Midterm Review

Computational Linguistics: Jordan Boyd-Graber
University of Maryland
PCFG, LOGISTIC REGRESSION, TRANSDUCERS

Roadmap

- Answer your questions
- Go through examples of free response questions

Your Questions

Logistic Regression / Feature Engineering

Take V to be the set of possible words (e.g. “the”, “cat”, “dog”, . . .). Take V' to be the set of all words in V **plus** their reverses (e.g. “the”, “eht”, “cat”, “tac”, “dog”, “god”). You can assume that there are no palindromes in v (e.g. “eye”). You want a logistic regression that models $(x,y) : x \in V, y \in V'$ pairs as follows:

- With probability $\frac{1}{2}$ he chooses y to be identical to x
- With probability $\frac{1}{3}$ he chooses y to be the reverse of x
- With probability $\frac{1}{6}$ he chooses y to be some string that is neither x nor the reverse of x

Create a logistic regression (i.e. supply features f and weights θ) of the form:

$$p(y|x,\vec{\theta}) = \frac{\exp \sum_i \theta_i f_i(x,y)}{\sum_{y'} \exp \sum_i \theta_i f_i(x,y')} \tag{1}$$

that models Nathan’s process perfectly.

Logistic Regression

Features

1. 1 iff $x == y$ (id)
2. 1 iff $\text{rev}(x) == y$ (rev)
3. 1 always one (bias)

Logistic Regression

Features

1. 1 iff $x == y$ (id)
2. 1 iff $\text{rev}(x) == y$ (rev)
3. 1 always one (bias)

$$\exp\{\theta_{id} + \theta_{bias}\} = \frac{1}{2} \quad (2)$$

$$\exp\{\theta_{rev} + \theta_{bias}\} = \frac{1}{3} \quad (3)$$

$$(V' - 2)\exp\{\theta_{bias}\} = \frac{1}{2} \quad (4)$$

$$\quad (5)$$

Solving for parameters

$$\theta_{id} + \theta_{bias} = -\log 2 \tag{6}$$

$$\theta_{rev} + \theta_{bias} = -\log 3 \tag{7}$$

$$\theta_{bias} + \log(V' - 2) = -\log 6 \tag{8}$$

Solving for parameters

$$\theta_{id} + \theta_{bias} = -\log 2 \tag{6}$$

$$\theta_{rev} + \theta_{bias} = -\log 3 \tag{7}$$

$$\theta_{bias} + \log(V' - 2) = -\log 6 \tag{8}$$

$$\theta_{id} = \log 3 + \log(V' - 2) \tag{9}$$

$$\theta_{rev} = \log 2 + \log(V' - 2) \tag{10}$$

$$\theta_{bias} = -\log 6 - \log(V' - 2) \tag{11}$$

PCFG + LM

Suppose we have the the following language model over the alphabet $\{a, b\}$.

Bigram	Probability
$p(a \vert \texttt{<s>})$	0.45
$p(b \vert \texttt{<s>})$	0.45
$p(\texttt{</s>} \vert \texttt{<s>})$	0.1
$p(a \vert a)$	0.6
$p(b \vert a)$	0.2
$p(\texttt{</s>} \vert a)$	0.2
$p(a \vert b)$	0.2
$p(b \vert b)$	0.4
$p(\texttt{</s>} \vert b)$	0.4

1. Write a pcfg with non-terminals and weights such that it is equivalent to this language model. You should not need more than three non-terminals.
2. Compute the probability of the string <s> a a b </s> using the original language model and the corresponding pcfg derivation to show that they're equivalent.

PCFG + LM

FST

For any binary string x, let $w(x)$ denote the the number of 1's in x.

- For any binary string x and any integer i, $0 \leq i < w(x)$, let $f(x,i)$ denote the number of 0's between the i^{th} 1 and the $(i+1)^{\text{st}}$ 1 in the binary string $1x$, where we index the $w(x)+1$ 1's in $1x$ from left to right starting at zero. Example: If $x = 11000100$, then $w(x) = 3, f(x,0) = 0, f(x,1) = 0, f(x,2) = 3$, and $f(x,i)$ is undefined for $i \geq 3$.
- For any binary string x, let $g(x)$ denote the binary string of length $w(x)$ with i^{th} bit (indexing the bits from left to right starting at zero) equal to the parity of $f(x,i)$ (that is, 0 if even, 1 if odd). Example: If $x = 11000100$, then $g(x) = 001$.

Design a finite state transducer that maps any given input binary string x to the output binary string $g(x)$.

FST