CMPUT 652—Probability Models for Artificial Intelligence

Assignment 1

Fall 2003
Department of Computing Science
University of Alberta

Due: in class, *Thursday, October 23*
Worth: 25% of final grade (5 questions of equal weight, so each question is worth 5%)
Instructor: Dale Schuurmans, Ath409, x2-4806, dale@cs.ualberta.ca

---

Note Some of the questions require you to write Matlab programs which are to be submitted by email. When finished, please send a *single* tar file containing all of your .m files to dale@cs.ualberta.ca with a subject heading "A1 solutions".

Automated spelling correction

"Aoccdrnig to a rscheearchr at an Elingsh uinervtisy, it deosn't mttaer in waht oredr the ltteers in a wrod are, the olny iprmoetnt tihng is taht the frist and lsat ltteer is at the rghit pclae. The rset can be a toatl mses and you can sitll raed it wouthit porbelm. Tihs is bcuseae we do not raed ervey lteter by it slef but the wrod as a wlohe."

In this assignment you are going to design and implement automated spelling correctors. The idea will be to use a probability model for English words to infer the intended spelling of a word from a noisy misspelled observation. The first probability model we will consider is a *hidden trigram model*, which has the form shown in Figure 1.

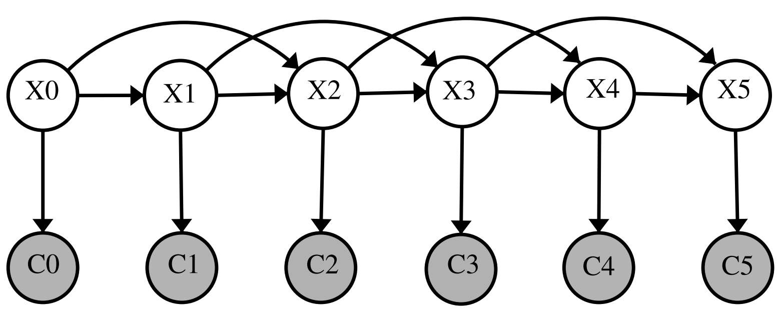

Figure 1: Hidden trigram model

In this model, we assume that the character sequence $c_0, c_1, ..., c_k, c_{k+1}$ is *observed*, where

c_0 is a whitespace character at the beginning of the observed word, and c_{k+1} is a whitespace character at the end of the observed word (and the word has length k). Given the observed character sequence $c_0, c_1, ..., c_k, c_{k+1}$, the idea is to infer the most likely hidden sequence $x_0, x_1, ..., x_{k+1}$ which will serve as the best guess at the correct spelling. Thus, we will assume that each variable takes on one of 27 values from the set

$$\{blank, a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z\}$$

Question 1 (Basic graphical representations—5%)

For this question, assume you have access to the conditional probability tables associated with the directed graphical model in Figure 1. Specifically, assume you are given $\text{P}(x_0)$, $\text{P}(x_1|x_0)$, and $\text{P}(x_i|x_{i-2}, x_{i-1})$ for $i = 2...k+1$, and $\text{P}(c_i|x_i)$ for $i = 0...k+1$,

(a) Draw the *factor graph* corresponding to Figure 1. Give each function node a unique label f_1, f_2, etc. Write the equation defining each f in terms of the conditional probability functions associated with the original directed model.

(b) Draw the *undirected graphical model* corresponding to Figure 1. Give each clique potential a unique label ψ_{U_1}, ψ_{U_2}, etc. Write the equation defining each ψ in terms of the functions given in Part (a).

(c) Draw the *clique graph (with separators)* corresponding to Figure 1. Give each clique potential a unique label ψ_{U_1}, ψ_{U_2}, etc., and each separator potential a unique label ϕ_{S_1}, ϕ_{S_2}, etc., Write the equation defining each ψ in terms of the functions given in Part (a).

(d) Write the formula which gives the probability of a complete configuration in terms of the ψ and ϕ functions from Part (c). How are the ψ and ϕ functions initialized?

(e) Is the clique graph obtained in Part (c) a tree? Is it a junction tree?

Question 2 (Estimating conditional probabilities—5%)

To build an actual spelling corrector, you are going to need to represent the tables $\text{P}(x_1|x_0)$, $\text{P}(c_i|x_i)$ and $\text{P}(x_i|x_{i-2}, x_{i-1})$ and fill them with appropriate numbers somehow. First, assume that the conditional probability tables will be the same at every point i in the sequence, and therefore you only need to represent one copy of $\text{P}(x_i|x_{i-2}, x_{i-1})$ and one copy of $\text{P}(c_i|x_i)$. Also, assume that the first and last observed characters are always blank and accurately delimit the word; that is $\text{P}(X_0 = \text{blank}) = \text{P}(X_{k+1} = \text{blank}) = 1$.

Write a Matlab function `[Tri, In] = estimatemodel1(filename)` that reads in a text file, converts all alphabetic characters to lowercase, converts all other characters to blank, and then estimates $\text{P}(x_i|x_{i-2}, x_{i-1})$ in the array `Tri(xi-2,xi-1,xi)` and estimates $\text{P}(x_1|x_0)$ in the array `In(x1)`. (Note that after conversion, all substrings separated by blanks are considered to be separate words.) The entry `Tri(xi-2,xi-1,xi)` should be assigned the observed frequency of `xi` given `xi-2` and `xi-1` at every location of `xi` within words in the file (excluding the first location in each word). The entry `In(x1)` should be assigned the observed frequency of `x1` as the first character in the words in the file. These two tables should be stored in two arrays of dimensions $27 \times 27 \times 27$ and 27 respectively (since we are assuming that x_0 always equals blank in the latter case).

