown to often have a very good classification performance; in some domains, even comparable with or better than that of Artificial Neural Networks (discussed in Chapter 7) and Decision Trees (discussed in Chapter 6).

Let us assume that we have a classification dataset with n variables (features), and let us assume that we receive a new observation, which we want to classify. Of course, the observation will have the form:

$$a_1,\ a_2,\ ...,\ a_n$$

i.e., n values, one for each variable. The space of our hypotheses H is now characterized by k hypotheses, where k is the number of classes in our dataset/problem:

- the new observation belongs to class C_1;
- the new observation belongs to class C_2;
- ...
- the new observation belongs to class C_k;

Analogously to the previous example, of all the hypotheses in the space H, we have to find the most probable, the maximum *a posteriori* hypothesis h_{MAP}. In other words, we want to find:

$$h_{MAP} = \underset{h \in H}{\textbf{argmax}}\, P(h|a_1, a_2, ..., a_n)$$

Applying Bayes' Theorem:

$$P(h|a_1, a_2, ..., a_n) = \frac{P(a_1, a_2, ..., a_n|h) \cdot P(h)}{P(a_1, a_2, ..., a_n)}$$

As previously remarked, for all hypotheses $h \in H$ the denominator $P(a_1, a_2, ..., a_n)$ is the same. So, this denominator is irrelevant for deciding what is the maximum *a posteriori* hypothesis, and it can be removed. So, we can conclude that:

$$h_{MAP} = \underset{h \in H}{\textbf{argmax}} P(a_1, a_2, ..., a_n|h) \cdot P(h)$$

where:

- $P(h)$ is easy to estimate for the various $h \in H$. We simply have to count the frequency with which the target value h occurs in the training data. In other words, for each class, we have to count the number of observations labeled with that class and divide it by the total number of observations.

- $P(a_1, a_2, ..., a_n|h)$ is instead very hard to estimate for the different $h \in H$ (even its intuitive meaning is arguably very hard to understand). But we can easily transform it into something that is very easy to calculate, making the very restrictive assumption, typical of the Naïve Bayes method, that the attribute values are conditionally independent, given the target value.

So, now our objective is to transform the term $P(a_1, a_2, ..., a_n|h)$ into an expression that is easy to calculate using the dataset. If we make the previously mentioned assumption of independence of the attribute values, we can write:

$$P(a_1, a_2, ..., a_n|h) = P(a_1|h) \cdot P(a_2|h) \cdot ... \cdot P(a_n|h) = \prod_{i=1}^{n} P(a_i|h)$$

For each $i = 1, 2, ..., n$ and for each $h \in H$, $P(a_i|h)$ can be easily calculated. Simply, for each class, we have to count the proportion of observations labeled with that class in which the *ith* attribute has exactly the value a_i. The next example should clarify.

Example 9.2. Let us recall the dataset reported in Table 6.1, on page 149, whose objective was to categorize days into two classes, to predict whether a given person would play tennis or not. The dataset is repeated here for the sake of convenience:

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Let us now assume that we want to classify the following new (unseen) observation:

$$\alpha = <\text{Outlook} = \text{sunny}, \text{Temperature} = \text{cool}, \text{Humidity} = \text{high}, \text{Wind} = \text{strong}>$$

Of course, the space of hypotheses H is:

- h_1: observation α belongs to class "Yes";
- h_2: observation α belongs to class "No";

Following the previous reasoning, in order to find the maximum *a posteriori* hypothesis h_{MAP}, we have to understand which one of the following two quantities is the bigger one:

- $P(\text{Outlook} = \text{sunny}, \text{Temperature} = \text{cool}, \text{Humidity} = \text{high}, \text{Wind} = \text{strong} \mid \text{class} = \text{Yes}) \cdot P(\text{class} = \text{Yes})$
- $P(\text{Outlook} = \text{sunny}, \text{Temperature} = \text{cool}, \text{Humidity} = \text{high}, \text{Wind} = \text{strong} \mid \text{class} = \text{No}) \cdot P(\text{class} = \text{No})$

