

COMP9417 - Machine Learning

Tutorial: Probabalistic Classification

Question 1 (Bayes Rule)

Assume that the probability of a certain disease is 0.01. The probability of testing positive given that a person is infected with the disease is 0.95, and the probability of testing positive given that the person is not infected with the disease is 0.05.

- (a) Calculate the probability of testing positive.

Solution:

Let T denote the event that the test returns a positive, and let D denote the event that an individual has the disease, and \bar{D} the complement of D . Then we wish to find $P(T)$. It follows by the law of total probability that

$$\begin{aligned}P(T) &= P(T|D)P(D) + P(T|\bar{D})P(\bar{D}) \\&= 0.95 \times 0.01 + 0.05 \times (1 - 0.01) \\&= 0.059.\end{aligned}$$

In other words, the probability that a randomly chosen individual tests positive is 5.9%.

- (b) Calculate the probability of being infected with the disease, given that the test is positive.

Solution:

We want to compute the probability: $P(D|T)$. From Bayes rule, we have

$$\begin{aligned}P(D|T) &= \frac{P(D \cap T)}{P(T)} \\&= \frac{P(T|D)P(D)}{P(T)} \\&= \frac{0.95 \times 0.01}{0.059} \\&= 0.16,\end{aligned}$$

so the probability that someone has the disease given that the test is positive is only 16%. This result is somewhat surprising, but is due to the fact that the disease is so rare, that it pops up only in 1% of the population. In other words, our prior information on the disease pushes the probability of having the disease even after testing positive to be quite low.

- (c) Now assume that you test the individual a second time, and the test comes back positive (so two

tests, two positives). Assume that conditional on having the disease, the outcomes of the two tests are independent, what is the probability that the individual has the disease? (note, conditional independence in this case means that $P(TT|D) = P(T|D)P(T|D)$, and not $P(TT) = P(T)P(T)$.)

Solution:

We now want to compute the probability $P(D|TT)$. First note that

$$\begin{aligned}P(TT) &= P(TT|D)P(D) + P(TT|\bar{D})P(\bar{D}) \\&= P(T|D)^2P(D) + P(T|\bar{D})^2P(\bar{D}) \\&= (0.95)^2 \times 0.01 + (0.05)^2 \times (1 - 0.01) \\&= 0.0115.\end{aligned}$$

Then,

$$\begin{aligned}P(D|TT) &= \frac{P(TT|D)P(D)}{P(TT)} \\&= \frac{(0.95)^2 \times 0.01}{0.0115} \\&= 0.7848.\end{aligned}$$

Now we see that our data (the two positives) is starting to contribute more and more evidence, and so our prior belief about the disease contributes less to the probability.

Question 2 (Lecture Review)

In this question, we will review some important ideas from the lecture.

- (a) What is probabilistic classification? How does it differ from non-probabilistic classification methods?

Solution:

In a classification problem, we are given a set of classes, $C = \{c_1, \dots, c_K\}$, a set of examples $D = \{(\mathbf{x}_i, c_i) : i = 1, \dots, n\}$, where $x_i = (x_{i1}, x_{i2}, \dots, x_{ip})^T$ is a vector of features for the i -th subject. The goal of classification is, given a new datum, \mathbf{x}_* , to predict the 'best' class, c_* . The definition of 'best' varies across methods, for example, under kNN (which is not a probabilistic classification method), 'best' meant the class in agreement with the k neighbours of \mathbf{x}_* . Under the probabilistic approach 'best' is taken to mean 'most probable'. In order to talk about probabilities, we need to talk about probability distributions. In particular, we are most interested in the conditional probability distribution of a class given the data:

$$p(c_k | \mathbf{x}_*).$$

We would like to compute this probability for each of the k classes, and return our prediction as

$$c_* = \arg \max_k p(c_k | \mathbf{x}_*).$$

There are a couple of approaches we could take to learning this conditional distribution. One approach, called the discriminative approach, is to learn the distribution $p(c_k | \mathbf{x})$ directly - this is what Logistic regression does for example. An alternative approach that we will focus on here would be the generative approach. To understand this second approach, recall Bayes rule:

$$p(c_k | \mathbf{x}) = \frac{p(\mathbf{x} | c_k) p(c_k)}{p(\mathbf{x})},$$

which tells us that instead of focusing on $P(c_k | \mathbf{x})$, we could instead model $p(\mathbf{x} | c_k)$ and $p(c_k)$, for each k - this is called a generative approach. $p(\mathbf{x} | c_k)$ is termed the class conditional distribution, and $p(c_k)$ is termed the class prior distribution. Note that we do not need to model $p(\mathbf{x})$ since it can be computed from knowledge of the class conditional and prior distributions, i.e.

