Homework 4

Hamsalekha Premkumar

Instructions: Although this is a programming homework, you only need to hand in a pdf answer file. There is no need to submit the latex source or any code. You can choose any programming language, as long as you implement the algorithm from scratch.
Use this latex file as a template to develop your homework. Submit your homework on time as a single pdf file to Canvas. Please check Piazza for updates about the homework.

1 Best Prediction Under 0-1 Loss (14 pts)

Suppose the world generates a single observation $x \sim \text{multinomial}(\theta)$, where the parameter vector $\theta = (\theta_1, \dots, \theta_k)$ with $\theta_i \geq 0$ and $\sum_{i=1}^k \theta_i = 1$. Note $x \in \{1, \dots, k\}$. You know θ and want to predict x. Call your prediction $\hat{x}$. What is your expected 0-1 loss:

$$\mathbb{E}\mathbb{1}[\hat{x} \neq x]$$

using the following two prediction strategies respectively? Prove your answer.
Strategy 1: $\hat{x} \in \arg\max_x \theta_x$, the outcome with the highest probability.

$$\mathbb{E}\mathbb{1}[\hat{x} \neq x] = 1 * P(\hat{x} \neq x) + 0 * P(\hat{x} = x) = P(\hat{x} \neq x)$$

Given $\hat{x} \in \arg\max_x \theta_x$, if θ_j is the maximum probability that corresponds to $\hat{x} = j$ where $j \in \{1, 2, 3, ...k\}$, then $P(\hat{x} \neq x) = P(x \neq j) = 1 - \theta_j$
Strategy 2: You mimic the world by generating a prediction $\hat{x} \sim \text{multinomial}(\theta)$. (Hint: your randomness and the world's randomness are independent)

$$\mathbb{E}\mathbb{1}[\hat{x} \neq x] = P(\hat{x} \neq x) = \sum_{a=1,b=1}^{K} P(\hat{x} = a, x = b)1[a \neq b]$$

$$1 - \sum_{a=1}^{K}\sum_{b=1}^{K} P(\hat{x} = a)P(\hat{x} = b)1(a = b)$$

For $a = b$,

$$1 - \sum_{a=1}^{K} P(\hat{x} = a)P(x = a) = 1 - \sum_{a=1}^{K} \theta_a^2$$

2 Best Prediction Under Different Misclassification Losses (12 pts)

Like in the previous question, the world generates a single observation $x \sim \text{multinomial}(\theta)$. Let $c_{ij} \geq 0$ denote the loss you incur, if $x = i$ but you predict $\hat{x} = j$, for $i, j \in \{1, \dots, k\}$. $c_{ii} = 0$ for all i. This is a way to generalize different costs on false positives vs false negatives from binary classification to multi-class classification. You want to minimize your expected loss:

$$\mathbb{E}c_{x\hat{x}}.$$

Derive your optimal prediction $\hat{x}$.

$$\mathbb{E}c_{x\hat{x}} = \sum_{j=1}^{K}\sum_{i=1}^{K} c_{ij}P(x = i)P(\hat{x} = j) * 1[i \neq j]$$

To minimize this loss, find a j that minimizes this equation. This is an optimization problem that takes the form:

$$argmin_j \sum_{j=1}^{K} \sum_{i=1}^{K} c_{ij}\theta_i$$

3 Language Identification with Naive Bayes (8 pts each)

Implement a character-based Naive Bayes classifier that classifies a document as English, Spanish, or Japanese - all written with the 26 lower case characters and space.
The dataset is languageID.tgz, unpack it. This dataset consists of 60 documents in English, Spanish and Japanese. The correct class label is the first character of the filename: $y \in \{e, j, s\}$.
We will be using a character-based multinomial Naïve Bayes model. You need to view each document as a bag of characters, including space. We have made sure that there are only 27 different types of printable characters (a to z, and space) – there may be additional control characters such as new-line, please ignore those. Your vocabulary will be these 27 character types. (Note: not word types!)

