11 Probabilistic Reasoning

Probability is important to artificial intelligence because it gives us a way of asking (and answering) questions such as: *what is the most likely reason for X?* Or *how likely is it that X occurred?* And don't forget: *A and B are true, is X likely to be true?* Searching for solutions through the space of *probable* relationships is a form of reasoning known as **probabilistic reasoning**. We might see such questions in medical diagnosis software: *the patient has symptom X and symptom Y, but the test for Z came back negative. What is the chance he has disease D?*

Probability doesn't just relate static facts with one another: it can also be used for time series. We can ask questions like *if A happened, then A happened again, then B happened, then A happened again, what is the most likely cause for these events?* Or *if A happened, then B happened, then B happened again, what is likely to happen next?* This kind of reasoning is often performed with simple probabilistic models called **Markov Models** and **Hidden Markov Models** (or HMMs). HMMs show up everywhere in AI, but they're particularly prominent in natural language processing, where ask questions like *the audio signal consisted of these series of sounds strung together. What's the chance that this was supposed to be saying "Hello, World?"*.

Another place where probabilistic reasoning shows up in time series is in **filters** used in robotics. These are extensions on Hidden Markov Models which allow us to ask questions like: *I started at location L, then at timestep 1 my sensors recorded X, and I then did action A. Then at timestep 2 my sensors recorded Y, and I then did action B. Where am I likely located now?* Filters are crucial for robots to maintain a belief about their likely current state in their world.

We are going to focus on static (non-time-series) probabilistic reasoning. Here's a famous example. In 2002, Paul Graham,[88] in an early blog entry[89] proposed an approach for doing spam filtering based on a simple form of probabilistic reasoning called **Naive Bayes Spam Filters**. This touched off the wave of spam filter software packages. The general relationship between spam and probabilistic reasoning goes like this: *if a message has features A, B, and C, what is the chance that it is spam?* Features might be words or phrases: or features like "the message is in HTML" or "the message came from Nigeria".

In this section, we'll review a few basic concepts in probability and then work our way up through the Naive Bayes model as an application of them.

11.1 Terminology and Review

Probability deals with the likelihood that **variables** will hold various **values**. We denote variables with capital letters like A or X or *Temperature*. Values can be anything: but often we need to represent the *notion* of a value in an equation. We do this with something like *Surface=flat* or *Temperature=23.0* or *Angry = false*.

Variables have **domains** which indicate the kinds of values they can take on. For example, a variable can be boolean (it can only be *true* or *false*), or it might be discrete (it can be set to one of

[88] Paul Graham is someone worth knowing about. He wrote two influential Lisp texts, *On Lisp* and *ANSI Common Lisp*. He then was an early World Wide Web pioneer, co-founding ViaWeb (famously written in Lisp), which later was bought by Yahoo! and became Yahoo! Stores. He has since co-founded Y Combinator, a micro-venture capital firm specializing in early two-guys-in-a-garage software startups. Y Combinator funded the development of Reddit, Scribd, Airbnb, Dropbox, Disqus, and Posterous, among many others. Graham is also well known for his blog, at http://www.paulgraham.com/

[89] http://www.paulgraham.com/spam.html

some N unique values), or it might be continuous (it can have any real-valued number within some prespecified range. Indeed, the domain of a variable could be more complex than this: it could be the domain of tree structures, or the domain of strings, etc.: but usually the first three (boolean, discrete, continuous) suffice for most things of interest.

We often want to talk about a variable having been bound to a known value but we don't want to be specific about it. In this case we'll use a lower-case version of the variable to refer to that value. Such as "suppose our variable A has been bound to the value a." If we want to be *specific* about the value, we will use a special subscript notation. For example, boolean values will be referred to with + and -, as in"suppose our variable A has been set to a_+", meaning it's been set to *true*. Likewise a_- means A has been set to false. Similarly if we have a larger discrete variable, say B, which has three possible values, we might refer to them as b_1, b_2, and b_3.

Simple Probabilities and Distributions The basic operator in probability is $P(\ )$, which denotes the probability that something is true. For example, we might say: $P(Angry = false) = 0.23$, or simply $P(angry\_) = 0.23$, which means that the probability that the *Angry* variable is set to *false* is 0.23 out of 1.0.

