Assigned: Tuesday 27 October

Due: Monday 2 November, 11:59 p.m.

Objectives:

- Practice **identifying** relevant rules of probability.
- **Apply** probabilistic inference for reasoning with uncertainty.
- Develop technical persuasive **writing** skills.

Collaboration: This laboratory will be completed in pairs assigned by the instructor.

1 Introduction

1.1 Background

In 1922, John Mills published *Letters of a Radio-Engineer to His Son* (New York: Harcourt, Brace and Company). In letter one, he writes (pp. 3–4):

> My Dear Son:
>
> You are interested in radio-telephony and want me to explain it to you. I'll do so in the shortest and easiest way which I can devise. The explanation will be the simplest which I can give and still make it possible for you to build and operate your own set and to understand the operation of the large commercial sets to which you will listen.
>
> I'll write you a series of letters which will contain only what is important in the radio of to-day and those ideas which seem necessary if you are to follow the rapid advances which radio is making. Some of the letters you will find to require a second reading and study. In the case of a few you might postpone a second reading until you have finished those which interest you most. I'll mark the letters to omit in this way.
>
> All the letters will be written just as I would talk to you, for I shall draw little sketches as I go along. One of them will tell you how to experiment for yourself. This will be the most interesting of all. You can find plenty of books to tell you how radio sets operate and what to do, but very few except some for advanced students tell you how to experiment for yourself.[1]

Clearly these are an important set of letters! To defend Mr. Mills's honor, please note the rest of this story is made up!

After hand writing these letters, Mr. Mills passed them off to his typist to transcribe. After a while he noticed the atrocious number of typographical errors his typist introduced. He then found another typist, but sadly the second was no better.

We have a few of the original letters composed that have no errors, along with the corresponding errorful typed letters by each typist. Only half of these letters are attributed to a particular typist, and we would like to recover which typist was responsible for each letter in the remaining half. (Maybe so we are sure to fire the one with the likely greater error rate.)

For task simplification, we remove all punctuation and make all letters lowercase so that we only consider a range of 28 characters (`a-z` plus space and newline). We also assume that typographical errors depend *only* on the character intended to be typed. We'll use the observed training data to calculate the following probabilities.

[1] http://www.gutenberg.org/ebooks/30688

1.2 Typist Model

Let

$$\begin{aligned} C &\equiv \text{Value of the actual character to type} \\ O &\equiv \text{Value of the observed typed character} \\ T &\equiv \text{The specific typist involved} \\ \mathbf{D} &\equiv \text{All pairs of actual and observed typed characters heretofore} \\ I &\equiv \text{All background information and assumptions above} \end{aligned}$$

then we calculate our belief for the next typo as

$$P\left(O = o \mid C = c, \mathbf{D} = \mathbf{d}, T = t, I\right) = \frac{n_D\left(c, o\right) + 1}{\sum_{o'} n_D\left(c, o'\right) + 28} \tag{1}$$

where $n_D\left(c, o\right)$ is the number of times we observed character c being typed as character o among the pairs in $\mathbf{D}$. Note that the denominator is derived from Laplace's rule of succession and simply ensures that the distribution sums properly to unity.

1.3 Data

We currently have a total of 12 letters available or study. All twelve of them have the corresponding corrupted and original versions, but only six identify the typist. These may be found in
`/home/weinman/courses/CSC261/data/mills`,
where corresponding file names (letter numbers) have the same name, and all are organized into a hopefully obvious directory hierarchy.

For this assignment, you will analyze letters 1, 8, and 16 by typist one, and 4, 9, and 18 by typist two. Letters 3, 7, 10, 11, 15, and 22 are unattributed and must be classified.

2 Programming

To help you in your analysis, the following procedures are provided in
`/home/weinman/courses/CSC261/probability/charmodel.scm`

2.1 Support Utilities

We need a structure to store the accumulated counts and an easy way of adding to them as we observe characters. The following procedures develop these structural tools.

`(create-counts num-chars initial-value)` produces a vector of length `num-chars` where each entry is another vector of length `num-chars` whose entries are all `initial-value`. This vector-of-vectors will act as the two dimensional array $n\left(\cdot, \cdot\right)$ used in the probability model above. To initialize a vector of counts with the single "pseudo-observation" indicated by the rule of succession, 1 should be given for *`initial-value`*.

`(char->index ch)` produces a number between 0 and 27 (inclusive) according to the following pattern

```
> (char->index #\a)
0
> (char->index #\b)
1
> (char->index #\z)
25
> (char->index #\space)
26
> (char->index #\newline)
27
```

It uses the fact that the characters `#\a` through `\#z` form a collating sequence and that white space should be encoded as index 26, but no other values are hard coded.

`(get-count-slice counts index)` returns the "slice" of counts for a particular character. That is, it gives us all values of $n(c, \cdot)$ for the character with *index* c.

`(get-count counts index-1 index-2)` returns the count value for $n(c_1, c_2)$.

`(increment-count! counts index-1 index-2)` has the side effect of incrementing $n(c_1, c_2)++$.

2.2 Probabilities

Using the utilities above, we provide the following procedures to learn and use $P(O = o \mid C = c, \mathbf{D} = \mathbf{d}, T = t, I)$.