$$p(\mathbf{x}) = \sum_{k=1}^K p(\mathbf{x} | c_k) p(c_k).$$

Now, what we need to do under a generative approach is to choose the form of the class conditional distribution (e.g. continuous, multinomial, multivariate bernoulli, etc.) and class prior distribution (e.g. discrete uniform, categorical, etc). The prior is usually taken to be a discrete distribution since the number of classes is finite, whilst the class conditional distribution can be either continuous or discrete. One example of this approach is Gaussian (linear) discriminant analysis.

- (b) What is the Naive Bayes assumption and why do we need it?

Solution:

In the previous section we saw that in order to do generative classification, we must learn a class conditional distribution for each of the k classes in our data. Now, if our observations live in a high-dimensional space (the number of features p is very large), then learning these distributions can be very difficult. For example, if $p = 1000$, and we assume that $x|c_k \sim N(\mu_k, \sigma_k^2)$, then we would have to learn a 1000-dimensional Gaussian for each class. Learning high dimensional distributions can be very tricky, and we often need a large amount of data (relative to p) to be able to do it well. Naive Bayes gives us a way out of this problem. Before describing it, let's go back and rewrite the quantity we are trying to estimate:

$$\begin{aligned}
p(\mathbf{x}|c_k)p(c_k) &= p(\mathbf{x}, c_k) \\
&= p(x_1, x_2, \dots, x_p, c_k) \\
&= p(x_1|x_2, \dots, x_p, c_k)p(x_2, x_3, \dots, x_p, c_k) \\
&= p(x_1|x_2, \dots, x_p, c_k)p(x_2|x_3, \dots, x_p, c_k)p(x_3|x_4, \dots, x_p, c_k) \\
&\quad \vdots \\
&= p(x_1|x_2, \dots, x_p, c_k)p(x_2|x_3, \dots, x_p, c_k)p(x_3|x_4, \dots, x_p, c_k) \times \dots \times p(x_p|c_k)p(c_k).
\end{aligned}$$

These computations are valid for any probability distribution, and we have so far made zero assumptions. Naive Bayes introduces the assumption that the features are conditionally independent, this is written as

$$x_i \perp x_j | c_k \quad \text{for all } j \neq i.$$

Now, let's interpret this assumption intuitively. Let's assume that we are trying to classify whether a patient has or doesn't have diabetes, and I tell you that a particular patient, \mathbf{x} , in our dataset belongs to the class diabetes. Let's also assume that x_1 describes the blood pressure of the patient, and x_2 describes the weight of the patient. Now, the naive bayes assumption tells us that if we know the patient is diabetic, then also knowing that the patient has high blood pressure tells us nothing about their weight. Realistically, we expect that for patients with diabetes, there exists a positive correlation between blood pressure and weight. Under Naive Bayes however, we ignore this correlation for the sake of mathematical tractability.

Mathematically, the Naive Bayes assumption means that $p(x_i|x_j, c_k) = p(x_i|c_k)$. Using this, we can rewrite:

$$\begin{aligned}
p(\mathbf{x}|c_k)p(c_k) &= p(\mathbf{x}, c_k) \\
&= p(x_1|x_2, \dots, x_p, c_k)p(x_2|x_3, \dots, x_p, c_k)p(x_3|x_4, \dots, x_p, c_k) \times \dots \times p(x_p|c_k)p(c_k) \\
&= p(x_1|c_k)p(x_2|c_k)p(x_3|c_k) \times \dots \times p(x_p|c_k)p(c_k) \\
&= p(c_k) \prod_{i=1}^p p(x_i|c_k).
\end{aligned}$$

This means that instead of estimating a p -dimensional distribution, we now just have to estimate p different 1-dimensional distributions (for each of the k classes) instead - since estimating 1-dimensional distributions is much more straight forward, this is a much easier task!