When only value bindings are found inside the P operator (as in $P(angry\_)$), then P denotes a **probability**: a single number from 0.0 to 1.0 inclusive.

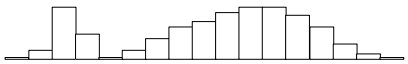

Figure 67 A Histogram

Probability Reminder 1 *For any value* a : $0.0 \leq P(a) \leq 1.0$

But we can also place unbound variables in there as well, as in: $P(Color)$. This is called a **distribution**, and it is the set of probabilities, one for every value that *Color* may take on. If *Color* has a boolean or discrete domain, then this probability distribution may be thought of as a table of elements. For example, if *Color* can take three values *red*, *blue*, and *teal*, then $P(Color)$ may look like:

$$P(Color) = \{P(Color = red) = 0.2,\ P(Color = blue) = 0.5,\ P(Color = teal) = 0.3\}$$

Simple distributions like this must **add to 1.0**. Note that **this is not a matrix**.[90] It's just a **set of probabilities**. The order of the rows is entirely unimportant. Often such tables are organized in a visualization known as a **histogram**, with one bar for each value, representing its probability.

When a distribution is continuous, we can't represent it as a table or a histogram. Instead, it takes the form of a continuous curve whose integration sums to 1.0. The probability of a given value x is represented by the corresponding the y value on the curve.

Figure 68 A Continuous Distribution

Probability Reminder 2 *In a simple discrete distribution* $P(D)$*, it must be the case that* $\sum_{d \in D} P(d) = 1.0$
In a simple continuous distribution $P(C)$*, it must be the case that* $\int P(C)dC = 1.0$

[90] If you think about it, one of these values can be omitted. Since the table sums to one, the probability of the final value is inferable from the probabilities of the others. Along that same vein, you don't *need* both values in a boolean distribution: you can get away with just one, since the other is simply 1 minus it.

Joint Probabilities and Distributions A **joint probability** is the probability of not one but N variables being set to specific values. For example, given three variables A, B, and C, let's say that the probability is 0.234 that A is set to *true*, B is set to *true*, and C is set to *false* at the same time. We denote this as:

$$P(A\text{=}true, B\text{=}true, C\text{=}false) = 0.234 \qquad \text{(or)} \qquad P(a_+, b_+, c_-) = 0.234$$

Note the use of commas to indicate "and".[91] Just as in the case of simple probabilities, joint probabilities must range from 0.0 to 1.0.

Probability Reminder 3 *The order of values in a joint probability are commutative. Thus* $P(a_+, b_+, c_-) = P(c_-, b_+, a_+)$*. It doesn't matter what order you put them.*

A **joint probability distribution** is the collective distribution of several variables together. A joint probability distribution is a set of probabilities, one for each possible permutation of values that the variables could simultaneously hold. For example, if A, B, and C were all boolean, then there would be eight probability values in the distribution $P(A, B, C)$. It might look like:

$$\begin{aligned} P(A, B, C) = \{&P(a_+, b_+, c_+) = 0.1,\ P(a_-, b_+, c_+) = 0.0,\ P(a_+, b_-, c_+) = 0.392,\ P(a_-, b_-, c_+) = 0.05,\\ &P(a_+, b_+, c_-) = 0.234,\ P(a_-, b_+, c_-) = 0.123,\ P(a_+, b_-, c_-) = 0.001,\ P(a_-, b_-, c_-) = 0.1\} \end{aligned}$$

(Yes, it sums to 1) Or what if D was boolean but E could have the values e_1, e_2, and e_3? Then we'd have a set of six elements.

$$\begin{aligned} P(D, E) = \{&P(d_+, e_1) = 0.1,\ P(d_-, e_1) = 0.2,\ P(d_+, e_2) = 0.3,\ P(d_-, e_2) = 0.0,\\ &P(d_+, e_3) = 0.4,\ P(d_-, e_3) = 0.0\} \end{aligned}$$

Probability Reminder 4 *The order of variables in a joint distribution are commutative. Thus* $P(D, E) = P(E, D)$*. It doesn't matter what order you put them.*

Just like simple distributions, the probabilities of all the elements in the joint distribution must sum to 1.0.