If we accept the hypothesis of independence of the single variables (which is clearly not true in this example), this is equivalent to understanding which one of the following quantities is the bigger:

- $h_1 \rightarrow P(\text{Outlook} = \text{sunny} \mid \text{class} = \text{Yes}) \cdot P(\text{Temperature} = \text{cool} \mid \text{class} = \text{Yes}) \cdot P(\text{Humidity} = \text{high} \mid \text{class} = \text{Yes}) \cdot P(\text{Wind} = \text{strong} \mid \text{class} = \text{Yes}) \cdot P(\text{class} = \text{Yes})$

- $h_2 \rightarrow P(\text{Outlook} = \text{sunny} \mid \text{class} = \text{No}) \cdot P(\text{Temperature} = \text{cool} \mid \text{class} = \text{No}) \cdot P(\text{Humidity} = \text{high} \mid \text{class} = \text{No}) \cdot P(\text{Wind} = \text{strong} \mid \text{class} = \text{No}) \cdot P(\text{class} = \text{No})$

The good news is that all the quantities above can be calculated using only the training set. Starting by saying that, in general, a computer will be able to calculate all these quantities incomparably faster than we can do it manually, let us make the effort of calculating all these quantities manually in this example, so that it

can be completely clear what each one of these quantities exactly represents. In the following pointed list, for each one of the quantities, we describe how we can calculate it using the training set, and then we give the result.

- $P(\text{class = Yes}) = \frac{\text{\#observations in which class is Yes}}{\text{Total \#observations}} = 9/14 \approx 0.64$
- $P(\text{class = No}) = \frac{\text{\#observations in which class is No}}{\text{Total \#observations}} = 5/14 \approx 0.36$
- $P(\text{Outlook = sunny} \mid \text{class = Yes}) = \frac{\text{\#observations where class is Yes and Outlook is sunny}}{\text{Total \#observations where class is Yes}} = 2/9 \approx 0.22$
- $P(\text{Temperature = cool} \mid \text{class = Yes}) = \frac{\text{\#observations where class is Yes and Temperature is cool}}{\text{Total \#observations where class is Yes}} = 3/9 \approx 0.33$
- $P(\text{Humidity = high} \mid \text{class = Yes}) = \frac{\text{\#observations where class is Yes and Humidity is high}}{\text{Total \#observations where class is Yes}} = 3/9 \approx 0.33$
- $P(\text{Wind = strong} \mid \text{class = Yes}) = \frac{\text{\#observations where class is Yes and Wind is strong}}{\text{Total \#observations where class is Yes}} = 3/9 \approx 0.33$
- $P(\text{Outlook = sunny} \mid \text{class = No}) = \frac{\text{\#observations where class is No and Outlook is sunny}}{\text{Total \#observations where class is No}} = 3/5 = 0.6$
- $P(\text{Temperature = cool} \mid \text{class = No}) = \frac{\text{\#observations where class is No and Temperature is cool}}{\text{Total \#observations where class is No}} = 1/5 = 0.2$
- $P(\text{Humidity = high} \mid \text{class = No}) = \frac{\text{\#observations where class is No and Humidity is high}}{\text{Total \#observations where class is No}} = 4/5 \approx 0.8$
- $P(\text{Wind = strong} \mid \text{class = No}) = \frac{\text{\#observations where class is No and Wind is strong}}{\text{Total \#observations where class is No}} = 3/5 \approx 0.6$

We are now able to calculate the *a posteriori* probability of the two hypotheses:

- $h_1 \rightarrow P(\text{class = Yes} \mid \text{Outlook = sunny, Temperature = cool, Humidity = high, Wind = strong}) = 0.64 \cdot 0.22 \cdot 0.33 \cdot 0.33 \cdot 0.33 \approx 0.005$
- $h_2 \rightarrow P(\text{class = No} \mid \text{Outlook = sunny, Temperature = cool, Humidity = high, Wind = strong}) = 0.36 \cdot 0.6 \cdot 0.2 \cdot 0.8 \cdot 0.6 \approx 0.02$

The *a posteriori* probability of event h_2 is larger than that of event h_1. The conclusion is that we classify (unseen) instance α into class "No", meaning that our prediction is that the person will *not* play tennis on that day.