- (c) Consider the problem from lectures of classifying emails as **spam** or **ham**, with training data summarised below: Each row represents an email, and each email is a combination of words taken

e_1	b	d	e	b	b	d	e		
e_2	b	c	e	b	b	d	d	e	c c
e_3	a	d	a	d	e	a	e	e	
e_4	b	a	d	b	e	d	a	b	
e_5	a	b	a	b	a	b	a	e	d
e_6	a	c	a	c	a	c	a	e	d
e_7	e	a	e	d	a	e	a		
e_8	d	e	d	e	d				

from the set $\{a, b, c, d, e\}$. We treat the words d, e as stop words - these are words that are not useful for classification purposes, for example, the word 'the' is too common to be useful for classifying documents as spam or ham. We therefore define our vocabulary as $V = \{a, b, c\}$. Note that in this case we have two classes, so $k = 2$, and we will assume a uniform prior, that is:

$$p(c_+) = p(c_-) = \frac{1}{2},$$

where $c_+ = \text{spam}$, $c_- = \text{ham}$. Review the multivariate Bernoulli Naive Bayes set-up and classify the test example: assume we get a new email that we want to classify: $e_* = \text{abbdebb}$

Solution:

Under the multivariate Bernoulli NB set-up, we encode the first email (e_1) as $\mathbf{x}_1 = (x_{1a} = 0, x_{1b} = 1, x_{1c} = 0)$, so that each of the features are binary and represent whether a word is present or not in a given email. Carrying this out for all emails in our dataset gives us:

	x_{ia}	x_{ib}	x_{ic}
e_1	0	1	0
e_2	0	1	1
e_3	1	0	0
e_4	1	1	0
e_5	1	1	0
e_6	1	0	1
e_7	1	0	0
e_8	0	0	0

So, under this model, we are choosing to ignore the frequency of words in the email and just consider whether a word appears or not in an email. This is equivalent to modelling the class conditional distribution as

$$p(\mathbf{x}|c_k) = \prod_{j \in V} p(x_j|c_k),$$

where $x_j|c_k \sim \text{Bernoulli}(p_j^k)$. In other words:

$$p(x_j|c_k) = (p_j^k)^{x_j} (1 - p_j^k)^{1-x_j}.$$

Note that p_j^k is the probability that a document of class k contains the word j or not. So, under this approach, we must estimate the following parameters:

$$p_a^+, p_b^+, p_c^+, p_a^-, p_b^-, p_c^-.$$

We will not go into detail here, but under the hood we are using maximum likelihood estimation to estimate these probabilities. It turns out that maximum likelihood gives us an intuitive estimate

$$p_j^k = \frac{\text{number of documents of class } k \text{ that contain the word } j}{\text{number of documents in class } k},$$

so, based on our data, this gives us:

$$\begin{aligned} p_a^+ &= \frac{2}{4}, & p_b^+ &= \frac{3}{4}, & p_c^+ &= \frac{1}{4}, \\ p_a^- &= \frac{3}{4}, & p_b^- &= \frac{1}{4}, & p_c^- &= \frac{1}{4}. \end{aligned}$$

Now, for $e_\star = abbdebb$, we have the encoding $\mathbf{x}_\star = (1, 1, 0)$, and we classify according to

$$c_\star = \arg \max_{k \in \{+, -\}} p(c_k) p(\mathbf{x}_\star | c_k).$$

First, for $k = +$, we have:

$$\begin{aligned} p(c_+) p(\mathbf{x}_\star | c_+) &\stackrel{(\text{NB})}{=} p(c_+) \prod_{j \in V} p(x_j = x_{\star j} | c_+) \\ &= p(c_+) \times p(x_a = x_{\star a} | c_+) \times p(x_b = x_{\star b} | c_+) \times p(x_c = x_{\star c} | c_+) \\ &= p(c_+) \times p(x_a = 1 | c_+) \times p(x_b = 1 | c_+) \times p(x_c = 0 | c_+) \\ &= p(c_+) \times p_a^+ \times p_b^+ \times (1 - p_c^+) \\ &= \frac{1}{2} \times \frac{2}{4} \times \frac{3}{4} \times \left(1 - \frac{1}{4}\right) \\ &= \frac{9}{64}, \end{aligned}$$

where (NB) is to signify that we are making the Naive Bayes assumption. A similar computation gives us:

$$\begin{aligned} p(c_-) p(\mathbf{x}_\star | c_-) &= p(c_-) \times p_a^- \times p_b^- \times (1 - p_c^-) \\ &= \frac{1}{2} \times \frac{3}{4} \times \frac{1}{4} \times \left(1 - \frac{1}{4}\right) \\ &= \frac{9}{128}. \end{aligned}$$