Also write a Matlab function `[Obs] = observationmodel1(noise)` where the entry `Obs(xi,ci)` represents $\mathrm{P}(c_i|x_i)$. `Obs` is an array of size 27×27 and the parameter `noise` is assumed to be between 0 and 1. If `xi` is not blank, then assign values according to

$$\texttt{Obs(xi,ci)} = \begin{cases} 1 - \texttt{noise} & \text{if } \texttt{xi}{=}\texttt{ci} \\ \texttt{noise}/25 & \text{if } \texttt{xi}{\neq}\texttt{ci} \text{ and } \texttt{ci}{\neq}\text{blank} \\ 0 & \text{if } \texttt{ci}{=}\text{blank} \end{cases}$$

That is, assign a uniform probability that a non-blank character gets changed into any other non-blank character. If `xi` is blank, then assign values according to

$$\texttt{Obs(xi,ci)} = \begin{cases} 1 & \text{if } \texttt{ci}{=}\text{blank} \\ 0 & \text{if } \texttt{ci}{\neq}\text{blank} \end{cases}$$

That is, blanks are never changed nor mis-observed.

Now you have a probability distribution!

Question 3 (Calculating conditional probabilities—5%)

Write a Matlab function `[p] = condprob1(prediction,string,Tri,In,Obs)` that calculates the conditional probability of `prediction` $= x_0, x_1, ..., x_{k+1}$ given `string` $= c_0, c_1, ..., c_{k+1}$, under the probability distribution defined by `Tri`, `In` and `Obs`. You must implement the sum-product form of the junction tree algorithm to do this. Note that the evidence variables $c_0, c_1, ..., c_{k+1}$ must remain clamped at their observed values.

Question 4 (Maximum probability completion—5%)

Write a Matlab function `[prediction] = maxcompletion1(string,Tri,In,Obs)` that produces a `prediction` of the correct spelling for the input `string`, where of course `string` $= c_0, c_1, ..., c_{k+1}$ and `prediction` $= x_0, x_1, ..., x_{k+1}$. You must implement the max-sum form of the junction tree algorithm which calculates the maximum probability assignment to $x_0, x_1, ..., x_{k+1}$ given $c_0, c_1, ..., c_{k+1}$, under the probability distribution defined by `Tri`, `In` and `Obs`. Note that the evidence variables $c_0, c_1, ..., c_{k+1}$ must remain clamped at their observed values.

It is apparent that the spelling corrector based on the first model may not necessarily perform well on the original text *"Aoccdrnig to a rscheearchr at an Elingsh uinervtisy ..."*. This is because the first model does not emphasize the effects of the first and last character in improving prediction ability. Our second model will address this shortcoming.

Consider the directed graphical model shown in Figure 2 which allows one to encode the additional assumption that the first and last letter observed in the word are always correct. (Note that here we drop the c_0 and c_{k+1} variables because, as before, we assume x_0 and x_{k+1} are always blank and never mis-observed.)

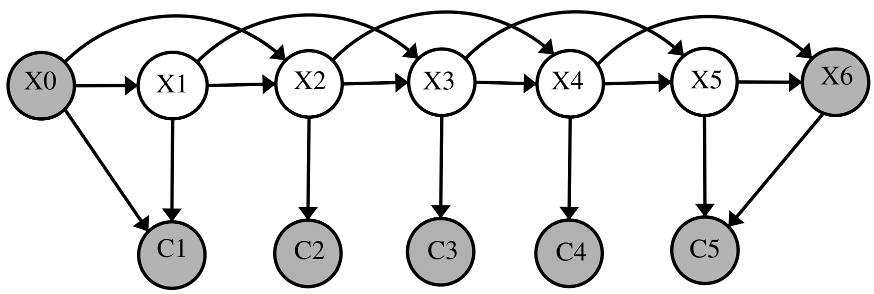

Figure 2: Hidden trigram model with end markers to improve the inference

Question 5 (Augmented prediction model—5%)

Repeat Questions 1-4 for the second model, illustrated in Figure 2. That is, first draw the different forms of graphical representations. Then write Matlab functions `estimatemodel2`, `observationmodel2`, `condprob2` and `maxcompletion2`. Note that you will have to extend the probability model to incorporate two additional components, $\mathrm{P}(c_1|x_0, x_1)$ and $\mathrm{P}(c_k|x_k, x_{k+1})$, as shown in Figure 2. We assume that the two characters observed at the start and the end of the word are more reliably observed, and therefore set

$$\mathrm{P}(c_1|x_0, x_1) = \begin{cases} 1 & \text{if } c_1 = x_1 \\ 0 & \text{if } c_1 \neq x_1 \end{cases}$$
$$\mathrm{P}(c_k|x_k, x_{k+1}) = \begin{cases} 1 & \text{if } c_k = x_k \\ 0 & \text{if } c_k \neq x_k \end{cases}$$

Test the accuracy of the two models and report your results. Can you suggest any further improvements to the probability models?