Naïve Bayes

NAME:

WITH HELP FROM:

Content Learning Objectives

After completing this activity, students should be able to:

- Explain conditional independence and its role in naïve bayes
- compute estimated priors for discrete features/dimensions
- compute estimated priors for continuous features
- hand compute probabilities for simple problem
- list the pros/cons of this classifier versus k nearest neighbors

Generative Setup

Naïve bayes is a generative model (versus KNN and decision trees which are discriminative models). Discriminative models are built so that, given a set of features/dimensions X (training data), they predict a class label y. A generative model starts by asking, if someone told me the class label, can I describe the features. That is, given y, can we learn the probabilities (a description) of each feature. When thinking about this in terms of probabilities, we are trying to compute:$P(X|y)$ To use this model as a classifier, we need to predict y, as in $P(y|X)$. Fortunately, due to bayes rule, we can rewrite this as:

$$P(y|X) = \frac{P(X|y)P(y)}{P(X)} \tag{1}$$

$P(X|y)$ and $P(y)$ are *estimated* using the training data, and predictions are made using the right hand side of equation 1. So, in classification, we want:

$$\hat{y} = \underset{y \in Y}{\operatorname{argmax}} P(y|x) \tag{2}$$

The $\hat{y}$ symbol is the estimate (predicted value) of y (the real class). This equation says, try each value of y and whichever value maximizes the probability, pick that as our prediction $\hat{y}$.
NOTICE that as y varies, the denominator in equation 1 stays the same (in other words, P(X) is the same regardless of the value selected for y). Thus, we can avoid computing P(X), which is extremely convenient. The new equation is written as:

$$P(y|x) \propto P(x|y)P(y) \tag{3}$$

The $\propto$ symbol means "proportional to". The purpose of the denominator ($P(X)$) in Eq. 1 is to make it so that the resulting value is still a probability (with values between 0 and 1, sometimes this value is called the normalizing constant). Equation 3 does not return a real probability, but since we only need to pick the class with produces the highest value, this is OK for this application.

Outlook	Temp	Humidity	Play (Y/N)
sunny	hot	high	no
sunny	hot	high	no
rainy	cool	normal	no
overcast	hot	high	yes
rainy	mild	high	yes
rainy	cool	normal	yes
overcast	cool	normal	yes

Figure 1: A 3 dimensional dataset where the class label is it is desirable to go outside to play.

Estimating Probabilities for Features

The training data will allow us to estimate $P(X|y)$. Here is some example data.

To calculate the $P(X|y)$, we first need to recognize the "short-hand" employed. This is really 2 probabilities:

- $P(X|y = yes)$
- $P(X|y = no)$

Lets work with the **yes** class. We would need to compute the following probabilities for each of the potential values of X. Fortunately, X is *discrete*, so we can enumerate all of them.

- $P(X = \{sunny, hot, high\} \mid y = yes)$
- $P(X = \{sunny, hot, normal\} \mid y = yes)$
- $P(X = \{sunny, mild, high\} \mid y = yes)$
- $P(X = \{sunny, mild, normal\} \mid y = yes)$
- ...
- $P(X = \{overcast, cool, normal\} \mid y = yes)$

In all, we would have 18 probabilities ($3 \times 3 \times 2$), and another 18 for when $y = 0$. Imagine if each feature could take k different values, then we would need k^d probabilities for each possible class label y.

1. (10 points) List two challenges with this setup.

Conditional Independence

The following probabilities would model the features in all cases. However, if the features were *conditionally independent* from one another, it would simplify the problem.

$$P(y|X) = \frac{P(X|y)P(y)}{P(X)} \propto P(y) \prod_{i=1}^{d} P(X_i|y) \tag{4}$$

Assuming conditional independence between the features is why this is called **naïve.** Notice we only need d probabilities for each class.

Class Labels

For discrete dimensions (like the ones in Table 1), we can simply use the frequencies of the values. Lets start with estimating $P(y)$ for the data in Table 1 . The examples data shows 4 examples when play is yes and 3 for no. This yields:

- $P(y = yes) = \frac{4}{7} \approx 0.57$
- $P(y = no) = \frac{3}{7} \approx 0.43$

So, for each dimension, we would have $P(X_i = c|y) = \frac{n_c}{n}$ where $\sum_c P(X_i = c|y) = 1$, where n_c is the count of training entries for dimension i that have value c and n is the total number of training examples. For a given class label y, the sum of these probabilities should be 1 (across all the potential values for the dimension c).

2. (10 points) Compute the following probabilities:

$P(x_1 = sunny\|y = no)$	$\frac{2}{3} \approx 0.67$
$P(x_1 = overcast\|y = no)$	$\frac{0}{3} = 0$
$P(x_1 = rainy\|y = no)$	
$P(x_2 = hot\|y = no)$	
$P(x_2 = mild\|y = no)$	
$P(x_2 = cool\|y = no)$	
$P(x_3 = high\|y = no)$	
$P(x_3 = normal\|y = no)$	

$P(x_1 = sunny\|y = yes)$	
$P(x_1 = overcast\|y = yes)$	$\frac{2}{4} = 0.5$
$P(x_1 = rainy\|y = yes)$	
$P(x_2 = hot\|y = yes)$	
$P(x_2 = mild\|y = yes)$	
$P(x_2 = cool\|y = yes)$	
$P(x_3 = high\|y = yes)$	
$P(x_3 = normal\|y = yes)$	

To Play or Not to Play

Now that we have some probabilities, lets make a prediction. Given $x = \{rainy, cool, normal\}$, will the person play?