1. Use files 0.txt to 9.txt in each language as the training data. Estimate the prior probabilities $\hat{p}(y = e)$, $\hat{p}(y = j)$, $\hat{p}(y = s)$ using add-1 smoothing. Give the formula for add-1 smoothing in this case. Print the prior probabilities. (Hint: Store all probabilities here and below in $\log()$ internally to avoid underflow. This also means you need to do arithmetic in log-space. But answer questions with probability, not log probability.)

$$\hat{p}(y = e) = \frac{10+1}{30+3} = \frac{1}{3}$$
$$\hat{p}(y = j) = \frac{10+1}{30+3} = \frac{1}{3}$$
$$\hat{p}(y = s) = \frac{10+1}{30+3} = \frac{1}{3}$$

In general, prior probability estimation would be affected by sample bias. To overcome this, add-1 smoothing is applied to prior probabilities as well. Since in this case, equal number of documents from each class is considered, applying add-1 smoothing does not affect the prior probabilities. By definition, add-1 smoothening is given by,

$$p(x_i) = \frac{x_i + 1}{N + d}$$

In this context,
x_i is the number of train samples belonging to class i.(10)
N is the total number of train instances in all classes.(30)
d is the number of unique classes.(3)

2. Using the same training data, estimate the class conditional probability (multinomial parameter) for English

$$\theta_{i,e} := \hat{p}(c_i \mid y = e)$$

where c_i is the i-th character. That is, $c_1 = a, \ldots, c_{26} = z, c_{27} = space$. Again use add-1 smoothing. Give the formula for add-1 smoothing in this case. Print θ_e which is a vector with 27 elements.

```
class conditional probability for class,  e : [0.06014789 0.01115806 0.02152383 0.021986   0.10530833 0.0189489
 0.01749637 0.04720718 0.05539416 0.00145253 0.00376337 0.02898455
 0.02053347 0.05790308 0.06443946 0.0167701  0.00059422 0.05380959
 0.06615608 0.08008715 0.02667371 0.00930939 0.01551565 0.00118843
 0.01386505 0.00066024 0.1791232 ]
```

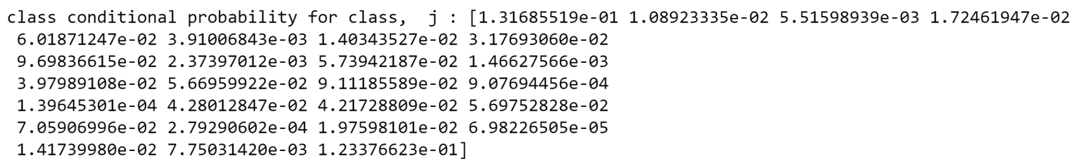
Fig: θ_e

Add-1 smoothening is given by,

$$\theta_{i,e} := \hat{p}(c_i \mid y = e) = \frac{n_{c_i,e} + 1}{N_e + V_e}$$

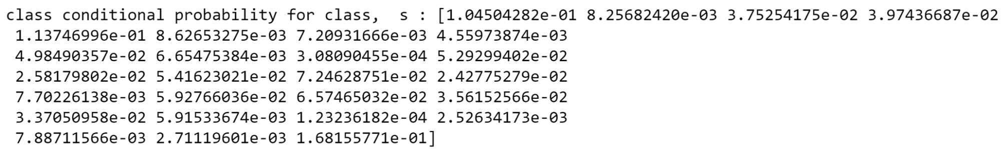
for $c_i \in \{a, \dots, z, space\}$ where,
$n_{c_i,e}$ = count of alphabet c_i in class e
N_e = count of all alphabets in class e
V_e = count of all unique alphabets in class e

3. Print θ_j, θ_s.



```
class conditional probability for class,  j : [1.31685519e-01 1.08923335e-02 5.51598939e-03 1.72461947e-02
 6.01871247e-02 3.91006843e-03 1.40343527e-02 3.17693060e-02
 9.69836615e-02 2.37397012e-03 5.73942187e-02 1.46627566e-03
 3.97989108e-02 5.66959922e-02 9.11185589e-02 9.07694456e-04
 1.39645301e-04 4.28012847e-02 4.21728809e-02 5.69752828e-02
 7.05906996e-02 2.79290602e-04 1.97598101e-02 6.98226505e-05
 1.41739980e-02 7.75031420e-03 1.23376623e-01]
```

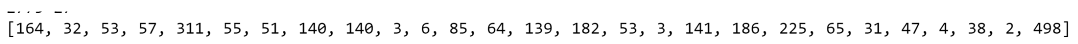
Fig: θ_j



```
class conditional probability for class,  s : [1.04504282e-01 8.25682420e-03 3.75254175e-02 3.97436687e-02
 1.13746996e-01 8.62653275e-03 7.20931666e-03 4.55973874e-03
 4.98490357e-02 6.65475384e-03 3.08090455e-04 5.29299402e-02
 2.58179802e-02 5.41623021e-02 7.24628751e-02 2.42775279e-02
 7.70226138e-03 5.92766036e-02 6.57465032e-02 3.56152566e-02
 3.37050958e-02 5.91533674e-03 1.23236182e-04 2.52634173e-03
 7.88711566e-03 2.71119601e-03 1.68155771e-01]
```