Therefore,

$$\begin{aligned} c_\star &= \arg \max_{k \in \{+, -\}} p(c_k) p(\mathbf{x}_\star | c_k) \\ &= \arg \max \left\{ \frac{9}{64}, \frac{9}{128} \right\} \\ &= c_+. \end{aligned}$$

- (d) Next, review Smoothing for the multivariate Bernoulli case. Why do we need smoothing What happens to our previous classification under the smoothed multivariate Bernoulli model?

Solution:

In the previous section, we saw how to make predictions under the Naive Bayes Multivariate Bernoulli model. Note that for each prediction, we have to multiply various probabilities with each other. These probabilities are of the form p_j^k or $(1 - p_j^k)$. Since we estimate these probabilities based on occurrences in the data, this can be problematic if we have not seen particular observations before. For example, assume that we never observe the word a in any spam emails, so that:

	x_{ia}	x_{ib}	x_{ic}
e_1	0	1	0
e_2	0	1	1
e_3	0	0	0
e_4	0	1	0
e_5	1	1	0
e_6	1	0	1
e_7	1	0	0
e_8	0	0	0

Now, our estimate of $p_a^+ = 0$. Therefore, any new data point, x_* , which has $x_{*a} = 1$, regardless of the values of the other features, gets $p(x_*|+) = 0$. Further note that if $p_a^+ = 1$, which means every spam email has the word a , we get similar behaviour, since any new email that does not have the word a gets probability of spam equal to zero, as $(1 - p_a^+) = (1 - 1) = 0$. This is not ideal behaviour, since we allow one feature to override the entire model, and in the case of document classification, the model is completely unable to deal with new words not before seen in our vocabulary. Therefore, the modification is often made to Naive Bayes in which we smooth our probability estimates. This amounts to altering our estimating equation slightly to

$$p_j^k = \frac{\text{number of documents of class } k \text{ that contain the word } j + 1}{\text{number of documents in class } k + \text{number of possible values of } X}.$$

Note that in the bernoulli model, the number of possible values of X is 2 since we can only have $X = 0$ or $X = 1$. If we had instead considered a model in which we had three categories {spam, ham, unknown}, then our random variable would 3 possible values. Such random variables are called Categorical random variables.

This modification ensures that $p(x|c_k)$ is never equal to zero, even if the number of times a word j appears in a document of class k is actually zero. So, our smoothed estimates now would be:

$$\begin{aligned} p_a^+ &= \frac{2+1}{4+2} = \frac{3}{6}, & p_b^+ &= \frac{3+1}{4+2} = \frac{4}{6}, & p_c^+ &= \frac{1+1}{4+2} = \frac{2}{6}, \\ p_a^- &= \frac{3+1}{4+2} = \frac{4}{6}, & p_b^- &= \frac{1+1}{4+2} = \frac{2}{6}, & p_c^- &= \frac{1+1}{4+2} = \frac{2}{6}. \end{aligned}$$

Estimation is done identically as before, except with our new estimates.

Note that smoothing can be thought of in a different way. We want our probability estimates to never be zero, which is equivalent to observing every possible combination of words and classes. In other words, we need to observe documents in each class that contain each word in our vocabulary, as well as documents in each class that do not contain each word in our vocabulary. We can therefore introduce ‘pseudo-documents’, that ensure we have seen every combination of outcomes at least once. This amounts to adding $e_{p1}, e_{p2}, e_{p3}, e_{p4}$ as follows

	x_{ia}	x_{ib}	x_{ic}
e_1	0	1	0
e_2	0	1	1
e_3	0	0	0
e_4	0	1	0
e_{p1}	1	1	1
e_{p2}	0	0	0
e_5	1	1	0
e_6	1	0	1
e_7	1	0	0
e_8	0	0	0
e_{p1}	1	1	1
e_{p2}	0	0	0

If this was our dataset, then none of our probability estimates based on the original estimation procedure would be zero.