What if D and E were continuous variables? Then the distribution would be a two-dimensional surface, and the volume under the surface would sum to 1.0.

What if D was continuous but E was boolean? Then the joint distribution would be best thought of as two one-dimensional continuous functions (one for $E = e$, one for $E = \neg e$). The sum of the areas of *both* functions must total to 1.0.

Mixed Distributions There's absolutely no reason why you can't have *both* variables *and* values inside $P()$. In this case, the values are assumed to be fixed, and the variables may, well, vary over their domain. For example, consider three variables X, Y, and Z. X and Y are boolean, and Z may be one of z_1, z_2, z_3. What does $P(x_-, Y, Z)$ mean?

It represents the following set (I made up the probabilities as usual):

$$\begin{aligned} P(x_-, Y, Z) = \{&P(x_-, y_+, z_1) = 0.01,\ P(x_-, y_-, z_1) = 0.02,\ P(x_-, y_+, z_2) = 0.03,\\ &P(x_-, y_-, z_2) = 0.04,\ P(x_-, y_+, z_3) = 0.05,\ P(x_-, y_-, z_3) = 0.81\} \end{aligned}$$

[91] Alternatively, you may see the notation $P(a \wedge b \wedge \neg c) = 0.234$.

Notice that x_- never varies. Important note: **mixed distributions don't have to sum to 1.0**. In essence, the values in the mixed distribution are projections of the full joint distribution into those subspaces where the such values are true. This is a subset of the full joint distribution, and so the sum is only part of the the full 1.0.

My Notation for Multiple Variables or Values Sometimes I group multiple variables together, or multiple values together, like this: $P(A...)$. This means the probability of some collection of variables designated collectively as $A...$ This might be a subset of the variables in the probability distribution, for example, it's fine to say $P(A, B..., C, D)$. This means the a probability distribution including A, C, D and some number of additional variables designated as $B...$

The same goes for values. For example, I could say: $P(x...)$, meaning the probability of some set of values collectively designated as $x...$. And you can mix things up too: $P(A, b..., c, X..., Y, z)$ Remember once again that order doesn't matter.[92]

11.2 Conditional Probability

So far we have seen **joint probability distributions**, which answer what the probability is of a set of variables holding a certain permutation of values. There's another kind of probability distribution, equally important, known as a **conditional probability distribution**[93] A conditional probability distribution answers the question: if certain values hold certain values, what is the probability of other variables holding still other values? For example: if I have a rash and neurological symptoms, what is the probability that I have Lyme Disease? Or give the cards that have already been played, what is the probability that my opponent has a Three-of-a-Kind? Or if a congressman votes Yes for Bill A and No for Bills B and C, what is the probability that he is a Democrat? Or if my mail message is from Nigeria and says "one million" in its body text, what is the probability that it is spam?

We write probabilities to answer "if-then" questions like this in the following form: $P(then \mid if)$. For example: $P(Democrat \mid a_+, b_-, c_-)$. You can have any number of values on either side of the vertical bar: $P(a, b, c... \mid d, e..., f, g)$.[94] The relationship between conditional probability and joint probability distributions looks like this:

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

Or more generally,

$$P(A.., B...) = P(A... \mid B...)\, P(B...)$$

The same goes for values too: you can mix in values anywhere there's a variable. Thus this is perfectly legal:

$$P(A, b, c, D) = P(A, b \mid c, D)\, P(c, D)$$

Note that you can stretch this out if you like:

[92] Russell and Norvig (*Artificial Intelligence: A Modern Approach*) write it differently, using boldface. Instead of $x...$ they'd say $\mathbf{x}$. And instead of $X...$ they'd say $\mathbf{X}$.

[93] Sometimes you'll here the term **posterior probability** instead of conditional probability; and **prior probability** instead of **joint probability**.