In [Domingos and Pazzani, 1996], Domingos and Pazzani show that the prediction made by Naïve Bayes can be optimal even if the attributes are not independent of each other, thus highlighting the excellent performance often obtained by Naïve Bayes, despite its simplicity.

9.3 Beyond Naïve Bayes: Hints on Bayesian Networks

Despite the excellent performance reported in the literature for many real-life applications, and despite the findings in [Domingos and Pazzani, 1996], which mitigate the negative effects of the hypothesis of independence of the variables, it is undeniable that this hypothesis can represent a limitation for Naïve Bayes in some scenarios, particularly when complex relationships of interdependency between variables are a fact. In those situations, more sophisticated paradigms, able to capture

variable interdependencies, may be needed. One of these formalisms is *Bayesian Networks* (BNs). A BN is a graph that allows us to represent and reason about an uncertain domain. The vertices (nodes) in BNs represent a set of random variables, $X = X_1, ..., X_i, ..., X_n$, from the domain, for instance the variables in a dataset. The edges (links) connecting pairs of nodes, $X_i \rightarrow X_j$, represent the direct dependencies between variables. The only constraint on the structure allowed for a BN is that there must not be any directed cycles: you cannot return to a node simply by following directed edges. So, BNs are directed acyclic graphs.

One first observation that could be made is that Naïve Bayes is also a (particular) BN. The characteristic of a graph representing Naïve Bayes is that it only has one edge for each input variable, joining the variable itself to the target variable. For instance, a BN modeling a problem like the one discussed in Example 9.2 could be represented as in Figure 9.1. The interpretation is that, in the model used for the

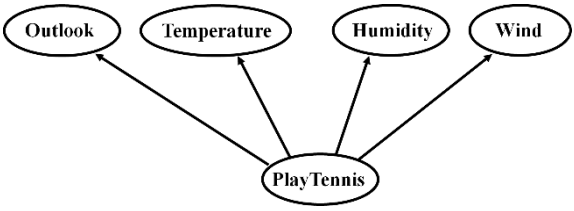

Fig. 9.1 The structure of the "naïve" Bayesian network of Example 9.2

problem of Example 9.2, the target variable PlayTennis is (of course) dependent on all the input variables[1] and there is no interdependence between any of the other variables. So, BNs are a formalism that also includes Naïve Bayes as a particular(ly simple) case.

Let us now consider a problem that can be modeled with a BN of a different shape compared to the one shown in Figure 9.1. This problem will be characterized by known variable interdependencies, and thus it will not be appropriate to model it using Naïve Bayes. A deliberately simple example is discussed.

Example 9.3. Let us study two events that can cause grass to be wet: the fact that a sprinkler is active and the fact that it is raining. Both these effects can cause grass to be wet, but still they are not mutually independent: in fact, one may reasonably imagine that when it rains, the sprinkler is not active. So, the fact of a sprinkler being active or not depends on whether it is raining or not. This situation can be modeled with the Bayesian network shown in Figure 9.2. Three Boolean variables are present in this BN: Wet Grass, Sprinkler and Rain. Variable Sprinker is dependent on variable Rain, while variable Wet Grass is dependent on both Sprinkler and Rain. A graph like the one in Figure 9.2 represents only a part of the definition of a BN, that is its *topology* (i.e., the way its vertices are connected by edges). For

[1] An effective feature selection algorithm should, in fact, remove possible variables on which the target does not depend, in the preprocessing phase.