- (e) Redo the previous analysis for the Multinomial Naive Bayes model without smoothing. Use the following test email: $e_* = abbdebbcc$

Solution:

Consider a k -sided die, with the probability of each side being θ_i , so that $\sum_{i=1}^k \theta_i = 1$. Now, if we throw the die n times (the n throws are independent) and record the frequency of each face of the die as a vector $X = (X_1, X_2, \dots, X_k)$, where X_i = number of times face i came up, then we say that X follows a multinomial distribution with parameters $\theta_1, \dots, \theta_k$ and n . We summarise this as:

$$X \sim \text{Multinomial}(n, \theta_1, \dots, \theta_k), \quad \sum_{i=1}^k \theta_i = 1, \quad \theta_i > 0 \forall i.$$

The probability mass function of the multinomial is:

$$\begin{aligned} P(X = (x_1, x_2, \dots, x_k)) &= \frac{n!}{x_1! \times x_2! \times \dots \times x_k!} \theta_1^{x_1} \times \dots \times \theta_k^{x_k} \\ &= \frac{n!}{\prod_{i=1}^k x_i!} \prod_{i=1}^k \theta_i^{x_i} \end{aligned}$$

Now, under this model, we choose to encode emails as count vectors,

	x_{ia}	x_{ib}	x_{ic}
e_1	0	3	0
e_2	0	3	3
e_3	3	0	0
e_4	2	3	0
e_5	4	3	0
e_6	4	0	3
e_7	3	0	0
e_8	0	0	0

where now the features represent the number of occurrences of a word, rather than whether or not they were present. We now model the class conditional distribution as a multinomial, which implicitly uses the Naive Bayes assumption since the multinomial distribution assumes independence across trials.

$$p(\mathbf{x}|c_k) = \frac{n!}{\prod_{j \in V} x_j!} \prod_{j \in V} p(x_j|c_k)^{x_j},$$

and we write $\theta_j^k = p(x_j|c_k)$ to denote the probability that a word in document of class k takes the value j . Note the subtle difference to the bernoulli model. Previously, the probabilities were to do with the probability that a particular word existed in a given class. The probabilities here are to do with how likely a randomly chosen word in the class is a particular word. Now, we need to estimate the following parameters for our model:

$$\theta_a^+, \theta_b^+, \theta_c^+, \theta_a^-, \theta_b^-, \theta_c^-.$$

We once more omit the details of how to derive the estimates of these parameters, and as one might expect, the estimates are:

$$\theta_j^k = \frac{\text{number of times word } j \text{ appears in class } k}{\text{number of words that appear in class } k}$$

so, based on our data, this gives us:

$$\begin{aligned} \theta_a^+ &= \frac{5}{17}, & \theta_b^+ &= \frac{9}{17}, & \theta_c^+ &= \frac{3}{17}, \\ \theta_a^- &= \frac{11}{17}, & \theta_b^- &= \frac{3}{17}, & \theta_c^- &= \frac{3}{17}. \end{aligned}$$

Now, given $\mathbf{x}_* = (1, 4, 2)$, we can compute

$$\begin{aligned} p(c_+)p(\mathbf{x}_*|c_+) &= p(c_+) \frac{7!}{1! \times 4! \times 2!} \times (\theta_a^1 \times \theta_b^4 \times \theta_c^2) \\ &= \frac{1}{2} \times \frac{7!}{4! \times 2!} \times \left(\frac{5}{17} \times \left(\frac{9}{17} \right)^4 \times \left(\frac{3}{17} \right)^2 \right) \end{aligned}$$

and similarly for $p(c_-)p(\mathbf{x}_*|c_-)$.

- (f) Repeat the analysis for the smoothed Multinomial Naive Bayes model.

Solution:

In our predictions here, just as in the multivariate bernoulli case, we must multiply various probabilities. Therefore, this model suffers the same potential problems when probabilities take the value zero. The difference here is that this only occurs when the *frequency* of a word in a particular class is zero. To get around this, we apply smoothing once more, which involves adding 1 to each of the numerators of our estimates. Note that for each class k , this means adding 1 to the numerator $|V|$ times, and so we add $|V|$ to the denominator to account for the $|V|$ extra (pseudo-documents) that we are artificially adding. Our estimates therefore become

$$\theta_j^k = \frac{\text{number of times word } j \text{ appears in class } k + 1}{\text{number of words that appear in class } k + |V|}.$$