[94] In fact, you can have *nothing* on the right side of the bar, in which case it's just a joint probability and you can take out the bar.

$$P(A, b, c, D) = P(A, b \mid c, D)\, P(c, D) = P(A, b \mid c, D)\, P(c \mid D)\, P(D)$$

Conditional distributions are commutative but only on the same side of the vertical bar. That is, $P(A, B|C, D) = P(B, A|D, C)$, but it's not the case that $P(A, B|C, D) = P(A, C|D, B)$.

Conditional Distributions Don't Have to Sum to 1.0 Well, they *do* but it's not how you think. A conditional distribution must sum to one for any given value permutation of the "if" variables. For example, imagine if W, X, Y, and Z are boolean variables. Thus for the distribution $P(W, X \mid Y, Z)$, we have:

$$\{P(w_+x_+ \mid y_+, z_+), P(w_-, x_+ \mid y_+, z_+), P(w_+x_- \mid y_+, z_+), P(w_-, x_- \mid y_+, z_+)\} \quad \text{must sum to 1.0}$$
$$\{P(w_+x_+ \mid y_-, z_+), P(w_-, x_+ \mid y_-, z_+), P(w_+x_- \mid y_-, z_+), P(w_-, x_- \mid y_-, z_+)\} \quad \text{must sum to 1.0}$$
$$\{P(w_+x_+ \mid y_+, z_-), P(w_-, x_+ \mid y_+, z_-), P(w_+x_- \mid y_-, z_+), P(w_-, x_- \mid y_-, z_+)\} \quad \text{must sum to 1.0}$$
$$\{P(w_+x_+ \mid y_-, z_-), P(w_-, x_+ \mid y_-, z_-), P(w_+x_- \mid y_-, z_-), P(w_-, x_- \mid y_-, z_-)\} \quad \text{must sum to 1.0}$$

Assuming W, X, Y, and Z are all boolean variables, you can think of $P(W, X \mid Y, Z)$ as consisting of the four subdistributions $P(W, X \mid y_+, z_+), P(W, X \mid y_-, z_+), P(W, X \mid y_+, z_-)$, and $P(W, X \mid y_-, z_-)$, *each of which* must sum to 1.0 independently. (Obviously for other kinds of variables, you just have more permutations of values).

Bayes Rule and Independence Consider again the relationship $P(A, B) = P(A \mid B)\, P(B)$. Since the A and B are commutative in a joint distribution, we also have $P(A, B) = P(B, A) = P(B \mid A)\, P(A)$. And thus:

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

This is Bayes' Rule, more commonly written as:

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

As usual, the same goes for values or mixtures of values and variables.

We say that two variables are **independent** of one another if the probabilities of one aren't conditional on the other. For example, if you roll two dice, the probability that one will come up as a 3 is independent of the probability that the other one will come up a 5. However if you take two cards from a deck, the probability that one is a Jack is **dependent** on the probability that the other one is also a Jack. If the second card was a Jack, the probability for the first card being a Jack is lower (because there are only three Jacks left).

When two variables are independent, we can say this fact:

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

That is, the particular value that B is set to has no influence on the probability of A. Plugging this into Bayes' rule, we get:

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

And thus

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

Thus if A is independent of B, then B is also independent of A. Last but not least, recall that $P(A, B) = P(A \mid B)\, P(B)$. Plugging in what we have learned, we get:

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

This means that when variables are independent, we don't need to store their (potentially large) joint probability distribution: we can just store smaller per-variable distributions.

Generalized Bayes Rule and Conditional Independence The more general form of Bayes' Rule looks like this:

$$P(A... \mid B..., C...) = \frac{P(B... \mid A..., C...)\, P(A... \mid C...)}{P(B... \mid C...)}$$

We say that two variables A and B are **conditionally independent** given a third variable C when, if we know the value of C, then A and B are independent. But if we don't know C, then A and B must be considered dependent. Conditional independence happens a lot in situations where A causes C, which in turn causes B, or where C causes both A and B separately. As an example of the former, imagine a class whose grade is largely based on a final exam. Studying for the exam increases your chance of passing the exam, and thus getting an A in the class. Thus the probability of passing the class is connected to the probability that a student has studied. But only if we don't know what the student got on the exam. If we know the student passed the exam, then the probability that the student studied is *immaterial* to the probability that he passed the class because, in this world, studying only effects the exam. Thus knowledge about the exam result changes whether or not studying and passing the class are related for a given student.

