Week 4: Naive Bayes, Entropy and Backpropagation

http://mlvu.github.io

February 28, 2020

1 preliminaries: probability

Before we start, it's helpful to review your general grasp of probability. If the exercises below give you any trouble we recommend that you follow the links provided to brush up a little.

question 1:

1. In a few sentences, explain the difference between the Frequentist and the Bayesian interpretation of probability. A frequentist considers a probability an objective value: a property of the universe that can be measured by repeated experimentation, like measuring the probability that a bent coin lands heads, by flipping it repeatedly. A Bayesian considers probability an expression of uncertain belief. A Bayesian can talk about the probability that "John is having an affair". To a frequentist this is nonsense, because John is either having an affair or he isn't.

2. What is the difference between a sample space and an event space? A sample space is a space of individual things that can happen (like a die landing on a 6) and an event space is a space of sets of things from the sample space (like a die landing on an even number). Technically, an event space should be a sigma-algebra of a sample space, but for discrete probability distributions, we can think of the event space as a the powerset of the sample space. For continuous probability distributions, we can ignore the technical details.

3. What is the difference between a probability distribution and a probability density function? A probability function assigns probabilities to

events. If the sample space is continuous (like in the case of a normal distribution) all atomic events like "X is exactly 2.1" will have zero probability. Instead, we define a *probability density function*. We then get the probability of a subset of the sample space, like "X is between 1.95 and 2.15", by integrating over the probability density function.

question 2: In the following, p is a probability function and A and B are random variables. Which of the following are true?

1. Joint probability is symmetric: $p(A, B) = p(B, A)$. True
2. Conditional probability is symmetric $p(A \mid B) = p(B \mid A)$. False. This is not true in general (although it may be true for specific A and B).
3. Two random variables X and Y are conditionally independent on a third Z. Once we know X and Y, we also know the value of Z. False.
4. Two random variables X and Y are conditionally independent on a third Z. Once we know Z, also knowing X will tell us nothing extra about Y. True: conditional independence means that given the value of the conditional, X and Y are independent (so knowing the value of one reveals nothing about the other).

question 3: Assume that the probability that a given patient has diabetes is 0.01. We have a test for diabetes with a false positive rate of 0.05: if a patient has no diabetes, the test diagnoses it 5% of the time. The false negative rate is 0.1. You are a doctor, and you administer the test to a patient (knowing nothing else). The test says she has diabetes. What is the probability that she doesn't? Hint: this is a question about Bayes' rule. Reflect on the result. Is this what you would've expected? If not, where does the unexpected result come from?

Let's write down the given probabilities first. We'll use D and $\neg D$ for the events that the patient has and doesn't have diabetes respectively. We'll use T for the event of the test *saying* that the patient has diabetes and $\neg T$ for the test saying that she doesn't. We are given $p(D) = 0.01$, $p(T \mid \neg D) = 0.05$ and $p(\neg T \mid D) = 0.1$. We can derive $p(\neg T \mid \neg D) = 0.95$ and $p(T \mid D) = 0.9$. Now, we want to know $p(\neg D \mid T)$. We'll use Bayes' rule to reverse the conditional probability:

$$
\begin{aligned}
p(\neg D \mid T) &= \frac{p(T \mid \neg D)p(\neg D)}{p(T)} \\
&= \frac{p(T \mid \neg D)p(\neg D)}{p(T, D) + p(T, \neg D)} \\
&= \frac{p(T \mid \neg D)p(\neg D)}{p(T \mid D)p(D) + p(T \mid \neg D)p(\neg D)} \\
&= \frac{0.05 \times 0.99}{0.9 \times 0.01 + 0.05 \times 0.99} \\
&= \frac{5 \times 99}{90 + 5 \times 99} \approx 0.85
\end{aligned}
$$

Whether you expected this result is up to you, of course, but it's certainly not what you want form a test like this. And yet, the false positive and false negative rates don't seem extremely high. Inn the last line, I've multiplied everything by 100 so you can see where the imbalance comes from: the high prior probability that someone doesn't have diabetes dominates (in other words, we have high lass imbalance). In order for the test to be reliable, the false positive rate needs to be low enough to make the numerator small in proportion to the denominator.[1]

More practice? Follow these links:

1. `https://seeing-theory.brown.edu/` (especially this one)
2. `https://betterexplained.com/articles/a-brief-introduction-to-probability-statistics/`
3. `https://www.khanacademy.org/math/probability/probability-geometry/probability-basics/a/probability-the-basics`
4. `http://dept.stat.lsa.umich.edu/~moulib/probrefresh.pdf`

2 Naive Bayes

The following dataset represents a spam classification problem: we observe 8 emails and measure two binary features. The first is 0 if the word ”pill” occurs in the e-mail and the second is 1 if the word “meeting” occurs.