Now, our estimates for the spam problem become

$$\begin{aligned} \theta_a^+ &= \frac{5 + 1}{17 + 3} = \frac{6}{20}, & \theta_b^+ &= \frac{9 + 1}{17 + 3} = \frac{10}{20}, & \theta_c^+ &= \frac{3 + 1}{17 + 3} = \frac{4}{20}, \\ \theta_a^- &= \frac{11 + 1}{17 + 3} = \frac{12}{20}, & \theta_b^- &= \frac{3 + 1}{17 + 3} = \frac{4}{20}, & \theta_c^- &= \frac{3 + 1}{17 + 3} = \frac{4}{20}. \end{aligned}$$

Question 3 (More Naive Bayes Practice)

Refer to the training data in the table below, which shows the number of times each of four words A , B , C and D occurs in each of eight documents, four in the positive class, and four negative.

Document No.	A	B	C	D	Class
1	2	0	4	4	+
2	0	3	3	0	+
3	3	0	0	2	+
4	0	0	2	0	+
5	0	0	0	1	-
6	3	0	0	0	-
7	4	3	0	0	-
8	4	0	0	1	-

Consider a new document x_* with 1 occurrence of A , no occurrences of B , no occurrence of C , and 2 occurrences of D . Recall that if $X = (X_1, X_2, X_3, X_4) \sim \text{Multinomial}(p_1, p_2, p_3, p_4)$, then

$$P(X = (x_1, x_2, x_3, x_4)) = \frac{(\sum_{i=1}^4 x_i)!}{\prod_{i=1}^4 x_i!} \prod_{i=1}^4 p_i^{x_i}.$$

- (a) Construct a Naive Bayes classifier for the problem of predicting whether a document should be classified as positive or negative using the multinomial distribution to model the probability of a word occurring or not in a class. Under your model, what is the value of $P(+|x_*)$? Do not use smoothing.

Solution:

We have the following (non-smoothed) multinomial probabilities:

$$\begin{aligned} p_A^+ &= \frac{5}{23}, & p_B^+ &= \frac{3}{23}, & p_C^+ &= \frac{9}{23}, & p_D^+ &= \frac{6}{23}, \\ p_A^- &= \frac{11}{16}, & p_B^- &= \frac{3}{16}, & p_C^- &= \frac{0}{16}, & p_D^- &= \frac{2}{16}, \end{aligned}$$

Therefore:

$$\begin{aligned} p(x_*|+) &= p((1, 0, 0, 2)|+) \\ &= \frac{3!}{1! \times 2!} \left(\frac{5}{23}\right)^1 \left(\frac{6}{23}\right)^2 \\ &= 0.04438235. \\ p(x_*|-) &= p((1, 0, 0, 2)|-) \\ &= \frac{3!}{1! \times 2!} \left(\frac{11}{16}\right)^1 \left(\frac{2}{16}\right)^2 \\ &= 0.03222656. \end{aligned}$$

and so

$$\begin{aligned}
 p(+|x_*) &= \frac{p(+|x_*)p(x_*)}{p(x_*)} \\
 &= \frac{p(+|x_*)p(x_*)}{p(+|x_*)p(x_*) + p(-|x_*)p(x_*)} \\
 &= \frac{0.04438235}{0.04438235 + 0.0322266} \\
 &= 0.5793366 \\
 &\approx 0.5793.
 \end{aligned}$$

- (b) Now, apply (add-1) smoothing to your probability estimates. Under the smoothed probabilities, what is $P(-|x_*)$ under the multinomial model?

Solution:

We have the following (smoothed) multinomial probabilities:

$$\begin{aligned}
 p_A^+ &= \frac{5+1}{23+4}, & p_B^+ &= \frac{3+1}{23+4}, & p_C^+ &= \frac{9+1}{23+4}, & p_D^+ &= \frac{6+1}{23+4}, \\
 p_A^- &= \frac{11+1}{16+4}, & p_B^- &= \frac{3+1}{16+4}, & p_C^- &= \frac{1+1}{16+4}, & p_D^- &= \frac{2+1}{16+4},
 \end{aligned}$$

where we add 4 to the denominator since that is the size of the vocabulary. Therefore:

$$\begin{aligned}
 p(x_*|+) &= p((1, 0, 0, 2)|+) \\
 &= \frac{3!}{1! \times 2!} \left(\frac{6}{27}\right)^1 \left(\frac{7}{27}\right)^2 \\
 &= 0.04481024 \\
 p(x_*|-) &= p((1, 0, 0, 2)|-) \\
 &= \frac{3!}{1! \times 2!} \left(\frac{12}{20}\right)^1 \left(\frac{3}{20}\right)^2 \\
 &= 0.0405.
 \end{aligned}$$

and so

$$p(-|x_*) = \frac{p(-)p(x_*|-)}{p(x_*)} = \frac{0.5 \times 0.0405}{0.5 \times 0.0405 + 0.5 \times 0.04481024} = 0.4747378 \approx 0.4747.$$