We can say A and B are conditionally independent given C like this:

$$P(A \mid B, C) = P(A \mid C)$$

... that is, if we know C, then B doesn't have any affect on the probability of A. We can use the general Bayes Rule to derive the same interesting facts as we did in the independence section above:

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

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

$$P(B \mid C) = P(B \mid A, C)$$

Furthermore, since $P(A, B, C) = P(A \mid B, C)\, P(B, C)$, and $P(B, C) = P(B \mid C)P(C)$, we can say (for variables A and B conditionally independent given C):

$$P(A, B, C) = P(A \mid C)\, P(B \mid C)P(C)$$

We can continue this pattern, so in general if we have a bunch of conditionally independent variables $V^{(1)}, ..., V^{(n)}$, we can say:

$$P(V^{(1)}, ..., V^{(n)}, C) = P(V^{(1)} \mid C)P(V^{(2)} \mid C)...P(V^{(n)} \mid C)P(C)$$

Or put more succinctly,

$$P(V^{(1)}, ..., V^{(n)}, C) = P(C)\prod_{i=1}^{n} P(V^{(i)} \mid C) \tag{2}$$

Just like independence allows us to reduce the complexity of our distributions, so does conditional independence. We'll later use conditional independence assumptions as a trick to make computation tractable.

11.3 Queries

Think of the probabilistic reasoning as operating over a universe of variables, each which may be set to some value. We divide these variables into three categories:

- **Evidence** These are variables which have a set value. This is the stuff we know. We will call our evidence **e**.... Notice that *e*... are values rather than variables: because we know the values already.
- **Query Variables** These are the variables whose distributions we're interested in knowing. Often, but not always, there is a single query variable. We will call our query variables **X**...
- **Hidden Variables** These are variables whose distributions we don't care about. We'll call these **Y**...

Thus we can define a **probabilistic query** as asking: given the evidence I know, what are the probability distributions over the things I want to know about? For example, given various features of an email message, what is the probability of it being spam? That is: what is ...

$$P(X... \mid e...)$$

In theory-world, the simplest way to arrive at a solution is to convert this query into an equation involving the **full joint distribution**, that is, the joint distribution which includes *all of the variables in our system*. This distribution is $P(X..., Y..., e...)$. Though in reality the full joint distribution is often astronomically huge, so this isn't an efficient approach, and we'll have to come up with some simplifying assumptions. For this Section, we'll use a very simplistic simplifying set of assumptions called the Naive Bayes model. One of the simplifications in Naive Bayes is that there are *no hidden variables* $Y...$, so we don't have to deal with them. Thus for purposes of this Section, but *not* the next Section 12, we may assume that the full joint distribution is simply $P(X..., e...)$.

How do we get to this joint distribution? Recall that we can do it as:

$$P(X..., e...) = P(X... \mid e...)\, P(e...)$$

Normalization Now we have a problem: we probably don't *know* the value of $P(e...)$. In the spam example, $P(e...)$ is the probability of messages in general having certain features (being from Nigeria, etc.). We don't know this. But it's okay: we can get around it with a sneaky trick called the **normalization constant**. Since $P(e...)$ is a single value, let's define the constant $\alpha = 1/P(e...)$. Now we can rewrite this equation as:

$$P(X... \mid e...) = \alpha P(X..., e...)$$

What does this buy us? It turns out we don't need to ever compute α. Instead we can treat it as a reminder to **normalize** everything on the right-hand side (in this case $P(X..., e...)$) after calculating it. Normalization is a simple procedure which ensures that the distribution sums to 1.0. It's easy: just divide all the probabilities by their sum.

