School of Computing and Information Systems
The University of Melbourne
COMP90049 Introduction to Machine Learning (Semester 1, 2021)

Week 4: Sample Solutions

1. What is optimisation? What is a "loss function"?

   In the context of Machine Learning, optimisation means finding the **optimal parameters** of the model that give us the most accurate results (predictions).

   To find the best possible results, optimisation usually involves minimising (the error) or maximising (the correct answers). Again, in the context of Machine Learning, most of the optimisation problems are described in terms of **cost** (i.e., error). We want to minimise undesirable outcomes (errors). To do so, we define a function that best describes our *undesirable outcomes* for each model. This function is called a *cost function* or a *loss function*.

2. Given the following dataset, build a Naïve Bayes model for the given training instances.

ID	*Outl*	*Temp*	*Humi*	*Wind*	PLAY
A	s	h	n	F	N
B	s	h	h	T	N
C	o	h	h	F	Y
D	r	m	h	F	Y
E	r	c	n	F	Y
F	r	c	n	T	N
G	o	m	n	T	?
H	?	h	?	F	?

   A Naïve Bayes model is probabilistic classification Model. All we need for building a Naive Bayes model is to calculate the right probabilities (Prior and Conditional).

   For this dataset, our class (or label or variable we trying to predict) is `PLAY`. So, we need the probability of each label (the prior probabilities):

   $P(\text{Play} = Y) = \frac{1}{2}$ $P(\text{Play} = N) = \frac{1}{2}$

   We also need to identify all the conditional probabilities between the labels of class (`PLAY`) and all the other attribute values such as `s`, `o`, `r` (for `Outlook`) or `h`, `m`, `c` (for `Temp`) and so on:

   $P(\text{Outl} = s \mid N) = \frac{2}{3}$ $P(\text{Outl} = o \mid N) = 0$ $P(\text{Outl} = r \mid N) = \frac{1}{3}$

   $P(\text{Outl} = s \mid Y) = 0$ $P(\text{Outl} = o \mid Y) = \frac{1}{3}$ $P(\text{Outl} = r \mid Y) = \frac{2}{3}$

   $P(\text{Temp} = h \mid N) = \frac{2}{3}$ $P(\text{Temp} = m \mid N) = 0$ $P(\text{Temp} = c \mid N) = \frac{1}{3}$

   $P(\text{Temp} = h \mid Y) = \frac{1}{3}$ $P(\text{Temp} = m \mid Y) = \frac{1}{3}$ $P(\text{Temp} = c \mid Y) = \frac{1}{3}$

   $P(\text{Humi} = n \mid N) = \frac{2}{3}$ $P(\text{Humi} = h \mid N) = \frac{1}{3}$

   $P(\text{Humi} = n \mid Y) = \frac{1}{3}$ $P(\text{Humi} = h \mid Y) = \frac{2}{3}$

P(Wind = T | N) = $\frac{2}{3}$ P(Wind = F | N) = $\frac{1}{3}$

P(Wind = T | Y) = 0 P(Wind = F | Y) = 1

3. Using the Naïve Bayes model that you developed in question 2, classify the given test instances.

(i). No smoothing.

For instance **G,** we have the following:

$$N: \quad P(N) \times P(Outl = o \mid N)\, P(Temp = m \mid N) P(Humi = n \mid N)\, P(Wind = T \mid N)$$

$$= \frac{1}{2} \times 0 \times 0 \times \frac{2}{3} \times \frac{2}{3} = 0$$

$$Y: \quad P(Y) \times P(Outl = o \mid Y)\, P(Temp = m \mid Y) P(Humi = n \mid Y)\, P(Wind = T \mid Y)$$

$$= \frac{1}{2} \times \frac{1}{3} \times \frac{1}{3} \times \frac{1}{3} \times 0 = 0$$

To find the label we need to compare the results for the two tested labels (`Y` and `N`) and find the one that has a higher likelihood (Maximum Likelihood Estimation).

$$\hat{y} = argmax_{y \in \{Y,N\}}\, P(y|T = G)$$

However, based on these calculations we find that both values are 0! So, our model is unable to predict any label for test instance G.

The fact is as long as there is a single 0 in our probabilities, none of the other probabilities in the product really matter.