- (c) Recall that if $X \sim \text{Bernoulli}(p)$ then

$$P(X = x) = p^x(1-p)^{1-x}.$$

Construct a Naive Bayes classifier for the problem of predicting whether a document should be classified as positive or negative using the multivariate Bernoulli distribution to model the probability of a word occurring or not in a class. What is $p(+|x_*)$ now? Do not use smoothing.

Solution:

We have the following (non-smoothed) multivariate bernoulli probabilities:

$$\begin{aligned} p_A^+ &= \frac{2}{4}, & p_B^+ &= \frac{1}{4}, & p_C^+ &= \frac{3}{4}, & p_D^+ &= \frac{2}{4}, \\ p_A^- &= \frac{3}{4}, & p_B^- &= \frac{1}{4}, & p_C^- &= \frac{0}{4}, & p_D^- &= \frac{2}{4}, \end{aligned}$$

Therefore:

$$\begin{aligned} p(x_\star|+) &= p((1, 0, 0, 1)|+) \\ &= \frac{2}{4} \times \left(1 - \frac{1}{4}\right) \times \left(1 - \frac{3}{4}\right) \times \frac{2}{4} \\ &= \frac{2}{4} \times \frac{3}{4} \times \frac{1}{4} \times \frac{2}{4} \\ &= 0.046875. \\ p(x_\star|-) &= p((1, 0, 0, 1)|-) \\ &= \frac{3}{4} \times \left(1 - \frac{1}{4}\right) \times (1 - 0) \times \frac{2}{4} \\ &= \frac{3}{4} \times \frac{3}{4} \times 1 \times \frac{2}{4} \\ &= 0.28125 \end{aligned}$$

and so

$$p(+|x_\star) = \frac{p(+)p(x_\star|+)}{p(x_\star)} = \frac{0.5 \times 0.046875}{0.5 \times 0.046875 + 0.5 \times 0.28125} = 0.1428571 \approx 0.1429.$$

- (d) Now, apply (add-1) smoothing to your probability estimates. Under the smoothed probabilities, what is $P(-|x_\star)$ under the multivariate Bernoulli model?

Solution:

We have the following smoothed multivariate bernoulli probabilities:

$$\begin{aligned} p_A^+ &= \frac{2+1}{4+2}, & p_B^+ &= \frac{1+1}{4+2}, & p_C^+ &= \frac{3+1}{4+2}, & p_D^+ &= \frac{2+1}{4+2}, \\ p_A^- &= \frac{3+1}{4+2}, & p_B^- &= \frac{1+1}{4+2}, & p_C^- &= \frac{0+1}{4+2}, & p_D^- &= \frac{2+1}{4+2}, \end{aligned}$$

Therefore:

$$\begin{aligned}p(x_{\star}|+) &= p((1, 0, 0, 1)|+) \\&= \frac{3}{6} \times \left(1 - \frac{2}{6}\right) \times \left(1 - \frac{4}{6}\right) \times \frac{3}{6} \\&= \frac{3}{6} \times \frac{4}{6} \times \frac{2}{6} \times \frac{3}{6} \\&= 0.05555556.\end{aligned}$$

$$\begin{aligned}p(x_{\star}|-) &= p((1, 0, 0, 1)|-) \\&= \frac{4}{6} \times \left(1 - \frac{2}{6}\right) \times \left(1 - \frac{1}{6}\right) \times \frac{3}{6} \\&= \frac{4}{6} \times \frac{4}{6} \times \frac{5}{6} \times \frac{3}{6} \\&= 0.1851852.\end{aligned}$$

and so

$$p(-|x_{\star}) = \frac{p(-)p(x_{\star}|-)}{p(x_{\star})} = \frac{0.5 \times 0.1851852}{0.5 \times 0.1851852 + 0.5 \times 0.05555556} = 0.7692308 \approx 0.7692.$$