Fig: θ_s

4. Treat e10.txt as a test document x. Represent x as a bag-of-words count vector (Hint: the vocabulary has size 27). Print the bag-of-words vector x.



```
[164, 32, 53, 57, 311, 55, 51, 140, 140, 3, 6, 85, 64, 139, 182, 53, 3, 141, 186, 225, 65, 31, 47, 4, 38, 2, 498]
```

Fig: bag of characters count vector for e10.txt

5. Compute $\hat{p}(x \mid y)$ for $y = e, j, s$ under the multinomial model assumption, respectively. Use the formula

$$\hat{p}(x \mid y) = \prod_{i=1}^{d} \theta_{i,y}^{x_i}$$

where $x = (x_1, \dots, x_d)$. Show the three values: $\hat{p}(x \mid y = e), \hat{p}(x \mid y = j), \hat{p}(x \mid y = s)$. Hint: you may notice that we omitted the multinomial coefficient. This is ok for classification because it is a constant w.r.t. y.

$$\hat{p}(x \mid y = e) = e^{-7841.7863}$$
$$\hat{p}(x \mid y = j) = e^{-8759.3251}$$
$$\hat{p}(x \mid y = s) = e^{-8452.3831}$$

6. Use Bayes rule and your estimated prior and likelihood, compute the posterior $\hat{p}(y \mid x)$. Show the three values: $\hat{p}(y = e \mid x), \hat{p}(y = j \mid x), \hat{p}(y = s \mid x)$. Show the predicted class label of x.

WKT, $\hat{p}(y \mid x) = \frac{\hat{p}(x|y)*p(y)}{p(x)}$

$$\hat{p}(y = e \mid x) = \frac{e^{-7841.7863}}{e^{-7841.7863} + e^{-8759.3251} + e^{-8452.3831}}$$

$$\hat{p}(y = j \mid x) = \frac{e^{-8759.3251}}{e^{-7841.7863} + e^{-8759.3251} + e^{-8452.3831}}$$

$$\hat{p}(y = s \mid x) = \frac{e^{-8452.3831}}{e^{-7841.7863} + e^{-8759.3251} + e^{-8452.3831}}$$

To simplify calculations and avoid underflow issue, the maximum value of the three terms in the denominator is taken. i.e., $e^{-7841.7863}$

$$\hat{p}(y = e \mid x) = \frac{e^{-7841.7863}}{e^{-7841.7863}} \approx 1$$

$$\hat{p}(y = j \mid x) = \frac{e^{-8759.3251}}{e^{-7841.7863}} \approx e^{-917.5388}$$

$$\hat{p}(y = s \mid x) = \frac{e^{-8452.3831}}{e^{-7841.7863}} \approx e^{-610.5968}$$

The predicted class is 'e'

7. Evaluate the performance of your classifier on the test set (files 10.txt to 19.txt in three languages). Present the performance using a confusion matrix. A confusion matrix summarizes the types of errors your classifier makes, as shown in the table below. The columns are the true language a document is in, and the rows are the classified outcome of that document. The cells are the number of test documents in that situation. For example, the cell with row = English and column = Spanish contains the number of test documents that are really Spanish, but misclassified as English by your classifier.