For **H**, we first observe that the attribute values for `Outl` and `Humi` are missing (?). In Naive Bayes, this just means that we calculate the product without those attributes:

$$N: \quad P(N) \times P(Outl = ? \mid N)\, P(Temp = h \mid N) P(Humi = ? \mid N)\, P(Wind = F \mid N)$$

$$\approx P(N) \times P(Temp = h \mid N) \times P(Wind = F \mid N)$$

$$= \frac{1}{2} \times \frac{2}{3} \times \frac{1}{3} = \frac{1}{9}$$

$$Y: \quad P(Y) \times P(Outl = ? \mid Y)\, P(Temp = h \mid Y) P(Humi = ? \mid Y)\, P(Wind = F \mid Y)$$

$$\approx P(Y) \times P(Temp = h \mid Y) \times P(Wind = F \mid Y)$$

$$= \frac{1}{2} \times \frac{1}{3} \times 1 = \frac{1}{6}$$

Therefore, the result of our argmax function for the test instance **H** is **`Y`**.

$$argmax_{y \in \{Y,N\}}\, P(y|T = H) = Y$$

(ii). Using the "epsilon" smoothing method.

For test instance G, using the 'epsilon' smoothing method, we can simply replace the 0 values with a small positive constant (like 10^{-6}), that we call ε. So we'll have:

$$N: \quad = \frac{1}{2} \times \varepsilon \times \varepsilon \times \frac{2}{3} \times \frac{2}{3} = \frac{2\varepsilon^2}{9}$$

$$Y: \quad = \frac{1}{2} \times \frac{1}{3} \times \frac{1}{3} \times \frac{1}{3} \times \varepsilon = \frac{\varepsilon}{54}$$

By smoothing, we can sensibly compare the values. Because of the convention of ε being very small (it should be (substantially) less than $\frac{1}{6}$ (*6 is the number of training instances*)), Y has the greater score (higher likelihood). So Y is the output of our **argmax** function and **G is classified as Y.**

A quick note on the 'epsilons':

This isn't a serious smoothing method, but does allow us to sensibly deal with two common cases:

- Where two classes have the same number of 0s in the product, we essentially ignore the 0s.
- Where one class has fewer 0s, that class is preferred.

For **H**, we don't have any zero probability, so the calculations are similar to when we had no smoothing:

$$N: \quad P(N) \times P(Temp = h \mid N)\, P(Wind = F \mid N)$$

$$\approx P(N) \times P(Temp = h \mid N)\, P(Wind = F \mid N)$$

$$= \frac{1}{2} \times \frac{2}{3} \times \frac{1}{3} = \frac{1}{9} \cong 0.1$$

$$Y: \quad P(Y) \times P(Temp = h \mid Y)\, P(Wind = F \mid Y)$$

$$\approx P(Y) \times P(Temp = h \mid Y)\, P(Wind = F \mid Y)$$

$$= \frac{1}{2} \times \frac{1}{3} \times \frac{3}{3} = \frac{1}{6} \cong 0.16$$

Therefore, the result of our argmax function for the test instance **H** is **Y**.

$$argmax_{y \in \{Y,N\}}\, P(y|T = H) = Y$$

(iii). Using "Laplace" smoothing ($\alpha = 1$)

This is similar, but rather than simply changing the probabilities that we have estimated to be equal to 0, we are going to modify the way in which we estimate a conditional probability:

$$P_i = \frac{x_i + \alpha}{N + \alpha d}$$

In this method we add α, which is 1 here, to all possible event (seen and unseen) for each attribute. So, all unseen event (that currently have the probability of 0) will receive a count of 1 and the count for all seen events will be increased by 1 to ensure that the monocity is maintained.

For example, for the attribute `Outl` that have 3 different values (`s`, `o`, and `r`). Before, we estimated *P(*`Outl` *=*`o`*|*`Y`*)* = $\frac{1}{3}$ before; now, we add 1 to the numerator (add 1 to the count of `o`), and 3 to the denominator (1 (for `o`) + 1 (for `r`)+ 1 (for `s`)). So now *P(*`Outl` *=* `o`*|* `Y`*)* have the estimate of $\frac{1+1}{3+3} = \frac{2}{6}$.