[1]There's a subtle difference between the phrase *false positive rate* as we use it here and as we've used it previously. What we described previously (like in lecture 3) is an estimate of the fpr, computed from the test set. What we discuss here is the “actual” fpr: an unknown quantity that we would normally only be able to estimate from data.

“pill”	“meeting”	label
T	F	Spam
T	F	Spam
F	T	Spam
T	F	Spam
F	F	Ham
F	F	Ham
F	T	Ham
T	T	Ham

question 4: We will build a naive Bayes classifier for this data. What is the defining property of naive Bayes? Why is it called “naive”?

The naive Bayes classifier assumes that the features are independent, *conditional on the class*. It is called naive, because this assumption is usually untrue. The resulting classifier nevertheless works well in many cases.

question 5: We build a naive Bayes classifier on this data, as described in the lecture. We get an email that contains both words. Which class does the classifier assign?

Call the class Y and the features X_p and X_m. The class probabilities are

$$p(Y \mid X_p, X_m) = \frac{p(X_p, X_m \mid Y)p(Y)}{p(X_p, X_m)} \,.$$

The class choice is

$$\arg\max_Y p(Y \mid X_p, X_m) = \arg\max_Y p(X_p, X_m \mid Y)p(Y) \,.$$

The *naive Bayes assumption* allows us to break the the probabilities up.

$$\arg\max_Y p(Y \mid X_p, X_m) = \arg\max_Y p(X_p \mid Y)p(X_m \mid Y)p(Y) \,.$$

We estimate the the class prior $P(Y)$ from the data (i.e. 0.5 for each class). We estimate the likelihood of the data given the class based on the relative frequencies in the data (i.e. $p(X_p \mid \text{Spam}) = \frac{3}{4}$). Since we only care about the class with the maximal probability, we can use Bayes rule without computing the denominator.

This gives us:

$$\begin{aligned}
p(\text{Spam} \mid X_p = 1, X_m = 1) &\propto p(X_p = 1 \mid \text{Spam})\ p(X_m = 1 \mid \text{Spam})\ p(\text{Spam}) \\
&= \frac{3}{4}\frac{1}{4}\frac{1}{2} = \frac{3}{32} \\
p(\text{Ham} \mid X_p = 1, X_m = 1) &\propto p(X_p = 1 \mid \text{Ham})\ p(X_m = 1 \mid \text{Ham})\ p(\text{Ham}) \\
&= \frac{1}{4}\frac{2}{4}\frac{1}{2} = \frac{2}{32}
\end{aligned}$$

Spam gives a higher value (these are not probabilities, just values proportional to the true probabilities), so the classifier classifies the email as Spam.

question 6: Which probabilities does the classifier assign to each class? For full class probabilities, we need to apply Bayes' theorem including its denominator:

$$p(Y \mid X_p, X_m) = \frac{p(X_p, X_m \mid Y)p(Y)}{p(X_p, X_m)} .$$

The denominator expands as

$$p(X_p, X_m) = \sum_{Y \in \{\text{Spam},\text{Ham}\}} p(X_p, X_m \mid Y)p(Y)$$

These are the two unnormalized probabilities ($\frac{3}{32}$ and $\frac{2}{32}$) that we've calculated already. Computing the proper probabilities is equivalent to normalizing the unnormalized ones. Thus, the class probabilities assigned are $\frac{3}{5}$ for Spam and $\frac{2}{5}$ for Ham.

To improve our accuracy, we add another feature:

"pill"	"meeting"	"hello"	label
T	F	F	Spam
T	F	F	Spam
F	T	F	Spam
T	F	F	Spam
F	F	F	Ham
F	F	F	Ham
F	T	F	Ham
T	T	T	Ham

question 7: For the class Spam, there are no emails recorded that contain the word "hello". Why is this a problem?