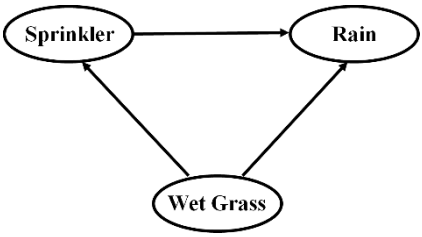

Fig. 9.2 The topology of the Bayesian network used in Example 9.3

the definition to be complete, the relationships between connected nodes have to be quantified. This is done by specifying a conditional probability distribution for each node, and, for the case of discrete variables as in this example, this can be done by means of conditional probability tables (CPTs). For instance, let us assume that the vertex representing variable Sprinkler is associated with the following CPT:

Rain	True	False
False	0.4	0.6
True	0.01	0.99

This table expresses the probability of variable Sprinkler being true or false, depending on the possible values of the variable Rain, on which it depends. The interpretation of this CPT is:

- If it is not raining (Rain = False), then the probability of the sprinkler being active (Sprinkler = True) is 0.4 and inactive (Sprinkler=False) is 0.6;
- If it is raining (Rain = True), then the probability of the sprinkler being active (Sprinkler = True) is 0.01 and inactive (Sprinkler=False) is 0.99.

Notice that, in general, a CPT is associated to *each* variable in the system, even if the variable does not depend on any other. In that case, the CPT simply contains the *a priori* probabilities for that variable. This is the case, in this example, for variable Rain. Its CPT is supposed to contain the probability of raining or not raining for an average day. So, the CPT of variable Rain may look like the following table:

True	False
0.2	0.8

Last but not least, the CPT of variable Wet Grass contains an entry for each possible combination of the variables it depends on. Let us assume that the CPT of variable Wet Grass, in our case, is:

Sprinkler	Rain	True	False
False	False	0.0	1.0
False	True	0.8	0.2
True	False	0.9	0.1
True	True	0.99	0.01

As it is easy to understand, for each variable, the larger the number of variables it depends on, the bigger the CPT. The BN can now be completely defined, as in Figure 9.3. Figure 9.3 is identical to Figure 9.2, except that in Figure 9.3 the vertices

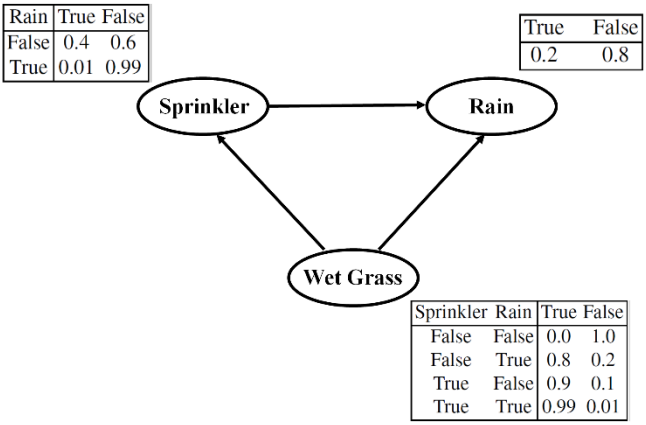

Fig. 9.3 The complete definition of the Bayesian network used in Example 9.3

have been annotated with the CPT of the corresponding variable. It should not be difficult to convince oneself that, usually, the CPTs are part of (or can be extracted from) the definition of the problem. For instance, if we have a supervised dataset, once the topology of the network has been defined (and this can be done manually, or by means of specific algorithms, as we will discuss in the continuation), the CPTs can usually be calculated directly from the dataset with simple counting operations, similar, for instance, to the ones that have allowed us to calculate probabilities $P(\text{Outlook} = \text{sunny} \mid \text{class} = \text{Yes})$ or $P(\text{Outlook} = \text{sunny} \mid \text{class} = \text{No})$ in Example 9.2.

Looking at the definition of a BN like the one in Figure 9.3, it is also not difficult to understand that BNs are a construct that allows for more general computations than the "simple" prediction of a class label for a given target variable. More specifically, using BNs it is possible to estimate the probability of *any* variable having any of its possible values, given actual values of any of the others.