`(count-conditionals! correct-filename corrupt-filename counts)` that takes the path to two files (one for the correct text and one for the text as corrupted by the typist—both should have the same number of characters) as well as a structure of the form produced by `(create-counts ...)` and adjusts the counts according to the number of times each correct character was observed as each corrupt character.

`(normalize-counts! counts)` normalizes each slice of *counts* so that it sums to one.

`(likelihood correct-filename corrupt-filename normalized-counts)` calculates the likelihood of seeing the corrupted file, given the correct file, for the particular typist's patterns represented by *normalized-counts*. That is,

$$P(\mathbf{O} = \mathbf{o} \mid \mathbf{C} = \mathbf{c}, \mathbf{D} = \mathbf{d}, T = t, I) = \prod_i P(O_i = o_i \mid C = c_i, \mathbf{D} = \mathbf{d}, T = t, I) \tag{2}$$

for all typed and true characters (o_i, c_i) in files *corrupt-filename* and *correct-filename*, respectively. ***Note:*** Because the likelihood is calculated exactly if the counts are exact, this procedure is prohibitively slow.

`(log-likelihood correct-filename corrupt-filename normalized-counts)` calculates the log-likelihood (base e) of seeing the corrupted file, given the correct file, for the particular typist's patterns represented by *normalized-counts*. That is,

$$\ln P(\mathbf{O} = \mathbf{o} \mid \mathbf{C} = \mathbf{c}, \mathbf{D} = \mathbf{d}, T = t, I) = \sum_i \ln P(O_i = o_i \mid C = c_i, D = d, T = t, I) \tag{3}$$

For speed, numbers are represented inexactly.

You will need to calculate the posterior $P(T = t \mid \mathbf{O} = \mathbf{o}, \mathbf{C} = \mathbf{c}, \mathbf{D} = \mathbf{d}, I)$ where $\mathbf{o}$ and $\mathbf{c}$ represent typed and true text from an unlabeled document using the likelihood given above and an appropriate prior.

To calculate Jaynes's evidence, recall the following two useful laws of logarithms:

$$\log_a x = \frac{\ln x}{\ln a} \tag{4}$$

$$\log_a \frac{p}{q} = \log_a p - \log_a q \tag{5}$$

3 Analysis

Imagine you are a computational historian trying to reconstruct the transcription record of Mr. Mills's typists. You will need to use probability to do some analysis of the data, convince your reader of the reliability of your conclusions, and ultimately identify the typists of some unlabeled manuscripts.

Write a persuasive report that briefly introduces the problem, describes the assumptions and method of analysis, clearly reports observations and results, and summarizes your analysis and conclusions.

To help you structure (and help the reader interpret) an analysis, you should address the following issues.

- Assess the likely reliability of your predictions. “Train” on a single letter from each typist and calculate Jaynes’s evidence for the typist of each of the other two labeled letters. (For example, train typist one’s model on letter 16 and typist two’s model on letter 18, then calculate the evidence on letter 4. There are 36 such ways to choose the two "training" letters and the one "testing" letter.) Would your resulting predictions be correct? How often? How strong is the evidence (that is, how confident should you be)? A one-dimensional “scatter” plot may be an effective visual aid; plot marker shape or color could indicate the true typist. See Figure 1 below as an example.

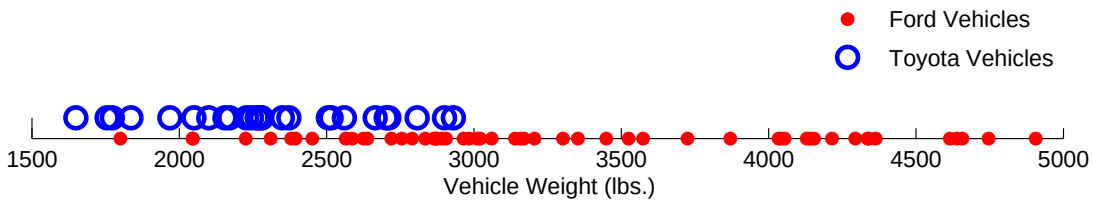

Figure 1: Vehicle weight versus manufacturer. (Source: http://lib.stat.cmu.edu/datasets/cars.data)

- Repeat that analysis by training on two letters from each typist and calculating the evidence for the typist of a held out letter (18 configurations total).
- Train on all available letters and give attribution (along with your data and reasoning) for the typist of the unlabeled letters. Considering your earlier results, how confident are you in each prediction? Why?

Note that while this report involves a slightly more technical form of writing than you may be accustomed to, the usual standards of composition apply (i.e., an introduction with a clearly stated thesis or question-answer pair, coherent paragraphs with clear topic sentences, a logically structured narrative, and a well-reasoned conclusion). Data should be easily read in tables or figures. In addition, any tables or figures should be captioned, titled, and labeled (i.e., with axes and/or legends) as appropriate.

Although no specific program form is required, good programming practice (legible formatting, code reuse, documentation, clear names, etc.) will be rewarded.

What to turn in

Your submission should include the following

- A `.scm` file(s) with your supporting analytical code (which should display all the raw data results)
- A single PDF containing (merged)
  - Your enscripted Scheme file(s)
  - A transcript of your program’s output
  - Your written report

  A missing PDF or files in any other format will receive a zero.