3. (10 points) Calculate

   - $P(x_1 = rainy|y = yes)P(x_2 = cool|y = yes)P(x_3 = normal|y = yes)P(y = yes)$
   - $P(x_1 = rainy|y = no)P(x_2 = cool|y = no)P(x_3 = normal|y = no)P(y = no)$

   Which one is higher? The higher value is our prediction.

4. (10 points) Utilize naïve bayes to predict the class label (y = yes or y = no) for the data x = {sunny, mild, normal}? Write out the equations first ($P(x_1 = ...)$) then write your answer showing each value corresponding to each value in the product. Do we see something bad happening here?

Handing Zero Probabilities

When one of the probabilities is 0, it causes the overall probability to be zero. To prevent this, Laplace smoothing can be utilized (see section Handling Zero Conditional Probabilities in section 4.4.2 in the textbook). Our new estimate for each class is:

$$P(x_i = c|y) = \frac{n_c + 1}{n + v} \tag{5}$$

where $c \in C_i$ (and C is the set of values this discrete dimension (X_i) can take), and v is the number of different values dimension X_i can take . This prevents any single probability from being zero.

5. (10 points) Compute the following probabilities, incorporating Laplace smoothing.

$P(x_1 = sunny\|y = no)$	$\frac{2+1}{3+3} = \frac{3}{6} = 0.5$
$P(x_1 = overcast\|y = no)$	$\frac{0+1}{3+3} = \frac{1}{6}$
$P(x_1 = rainy\|y = no)$	
$P(x_2 = hot\|y = no)$	
$P(x_2 = mild\|y = no)$	
$P(x_2 = cool\|y = no)$	
$P(x_3 = high\|y = no)$	
$P(x_3 = normal\|y = no)$	

$P(x_1 = sunny\|y = yes)$	
$P(x_1 = overcast\|y = yes)$	$\frac{2+1}{4+3} = \frac{3}{7}$
$P(x_1 = rainy\|y = yes)$	
$P(x_2 = hot\|y = yes)$	
$P(x_2 = mild\|y = yes)$	
$P(x_2 = cool\|y = yes)$	
$P(x_3 = high\|y = yes)$	
$P(x_3 = normal\|y = yes)$	

6. (10 points) Rework question 4.

Estimating Continuous Attributes

Compute probabilities for discrete dimensions can be accomplished with counting. However, what if one of our dimensions is continuous? To handle that, we will pick a continuous distribution to model the dimension. Usually, this is a Gaussian, or normal distribution. The two parameters of a Gaussian are the mean (or average), represented by μ, and the standard deviation, represented by σ. For each continuous attribute, we simply calculate these numbers using the training data. Recall that computing the standard deviation uses the following formula:

$$s = \sqrt{\frac{\sum_{i=1}^{N}(x_i - \bar{x})^2)}{N-1}} \tag{6}$$

Where $\bar{x}$ is the average (or μ). Now, given a value for this dimension, we can compute an estimate of its probability using the following formula:

$$f(x|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \tag{7}$$

For example, if $\mu = 2.4$ and $\sigma = 1.2$, lets calculate out an estimate of the probability for the value 2.1. Recall that *exp(x)* is just a function that raises e to the x power (e^x).

$$f(x = 2.1|\mu = 2.4, \sigma = 1.2) = \frac{1}{\sqrt{2\pi \times 1.2^2}} exp\left(-\frac{(2.1-2.4)^2}{2 \times (1.2^2)}\right)$$

$$= \frac{1}{\sqrt{9.0432}}\ exp\left(-\frac{0.09}{2.88}\right)$$

$$= \frac{1}{3}\ exp\left(-\frac{0.09}{2.88}\right)$$

$$= 0.33\ exp(-0.03125) \approx 0.32$$

In python, you can use the following code to do this:

```
import scipy
from scipy import stats
scipy.stats.norm(2.4,1.2).pdf(2.1)
```

7. (10 points) Given the following data for a dimension and using a Gaussian to model the dimension, compute the proportional probability for the value 1.9. The data is: {2.3, 1.9, 0.3, 2.9, 3.1, 2.5}.

Multiplying very small numbers

When working with lots of dimensions, it is very possible that we will be multiplying a large set of numbers, which are all less than 1, and some that are very close to zero. This causes numerical instabilities. To solve this problem, recall that $log(x)$ grow monotonically with x. That is, as x increases, the log(x) increases. Logs have the nice property where:

$$log(ab) = log(a) + log(b) \tag{8}$$

So, we can transform our original equation to a sum as shown below:

$$P(X, y) = \frac{P(X|y)P(y)}{P(X)} \propto P(y) \prod_{i=1}^{d} P(X_i|y)$$

$$log(P(X, y)) \propto log(P(y)) + \sum_{i=1}^{d} log(P(X_i|y)) \tag{9}$$

8. (10 points) Rework problem 3 using the sum of the logs. Do you still select the same class?