Just as an example, let us now calculate the probability that it is raining (Rain=True), knowing that the grass is wet (Wet Grass = True). From now on, for simplicity, variable Wet Grass will be abbreviated as G, Rain as R, Sprinkler as S. Furthermore, for any variable β, $\beta = \text{True}$ will be abbreviated by simply writing β,

while $\beta =$ False will be represented using the notation $\neg\beta$. So, the probability we want to estimate is: $P(R|G)$. Directly from the definition of conditional probability, we have:

$$P(R|G) = \frac{P(G,R)}{P(G)} \tag{9.1}$$

The right part of Equation (9.1) can be transformed, obtaining:

$$P(R|G) = \frac{P(G,S,R) + P(G,\neg S,R)}{P(G,S,R) + P(G,S,\neg R) + P(G,\neg S,R) + P(G,\neg S,\neg R)} \tag{9.2}$$

Now, remembering that, for the definition of joint probability, for each triple of events A, B and C we have:

$$P(A,B,C) = P(A|B,C)\ P(B|C)\ P(C) \tag{9.3}$$

we can transform each term of Equation (9.2) into a form that is similar to Equation (9.3), and then calculate its value using the values in the CPTs.

Let us begin with term $P(G,S,R)$. From Equation (9.3), we have:

$$P(G,S,R) = P(G|S,R)\ P(S|R)\ P(R) \tag{9.4}$$

But each term on the left-hand side of Equation (9.4) can be extracted directly from the CPTs. So, Equation (9.4) becomes:

$$P(G,S,R) = P(G|S,R)\ P(S|R)\ P(R) = 0.99 \cdot 0.01 \cdot 0.2 = 0.00198 \tag{9.5}$$

Analogously, for the other terms in Equation (9.2), we have:

$$P(G,\neg S,R) = P(G|\neg S,R)\ P(\neg S|R)\ P(R) = 0.8 \cdot 0.99 \cdot 0.2 = 0.1584 \tag{9.6}$$

$$P(G,S,\neg R) = P(G|S,\neg R)\ P(S|\neg R)\ P(\neg R) = 0.9 \cdot 0.4 \cdot 0.8 = 0.288 \tag{9.7}$$

$$P(G,\neg S,\neg R) = P(G|\neg S,\neg R)\ P(\neg S|\neg R)\ P(\neg R) = 0.0 \cdot 0.6 \cdot 0.8 = 0.0 \tag{9.8}$$

Finally, substituting Equations (9.5), (9.6), (9.7) and (9.8) into Equation (9.2), we obtain:

$$P(R|G) \ = \ \frac{0.00198 + 0.1584}{0.00198 + 0.288 + 0.1584 + 0.0} \ \approx \ 0.3577 \tag{9.9}$$

So, we can conclude that, if the grass is wet, the *a posteriori* probability that it is raining is approximately equal to 35%.

Fixing some values of some variables and, with that information, calculating the probability that another variable has some value is a task that is left as an exercise. For instance, analogously to what happens for Naïve Bayes, to predict a class label in a binary classification task, one may calculate the probability of a variable having a value and the probability of the same variable having the other possible value in the same conditions. The higher of these two probabilities is finally the one that

directs our decision on the predicted class label. Interestingly, as we have seen in this example, when we use BNs, we do not need an unseen observation to be "complete" (i.e., all the variables to have a value) to calculate the probability of a possible prediction outcome. For instance, here we have calculated the probability that it is raining, knowing that the grass is wet, without having any information on whether the sprinkler is active or not.

In the process discussed in the above example, all the steps were formal, except for one, which still appears to be quite heuristic: the determination of the BN topology. There are basically two ways of building the topology of a BN: a manual construction or an automatic design (so called "topology learning").

Manual construction of a Bayesian network assumes prior expert knowledge of the underlying domain. In particular, the variable interdependencies must be known. However, in some cases, the task of manually defining the topology of the network is too complex for humans. In those cases, one may try to employ algorithms that have the objective of inferring the topology directly from the data. The interested reader is referred to [Singh and Valtorta, 1995, Chen, 2016] for surveys and discussions on existing algorithms. Even though research on this subject is still ongoing, interesting results have been recently obtained, among others, in [Wu et al., 2001, Beretta et al., 2018].