	English	Spanish	Japanese
English	10	0	0
Spanish	0	10	0
Japanese	0	0	10

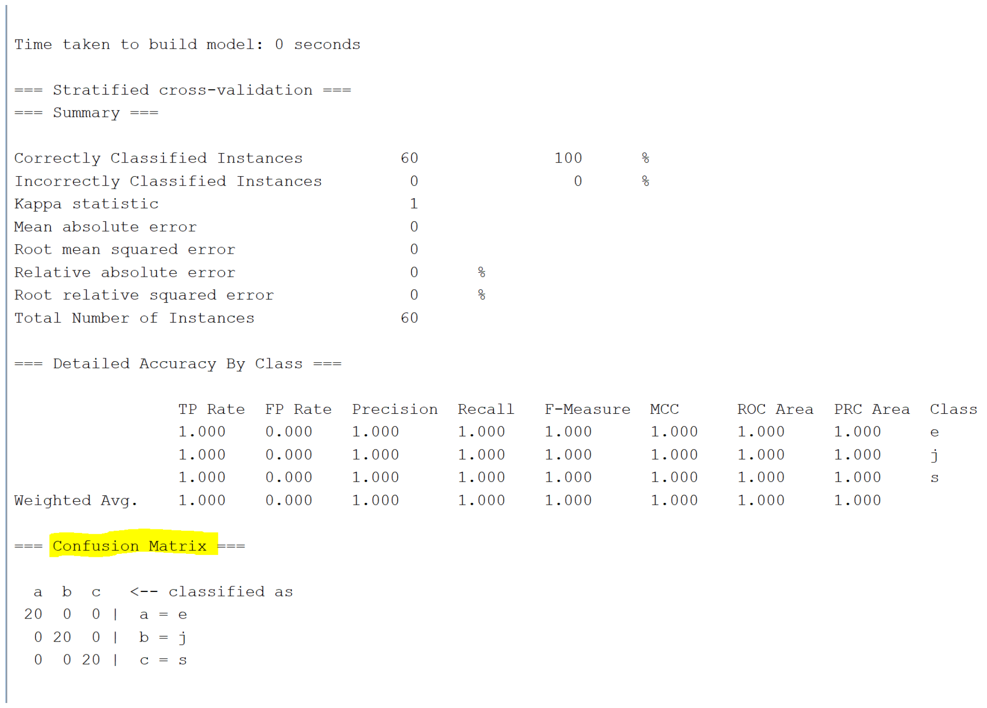
8. Take a test document. Arbitrarily shuffle the order of its characters so that the words (and spaces) are scrambled beyond human recognition. How does this shuffling affect your Naive Bayes classifier's prediction on this document? Explain the key mathematical step in the Naive Bayes model that justifies your answer. Shuffling the order of characters in the document does not affect the Naive Bayes classifier's prediction on the document as it is built on the foundations of probability / frequency occurrence of the characters and does not exploit how the characters interact with one another. Besides, the naive assumption is that every input value is entirely independent of every other feature, thus each contributing independently to the classification decision. This is formulated as :
$P(X_1, X_2, X_3, X_4, ..X_K, Y) = P(X_1, X_2, X_3, ...X_K|Y)P(Y) = \left(\prod_{k=1}^{K} P(X_k|Y)\right) P(Y).$
Thus shuffling does not affect the Classifier prediction

4 Weka (10 pts)

Perform multinomial Naive Bayes classification. Suggested key steps:

- We want you to experience Weka's TextDirectoryLoader. For this, you need to prepare our documents such that each character becomes a word. The easiest way is to insert a space between characters, but turn original space into the word "space". For example, the document "is the sun dying" should be turned into "i s space t h e space s u n space d y i n g". Then, create 3 subdirectories e, j, s and place each of the 20 documents in that language into the corresponding subdirectory.
- Find out how to ask Weka to use TextDirectoryLoader to load those 60 documents and use the subdirectory name as the class name. This happens in the Preprocess tab in Weka Explorer.
Once the documents are subdivided into folders based on the classes they belong to, use this command java weka.core.converters.TextDirectoryLoader -dir location/of/the/folder >
location/of/the/ combined/arff/file/with/filename Next step is to vectorize all the 60 documents by using StringToVector filter. This results in a file with documents as rows, characters as columns and values as numbers. This is a document vector with will contain 27 features and the classification attribute. We set the @@$class$@@ to be the class attribute and save the arff file. This arff file is now fed to the Bayes Multinomial Classifier which is set to perform 10 fold cross validation to produce confusion matrix.
- Apply Filter: filters/unsupervised/Attribute/StringtoWordVec. Click to see options, set outputWordCounts to True (otherwise Weka uses binary absence/presence features, which is less interesting for our purpose). You should see @@class@@ and 27 features.

- Choose Edit... A big document by feature table will show up, where you should see word counts roughly in the range 1–100. Right click on @@class@@, select “Attribute as class”. You may save it as an arff file. The tri-color histogram for different features is interesting.
- From the Classify tab, choose Classifier / bayes / NaiveBayesMultinomial
- Let Weka perform 10-fold cross validation. Report the resulting confusion matrix.



```
Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances          60              100      %
Incorrectly Classified Instances         0                0      %
Kappa statistic                          1
Mean absolute error                      0
Root mean squared error                  0
Relative absolute error                  0      %
Root relative squared error              0      %
Total Number of Instances               60

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 1.000    0.000    1.000      1.000    1.000      1.000    1.000     1.000     e
                 1.000    0.000    1.000      1.000    1.000      1.000    1.000     1.000     j
                 1.000    0.000    1.000      1.000    1.000      1.000    1.000     1.000     s
Weighted Avg.    1.000    0.000    1.000      1.000    1.000      1.000    1.000     1.000

=== Confusion Matrix ===

  a  b  c   <-- classified as
 20  0  0 |  a = e
  0 20  0 |  b = j
  0  0 20 |  c = s
```

Fig: Confusion Matrix