Algorithm 34 *Normalize a Discrete Conditional Distribution*

1: $P(A...|b...) \leftarrow$ distribution
2: $n \leftarrow 0$
3: **for** each permutation $[a...] \in [A...]$ **do**
4: $\quad n \leftarrow n + P(a...|b...)$
5: **for** each permutation $[a...] \in [A...]$ **do**
6: $\quad P(a...|b...) \leftarrow P(a...|b...)/n$
7: **return** $P(A...|b...)$

For example, let's say that our distribution $P(A|b) = \{P(a_1|b) = 0.2, P(a_2|b) = 0.3, P(a_3|b) = 0.0\}$. The sum of the probabilities is 0.5. So we divide each of them by 0.5, resulting in $P(A|b) = \{P(a_1|b) = 0.4, P(a_2|b) = 0.6, P(a_3|b) = 0.0\}$. Now they sum to 1.0. Of course, if our distribution is continuous, then we have to divide the distribution function by the area under its curve: we are in essence forcing the distribution function to have an area of 1.0.

So the normalization constant is another trick in our arsenal to reduce what we have to know in order to solve queries. By the way, another common use of the normalization constant is when we apply Bayes rule to queries of the form $P(X... \mid e...)$, like this:

$$P(X... \mid e...) = \frac{P(e... \mid X...)\, P(X...)}{P(e...)}$$

Since $P(e...)$ is a single probability value, we assign $\alpha = 1/P(e...)$ and we get:

$$P(X... \mid e...) = \alpha P(e... \mid X...)\, P(X...)$$

This is again very useful because often we don't know what $P(e...)$ is.

11.4 The Naive Bayes Model: One Possible Simplification of the Joint Distribution

One simple, indeed simplistic, approximation of the full joint distribution is the Naive Bayes Model. This is a trivial model useful for cause-and-effect relationships among variables. It's almost *never* a correct description of the world. But it's good enough to get by with in some cases, and it simplifies the joint probability *radically*.

The Naive Bayes Model assumes that one or more variables are **causes** and the other variables are **effects**. The effects are our evidence, and the causes are our query variables. There are no hidden variables. We see effects occur in the world and want to know what the probability is that they're being caused by some cause.

The simplifying trick in the Naive Bayes Model is that it assumes that all of the effects are conditionally independent given the causes. That is, the only relationship between the effects is in how they all stem from a given cause. This radically simplifying assumption allows us dramatically reduce the full joint distribution into a bunch of very small and simple distributions which are easily computed.

As mentioned before, Naive Bayes also assumes that there are no hidden variables Y..... Thus the full joint distribution is simply $P(X..., e...)$ where $X...$ are the causes and $e...$ are the effects (and we now the effects, hence they're evidence values). Let's break out the effect variables, rewriting this as $P(X..., e^{(1)}, e^{(2)}, ..., e^{(n)})$. Because $e^{(1)}$ is conditionally independent of the other effects given the $X...$, we can use Equation 2 (page 151) to get:

$$P(X..., e^{(1)}, e^{(2)}, ..., e^{(n)}) = P(X...) \prod_{i=1}^{n} P(e^{(i)} \mid X...)$$

Now we can express the query as:

$$P(X... \mid e^{(1)}, e^{(2)}, ..., e^{(n)}) = \alpha P(X..., e^{(1)}, e^{(2)}, ..., e^{(n)}) = \alpha P(X...) \prod_{i=1}^{n} P(e^{(i)} \mid X...)$$

Example: A Spam Filter Let's say that we want to know if a given email message is spam. This is a single boolean query variable (X = spam/not spam). We also have a bunch of *evidence*: these are features of email messages, and we can test to see if these features occur in our email message of interest. For example, we may have assembled a list of words which may or may not be in our email message, plus other features like whether the message is an HTML message; whether it has forged email headers; whether it comes from Nigeria, an so on. To keep things simple we'll assume that all these variables are boolean (yes/no).

The spam filter is **trainable**: the user is required to label all his email as either *spam* or *non-spam*, and submit this information to the filter so it can gather statistics.

