HOMEWORK 4

>>Haley Massa<<
>>9071903141<<

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 $x \sim$ multinomial(θ), where the parameter vector $\theta = (\theta_1, \ldots, \theta_k)$ with $\theta_i \geq 0$ and $\sum_{i=1}^{k} \theta_i = 1$. Note $x \in \{1, \ldots, k\}$.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: $\hat{x} \in_x \theta_x$, the outcome with the highest probability. When we are choosing the largest theta, we can calculate loss with:

$$\sum_{i=1}^{k} P(x = i)\mathbb{1}[\hat{x} \neq x]$$

$$= \sum_{i=1}^{k} \theta_i \mathbb{1}[\hat{x} \neq x]$$

$$= \sum_{i=1}^{k} \theta_i - \theta_{\hat{x}}$$

with the definitions above this means that

$$= 1 - \theta_{\hat{x}}$$

This is because the loss will be zero when they are equal and otherwise we want to sum up the loss between the two thetas. This will do that for us.

Strategy 2: You mimic the world by generating a prediction $\hat{x} \sim$ multinomial(θ). (Hint: your randomness and the world's randomness are independent)
Now if x is some sort of multinomial distribution we have to find the probability of more than just our original theta.

$$= \sum_{i=1}^{k} \sum_{j=1}^{k} P(x = i, \hat{x} = j)\mathbb{1}[\hat{x} \neq x]$$

$$= \sum_{i=1}^{k} \sum_{j=1}^{k} \theta_x \theta_{\hat{x}} \mathbb{1}[\hat{x} \neq x]$$

With probability laws we know that this is equal to

$$= 1 - \sum_{i=1}^{k} \sum_{j=1}^{k} \theta_x \theta_{\hat{x}} \mathbb{1}[\hat{x} == x]$$

Now we are saying that they need to be equal to one another. When x hat equals x that means :

$$= 1 - \sum_{i=1}^{k} \theta_i^2$$

2 Best Prediction Under Different Misclassification Losses (12 pts)

Like in the previous question, the world generates a single observation $x \sim$ multinomial(θ).Let $c_{ij} \geq 0$ denote the loss you incur, if $x = i$ but you predict $\hat{x} = j$, for $i, j \in \{1, \ldots, k\}$.$c_{ii} = 0$ 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 $\hat{x}$.

The optimal prediction, would result in a minimization of our cost function so :

$$\hat{x} = argmin\mathbb{E}c_{x\hat{x}}$$

$$= argmin \sum_{i=1}^{k} P(x = i)C_{i\hat{x}}$$

$$= argmin \sum_{i=1}^{k} \theta_i C_{i\hat{x}}$$

this is similar to problem one strategy one where we are predicting on one selected j. This chooses the smallest loss of all the x hats that are multiplied by their correct column of the cost matrix with the original 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 vieweach 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.)

   The formula for +1 smoothing for the Prior Probabilities

   N_c = Number of documents with class type

   N = Total Number of documents

   C = Number of class types

   $$P[Y = certainClass] = \frac{N_c + 1}{N + C}$$

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

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

Table 1: Prior Probabilities

y =	Prior Probability
'e'	0.333
's'	0.333
'j'	0.333

for $c \in \{a, \ldots, z, space\}$.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.

The formula for add-1 smoothing is :

$C_{c|class}$ = Total Number of the specific character in the given class

C = Total Number of characters in the class

V = Total number of characters in the vocabulary

$$p(c|y = class) = \frac{C_{c|class} + 1}{C + V}$$

This can be seen in table 2

Table 2: Conditonal Probabilities

Character	Theta E	Theta J	Theta S
' '	0.17912	0.12336	0.1681557
'a'	0.060148	0.13167	0.104504
'b'	0.011158	0.018915	0.00825682
'c'	0.021523	0.00515604	0.0375254
'd'	0.021986	0.017244	0.03974366
'e'	0.105308	0.0601829	0.11374699
'f'	0.018677	0.00390979	0.00862653
'g'	0.01724	0.014033	0.0072093166
'h'	0.04653	0.03176	0.00455973
'i'	0.054601	0.09697	0.049849
'j'	0.0014317	0.0002373804	0.00665475
'k'	0.003709	0.057390211	0.00030809
'l'	0.0285695	0.001466	0.0529299
'm'	0.0202394	0.039796	0.0258179
'n'	0.057074	0.056692	0.0541623
'o'	0.063516	0.0911219	0.0724628
'p'	0.01653	0.00090763	0.024277
'q'	0.00058571	0.000139635	0.00770226138
'r'	0.05303918	0.042798	0.059766
's'	0.0652089	0.0421699	0.0657465
't'	0.07894	0.0569713	0.0356152566
'u'	0.02629	0.07058577	0.03370509
'v'	0.0091761	0.0002792711	0.0059153367
'w'	0.01529351	0.0197584	0.00012323618
'x'	0.00117142	0.000069817	0.002526341
'y'	0.0136654	0.014173	0.007887115
'z'	0.00065079	0.00774977	0.002711196

3. Print θ_j, θ_s.

This can be seen in table 2

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.

With the first value representing space, and then continuing through the alphabet

$x = ([498., 164., 32., 53., 57., 311., 55., 51., 140., 140., 3., 6., 85., 64., 139., 182., 53., 3., 141., 186., 225., 65., 31., 47., 4., 38., 2.])$

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 $x = (x_1, \ldots, x_d)$.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.

Table 3: p(x $\mid y$)

p(x $\mid y =$)	Probability
'e'	2.26696E-3406
's'	1.5018E-3671
'j'	6.155786E-3805

6. Use Bayes rule and your estimated prior and likelihood, compute the posterior $(y \mid x)$.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.

In this problem, there were some questions on piazza regarding whether or not we would have to do the dividing by P(X). Since this is the Bayes rule, I used the formula

$$= \frac{P(y)P(x|y)}{P(x)}$$

The P(x) in the denominator is not needed in the algorithm to choose which one to pick. This is because since it is the same denominator, it is computationally more exhaustive for the same answer.

Table 4: p(y $\mid x$)

p(y $\mid x =$)	Probability
'e'	1.000000
's'	6.624E-266
'j'	2.715E-399

The predicted class is English, which is also the right one. yay

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 thetypes 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.

No this does not change our classifier. This is because we are using a bag of characters classifier. It does not take into account the proximity to other characters or character/word types when using a bag of methodology. This means that since we are not taking characters out, just mixing them around, the conditional probability for each character given each class will remain the same.

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.
- 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.

  The confusion matrix can be seen in the image below. As you can tell it worked very well, similar to my Bayes Algorithm so I believe I am on the right track with it.