This makes our estimate of the probability $p(X_h = 1 \mid \text{Spam})$ zero. This causes the entire class probability to collapse to 0 even if the other features make Spam very likely.What solution is suggested in the slides?

To add *pseudo-observations*, so that for each feature each value is observed at least once for each class.

question 8: Implement this solution, and give the class probabilities for an email containing all three words.

After adding the pseudo observations, our dataset looks like this:

"pill"	"meeting"	"hello"	label
T	F	F	Spam
T	F	F	Spam
F	T	F	Spam
T	F	F	Spam
F	F	F	Ham
F	F	F	Ham
F	T	F	Ham
T	T	T	Ham
T	T	T	Spam
F	F	F	Spam
T	T	T	Ham
F	F	F	Ham

Note that we don't add every possible combination of features (that would be 8 extra instances per class). We just make sure that for every feature there is one email that has F for that feature and Spam as a class, one email that has T for that feature and Spam as a class and the same for Ham.

This gives us:

$$\begin{aligned}
&p(\text{Spam} \mid X_p = 1, X_m = 1, X_h = 1) \\
&\quad \propto p(X_p = 1 \mid \text{Spam})\, p(X_m = 1 \mid \text{Spam})\, p(X_h = 1 \mid \text{Spam})\, p(\text{Spam}) \\
&\quad = \frac{4}{6}\frac{2}{6}\frac{1}{6}\frac{1}{2} = \frac{8}{6^3 2} \\
&p(\text{Ham} \mid X_p = 1, X_m = 1, X_h = 1) \\
&\quad \propto p(X_p = 1 \mid \text{Ham})\, p(X_m = 1 \mid \text{Ham})\, p(X_h = 1 \mid \text{Ham})\, p(\text{Ham}) \\
&\quad = \frac{2}{6}\frac{3}{6}\frac{2}{6}\frac{1}{2} = \frac{12}{6^3 2}
\end{aligned}$$

Normalizing these gives us $\frac{2}{5}$ for Spam and $\frac{3}{5}$ for Ham.

3 Entropy

We define two probability distributions p and q on a set of four outcomes $\{a, b, c, d\}$.

$p(a)$	$p(b)$	$p(c)$	$p(d)$
$1/4$	$1/4$	$1/4$	$1/4$

$q(a)$	$q(b)$	$q(c)$	$q(d)$
$1/2$	$1/4$	$1/8$	$1/8$

question 9: Could you simulate sampling from these distributions using coinflips as described in the lecture?

Yes. p can be simulated by assigning each unique sequence of two coinflips to one of the outcomes. For q, we can assign the following sequences to each outcome:

$q(a)$	$q(b)$	$q(c)$	$q(d)$
H	TH	TTH	TTT

question 10: Compute the entropy of p and q.

$$\begin{aligned} H(p) &= -\sum_{y \in \{a,b,c,d\}} p(y) \log_2 p(y) \\ &= -4\left(\frac{1}{4}\log\frac{1}{4}\right) = -\log 4 \times -1 = \log 4 = 2 \end{aligned}$$

$$\begin{aligned} H(q) &= -\sum_{y \in \{a,b,c,d\}} q(y) \log_2 q(y) \\ &= -\frac{1}{2}\log\frac{1}{2} - \frac{1}{4}\log\frac{1}{4} - \frac{1}{8}\log\frac{1}{8} - \frac{1}{8}\log\frac{1}{8} \\ &= \frac{1}{2} + \frac{2}{4} + \frac{3}{8} + \frac{3}{8} = \frac{4}{8} + \frac{4}{8} + \frac{3}{8} + \frac{3}{8} = 1.75 \end{aligned}$$

The entropy of q is lower than the entropy of p. What does this tell you about the difference between the two distributions?

This means that p is more *uniform* than q. In other words, we have more *information* about what the outcome of a sample from q will be than we do about the outcome of a sample from p.

question 11: In the definition of entropy that we use (information entropy), the logarithms have base 2 (i.e. $\log_2$ instead of $\log_{10}$ or $\ln$). This follows directly from our decision to model probability distributions with coinflips. How?