For each of our evidence variables $E^{(i)}$, we need to assemble a distribution which says how often (the probability) that evidence appears in spam; and a separate distribution which says how often (the probability) the evidence appears in non-spam. This is $P(E \mid x_+)$ and $P(E \mid x_-)$ respectively. It suffices to store each of these distributions as a single number: after all, if we have $P(e^{(i)} \mid x_+)$, we can easily compute $P(e_-^{(i)} \mid x_+) = 1 - P(e_+^{(i)} \mid x_+)$. But if you like, you can store them as the full set of two numbers $\{P(e_+^{(i)} \mid x_+), P(e_-^{(i)} \mid x_+)\}$ (and of course $\{P(e_+^{(i)} \mid x_-), P(e_-^{(i)} \mid x_-)\}$). How would you collect these numbers? Easy: by maintaining the full corpus of spam messages and non-spam messages, as labelled by the user, and then counting the rate at which the particular evidence feature appears (or doesn't appear) in the spam, and the rate at which appears (or doesn't appear) in the non-spam. Those are your four numbers.

We'll also need $P(X)$, the overall distribution of spam and non-spam messages. This is easy: $P(x_+)$ is the fraction of email that the user labelled as spam, and $P(x_-)$ is the fraction labelled as legitimate.

Armed with our $P(X)$ distribution, and our various $P(E^{(i)}|X)$ distributions, we're ready to go. We examine the message of interest and extract its feature values: $e_+^{(1)}, e_-^{(2)}, e_-^{(3)}$, and so on. Now we compute the query distribution $P(X \mid e_+^{(1)}, e_-^{(2)}, e_-^{(3)}, ...)$ as:

1. Compute the probability of spam: $P(x_+ \mid e^{(1)}, e^{(2)}, e^{(3)}, ..., e^{(n)}) = P(x_+) \prod_{i=1}^{n} P(e^{(i)} \mid x_+)$ (for various true and false settings of the $e^{(i)}$). Let's say this comes to 0.8.

2. Also the non-spam probability: $P(x_- \mid e^{(1)}, e^{(2)}, e^{(3)}, ..., e^{(n)}) = P(x_-) \prod_{i=1}^{n} P(e^{(i)} \mid x_-)$ (for various true and false settings of the $e^{(i)}$). Let's say this comes to 0.7.

3. Our probability distribution is $\{0.8, 0.7\}$. But remember α! The full formula was $P(X \mid e...) = \alpha P(X) \prod_{i=1}^{n} P(e^{(i)}|X)$. We have to normalize. As a result, we actually have $\{0.5333, 0.4666\}$.

4. The former tells us the likelihood of the message being spam. The second one tells us the likelihood of the message being non-spam. So our mail message is 0.53333 likely to be spam. Based on some threshold set by the user, we label this as "likely spam" (or not), then wait for the user to correct us (or not), at which time, we add the tagged message to our corpus and rebuild our statistics.

Notice that this is *extremely fast to compute* and *extremely small to store*. After all, since the variables are all boolean, $P(X)$ is only two numbers (or 1 if you're compact), and each $P(E^{(i)}|X)$ distribution is only four numbers (or two if you're compact). You could have statistics on thousands of words and hundreds of other features and still have only a couple thousand numbers.

But the Naive Bayes Model isn't naive for nothing: there's a big whopping assumption in this filter which makes it less than optimal. That assumption is: two evidence features are only related by their common impact on spam (the conditional independence assumption).

Consider: my family has an email list on Yahoo! Mail. Like many people, family members often send their email in HTML format (sigh). Also, Yahoo! Mail is, or was, a favorite (forged) source of spam messages. Imagine if I had "is the message in HTML?" as an evidence feature, and "does the mail appear to originate from Yahoo! Mail?" as another evidence feature. Say I get a lot of spam which is in HTML and a lot of spam which appears to come from Yahoo! Mail. It doesn't even have to be the *same spam*. As a result, the Naive Bayes Model eventually thinks that the only way that mail could *both* be in HTML and come from Yahoo! Mail is if it's spam. But there's another way: it's being sent by gentle and well-meaning relatives eager to share their latest baby picture. All of whom will be labelled as spam.

So it's sort of a crummy spam filter. But this simple approach spawned the whole field (see page 145 earlier).

Essentially the Naive Bayes Model is doing a weighted sum of features. A smarter approach would be to build a model which recognizes that there are various relationships *between* the evidence besides their impact on the query variable. The most common approach to do this is known as a **Bayes Network**.