In another example, *P(Wind = T| Y)* is not presented (unseen) in our training dataset (*P(Wind = T| Y)* = $\frac{0}{3}$). Using the Laplace smoothing (α =1), we add 1 to the count of Wind = T (given Play = Y) and 1 to the count of Wind = F (given Play = Y) and so now we have *P(Wind = T| Y)* = $\frac{0+1}{3+2} = \frac{1}{5}$.

Typically, we would apply this smoothing process when building the model, and then substitute in the Laplace-smoothed values when making the predictions. For brevity, though, I'll make the smoothing corrections in the prediction step.

For G, this will look like:

$$N: \quad P(N) \times P(Outl = o \mid N)\, P(Temp = m \mid N) P(Humi = n \mid N)\, P(Wind = T \mid N)$$

$$= \frac{1}{2} \times \frac{0+1}{3+3} \times \frac{0+1}{3+3} \times \frac{2+1}{3+2} \times \frac{2+1}{3+2}$$

$$= \frac{1}{2} \times \frac{1}{6} \times \frac{1}{6} \times \frac{3}{5} \times \frac{3}{5} = 0.005$$

$$Y: \quad P(Y) \times P(Outl = o \mid Y)\, P(Temp = m \mid Y) P(Humi = n \mid Y)\, P(Wind = T \mid Y)$$

$$= \frac{1}{2} \times \frac{1+1}{3+3} \times \frac{1+1}{3+3} \times \frac{1+1}{3+2} \times \frac{0+1}{3+2}$$

$$= \frac{1}{2} \times \frac{2}{6} \times \frac{2}{6} \times \frac{2}{5} \times \frac{1}{5} \cong 0.0044$$

Unlike with the epsilon procedure, N has the greater score — even though there are two attribute values that have never occurred with N. So here **G is classified as N.**

For H:

$$N: \quad = \frac{1}{2} \times \frac{2+1}{3+3} \times \frac{1+1}{3+2} = 0.1$$

$$Y: \quad = \frac{1}{2} \times \frac{1+1}{3+3} \times \frac{3+1}{3+2} \cong 0.13$$

Here, Y has a higher score — which is the same as with the other method, which doesn't do any smoothing here — but this time it is only slightly higher.

4. For the following set of classification problems, design a Naive Bayes classification model. Answer the following questions for each problem: (1) what are the instances, what are the features (and values)? (2) explain which distributions you would choose to model the observations, and (3) explain the significance of the Naive Bayes assumption.

(i). You want to classify a set of images of animals in to 'cats', 'dogs', and 'others'.

(1) Here the images are the instances, and the features are the pixels of the image. Each pixel can have values such as pixel intensity or colour code or shade. The important notice here is that these values (in the context of image processing) are continues.

(2) Since our features are continuous, the Gaussian (or normal) distribution is most appropriate (assuming that our feature values are (roughly) Gaussian distributed). The Gaussian distribution has a bell shape curve and useful features that make the calculations fairly easy.

(3) The Naïve Bayes assumption tells us that given each class ('cat', 'dog', 'others'), we treat all features as independent. But the reality is that this assumption is not true at all. In fact clearly the {intensity, colour, ...} of neighbouring pixels depend on one another. However, we can still use Naïve Bayes for developing a model and predicting the labels.

(ii). You want to classify whether each customer will purchase a product, given all the products (s)he has bought previously.

(1) In this problem, each customer can be used as an instance. The features can be the products (or types of products) in the catalogue. The value for these features can be 0 and 1 as an indicator of whether the customer has purchased the product; values = 0/1 (did or did not purchase), or perhaps counts of how many times the customer bought a specific (type of) product.

(2) In this setting, the features are discrete. If we assume count-based features, we define a Multinomial distribution over K dimensions (K=number of products) and values are the counts of purchases of that particular customer of each product; we can use essentially the same approach using binary indicators (leading to the Binomial distribution). The binomial distribution with parameters n and p is the discrete probability distribution of the number of successes in a sequence of n independent Boolean experiments (with the probability of p for success in each experiment).

(3) Here the NB assumptions tell us that given the label ('purchase', 'not purchase') all previous purchases are treated as independent features. But clearly, this is not the case (e.g., if a customer purchased Game of Thrones seasons 1-5, it should influence the probability of the customer also having purchased Game of Thrones season 6).