CSC 261 **Lab 8: Sequential Probabilistic Inference** **Spring 2020**

Assigned: Tuesday 31 March

Due: Monday 6 April, 10:30 p.m.

Objectives:

- Analyze **sequential inference** algorithms
- Investigate and assess **model quality**
- Develop technical persuasive **writing** skills

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

1 Introduction

As it turns out, some of Mr. Mill's letters are not only unattributed to a specific typist, but we also lack the original, uncorrupted text. Is it possible to still predict the typist? Could we recover the text? Yes, with the right model.

As we saw in chapter 15 of AIMA, dynamic Bayesian networks are used for reasoning about sequences of interrelated propositions. Our previous model of typographical errors not only allowed us to infer which typist produced a particular letter, assuming we knew the real text, but we could have also used Bayes rule to invert the likelihood of a particular error and give a prediction of the most probable true character if we *did not* know the true text. Unfortunately, such a prediction strategy would not have produced very satisfactory results, because most of the time the typists didn't make errors. Bayes rule would have us leave all characters unchanged.

Fortunately, there are statistical regularities not only in the errors made by each typist, but also in the English language. What character would you predict to follow "q"? What character would you predict to follow a "t"? In the same way we learned how to predict the character that would be typed by each person, we can predict the next character in a stream of English text. (Your answers should be "u" for obvious reasons, and "h" due to the frequent use of "the".)

A simple Bayesian structure for this problem looks like AIMA Figure 15.2. The "sensor model" now becomes our likelihood from the previous lab: $P\left(O = o \mid C = c, \mathbf{D} = \mathbf{d}, T = t, I\right)$, the probability a typist produces a particular observed character o given the character c to type. What is the transition model? To asses that, we formalize the simple language model suggested above.

1.1 Language Model

Let

$$\begin{aligned} C &\equiv \text{Value of the next character to appear} \\ C' &\equiv \text{Value of the previous character} \\ \mathbf{L} &\equiv \text{All known text of actual letters} \\ I &\equiv \text{All background information and assumptions above} \end{aligned}$$

then we shall calculate our belief for the next *actual* character as

$$P\left(C = c \mid C' = c', \mathbf{L} = \mathbf{l}, I\right) = \frac{n_L\left(c', c\right) + 1}{\sum_{c''} n_L\left(c', c''\right) + 28}. \tag{1}$$

where $n_L\left(c', c\right)$ is the number of times character c' was followed by c in a stream of text $\mathbf{L}$. This is called a *bigram* model of characters and will be useful in using context to help us reason about the true (i.e., hand written) text using only the typed version.

Of course, this model is a very crude approximation to what we know about text. For instance, it ignores our knowledge about characters forming regular structures we call words. In those immortal words of statisticians George E. P. Box and Norman R. Draper,

> Essentially, all models are wrong, but some are useful. However, the approximate nature of the model must always be borne in mind.[1]

Will *this* model be useful? As computational historians, you will have to make that assessment.

What about the *first* character in a sequence? To start the process we also need to identify $P(C = c \mid \mathbf{L} = \mathbf{l}, I)$, which can easily be calculated from $n_L(c'c,)$ by summing over all c' in the numerator and denominator.

1.2 Data

We now have twelve additional corrupted letters available for study, but none of them have original versions. Only half identify the typist. These may also be found in `/home/weinman/courses/CSC261/data/mills` where letters 2, 14, and 17 are attributed to typist one and letters 12, 13, and 19 are attributed to typist two, but we lack the originals. Moreover, we have corrupted letters 5, 6, 20, 21, 23, and 24 with neither attribution nor original.

2 Programming

In addition to the procedures from your prior analysis, you will need the following procedures (and their many helpers) provided in the Scheme file `/home/weinman/courses/CSC261/hmm/hmm.rkt`

`(count-transitions! filename counts)` takes the path to a "correct" file and a *`counts`* structure (as in the prior lab) and adjusts *`counts`* according to the number of times each character was preceded by another.

`(marginal-counts counts)` takes a *`counts`* structure containing $n_L(c'c,)$ and produces a single vector $n_L(c) = \sum_{c'} n_L(c'c,)$

`(normalize-marginal-counts! counts-marg)` takes a vector (i.e., a result from `marginal-counts`) and normalizes it so the entries sum to unity.