The logarithm is defined as the expected code length under an idealized optimal binary code. If a binary code assigns a codeword of length of L to an outcome, the probability of sampling that outcome by generating a random code by flipping a coin is $\left(\frac{1}{2}\right)^L = 2^-L$. If we reverse this, to find the optimal code belonging to an event with probability $p(x)$, we get $L = -\log_2 p(x)$

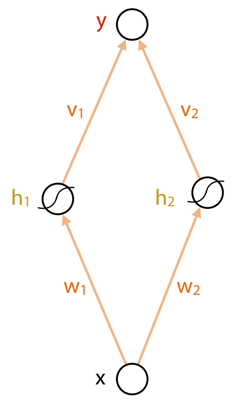
4 Backpropagation

We will practice the backpropagation algorithm as described in the slides. We will use a neural network defined by the following function:

$$\begin{aligned} y &= v_1 h_1 + v_2 h2 \\ h_1 &= \sigma(k_1) \\ h_2 &= \sigma(k_2) \\ k_1 &= w_1 x \\ k_2 &= w_2 x \end{aligned}$$

Where σ represents the logistic sigmoid. The network has a single input node x and a single output node y. The weights are w_1, w_2, v_1 and v_2. We've left out bias nodes to keep things simple.

question 12: Draw this network.



We will use stochastic gradient descent to train this network. That means we define the loss function for a single instance. We will use basic least-squares loss to train this network for regression. Our loss function for instance x is

$$\text{loss}_x(w_1, w_2, v_1, v_2) = \frac{1}{2}(y - t)^2$$

where t is the target value provided by the dataset, and y is the output of the network.

We see each line in the definition above as a module, with some inputs and some outputs. Using the chain rule, we will express the derivative with respect to weight w_1. We will use only *local derivatives*, expressing the derivative for each module with respect to its inputs, but not working out the derivative beyond that.

question 13: Fill in the gaps:

$$\begin{aligned}\frac{\partial \text{loss}}{\partial w_1} &= \frac{\partial \text{loss}}{\partial y}\frac{\partial y}{\partial h_1}\frac{\partial h_1}{\partial k_1}\frac{\partial k_1}{\partial w_1} \\ &= (y - t) \times v_1 \times \sigma(k_1)(1 - \sigma(k_1)) \times x \qquad (1)\end{aligned}$$

question 14: The network has a diamond shape, just as shown in the slide when explaining the multivariate chain rule (in the *Deep Learning 1* lecture). However, in this case, we don't need the multivariate chain rule. Why not? In what kind of situation would the multivariate chain rule be required?

The multivariate chain rule is needed when the output depends on one of the variables for which we are taking the derivative (like w_1) along multiple paths in the computation graph. In this case we have a diamond because y depends on x along two paths, but we never take the derivative with respect to x (the input), only with respect to the weights.

In a way, we *are* using the multivariate chain rule, if we write

$$\frac{\partial l}{\partial w_1} = \frac{\partial v_1 h_1}{\partial w_1} + \frac{\partial v_2 h_2}{\partial w_1}$$

(because the sum rule is an instance of the multivariate chain rule) but the second term becomes zero because the numerator is constant with respect to w_1.

The multivariate chain rule comes into effect when the output depends on a *weight* along multiple paths in the computation graph. This happens, for instance, when we compute the loss function over a batch of multiple data points.

We could take the formulation above and fill in the symbolic expressions for y, k_1, etc, and come up with a general symbolic formula for the derivative for all inputs. However, that is usually expensive to do for large neural nets. Instead, we leave it as is, and fill in the *numeric* values for these variables for a specific input and for specific weights.

question 15: Assume that the input is $x = 1$, with target output $t = \frac{1}{2}$ and that all weights are set at 1. Do a *forward pass*: compute the loss and all intermediate values k_1, k_2, h_1, h_2 and y.

To simplify calculation, you can use the approximation $\sigma(1) = \frac{3}{4}$.

$$\begin{aligned} k_1 &= 1 \\ k_2 &= 1 \\ h_1 &= {}^3\!/_4 \\ h_2 &= {}^3\!/_4 \\ y &= {}^3\!/_2 \\ \text{loss} &= \frac{1}{2} \end{aligned}$$

Now we do the *backward pass*. Fill these intermediate values in to the loss function decomposed by the chain rule from line 1, and compute the derivative with respect to w_1 for these inputs and weights.

$$\frac{\partial \text{loss}}{\partial w_1} = 1 \times 1 \times \frac{3}{4}\frac{1}{4} \times 1 = \frac{3}{16}$$