`(evidence-exact corrupt-filename typo-cond lang-cond lang-marg)` takes the name of a corrupted file, and the normalized count structures *`typo-cond`* (representing the sensor model) and *`lang-cond`* (representing the transition model) as well as the normalized count vector *`lang-marg`* (i.e., after applying `normalize-marginal-counts!` to the result of `marginal-counts`) for the first state. This procedure produces the marginal (over all unknown true characters) likelihood of the observed characters in the corrupted file

$$P(\mathbf{O} = \mathbf{o} \mid \mathbf{L} = \mathbf{l}, \mathbf{D} = \mathbf{d}, T = t, I) = \sum_{\mathbf{c}} P(\mathbf{O} = \mathbf{o}, \mathbf{C} = \mathbf{c} \mid \mathbf{L} = \mathbf{l}, \mathbf{D} = \mathbf{d}, T = t, I) \tag{2}$$

using the recursive update given by AIMA equation (15.7). ***Note:*** as `likelihood` before it, this procedure uses exact computations and is therefore prohibitively slow. However, it is easier to read and understand than the subsequent inexact version.

`(log-evidence corrupt-filename typo-cond lang-cond lang-marg)` calculates the log evidence (base e)

$$\ln P(\mathbf{O} = \mathbf{o} \mid \mathbf{L} = \mathbf{l}, \mathbf{D} = \mathbf{d}, T = t, I). \tag{3}$$

As AIMA points out (p. 573), inexact floating-point arithmetic typically results in numerical underflow; this implementation addresses that problem by carefully computing with log probabilities.

`(most-likely-sequence corrupt-filename typo-cond lang-cond lang-marg)` infers the most likely sequence of true characters

$$\widehat{\mathbf{c}} = \arg\max_{\mathbf{c}} P(\mathbf{C} = \mathbf{c} \mid \mathbf{O} = \mathbf{o}, \mathbf{L} = \mathbf{l}, \mathbf{D} = \mathbf{d}, T = t, I) \tag{4}$$

using the recursive Viterbi algorithm given by AIMA equation (15.11).

[1] Box, G. E. P., & Draper, N. R. (1987). *Empirical model building and response surfaces.* New York: John Wiley & Sons. (p. 424)

(count-errors-file *correct-filename other-filename*) counts the number of characters that differ between two text files of the same length.

(count-errors-list *correct-filename char-list*) counts the number of characters that differ between a text file and a character list of the same length.

(list->file *char-list out-filename*) writes a list of characters, `char-list`, to a file at *`out-filename`*.

3 Analysis

Your continuing quest as a computational historian trying to reconstruct the transcription record of Mr. Mills's typists now has you battling even more uncertainty. You will need to use probability to further analyze (less) data, convince your reader of the reliability of your conclusions, identify the typists of some unlabeled manuscripts *without* their accompanying originals, and possibly even restore some corrupted text.

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

The usual standards of composition apply to this report also (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 and labeled (i.e., with titles, row/column headers, axes and/or legends) as appropriate.

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

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

- Assess how your predictions and confidence from earlier change when the true text is missing. As before, "train" on a single letter from each typist and calculate Jaynes's evidence for the typist of each of the other two labeled letters *without* the original. (Again, there are 36 such configurations.) Your language model should be learned from the two "available" uncorrupted letters (i.e., one from each typist).

  How are your resulting predictions affected? What happens to the value of the Jaynes evidence when the original text is unavailable? Why? A two-dimensional "scatter" plot may be an effective visual aid (one axis for the earlier data, and the second axis for this scenario with unknown text); plot marker shape and/or color could indicate the true typist. See Figure 1 below as an example.

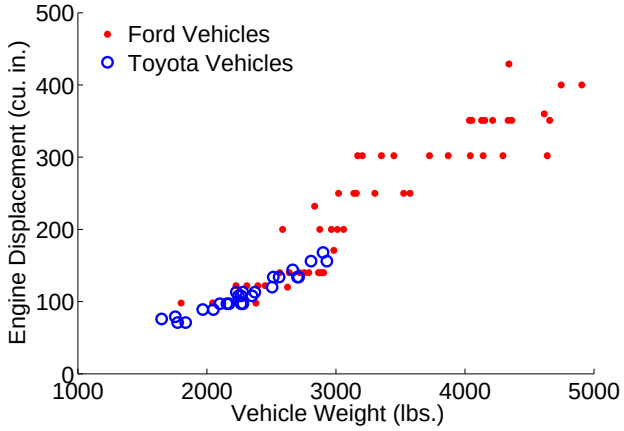

Figure 1: Vehicle weight versus engine size for two auto manufacturers. (Source: http://lib.stat.cmu.edu/datasets/cars.data)

- Assess the overall reliability of your predictions on letters without the original text. Train models for both typists on all six letters available with attribution and correct text (three each). The language model would also be learned from those six uncorrupted letters. Would your predictions for the typists of the six new attributed letters lacking accompanying text be correct? How often? How strong is the evidence (that is, how confident should you be)?
- Give attribution (along with your data and reasoning) for the typist of the six new unattributed letters lacking originals. Considering all your earlier results, how confident are you in each prediction?

What to turn in

- A file `analysis.rkt` with your supporting analytical code (which should display all the raw data results)
- A transcript `output.txt` of your analytical program's output
- A single PDF file `analysis.pdf` of your written report