CS 533: Natural Language Processing **(Due: 2/4/20)**

Assignment 1

Name: Revanth Korrapolu*, NetID:* rrk69 *Students discussed with:* Joe Redling-Pace, Yoni Friedman

Problem 1: Preliminaries

((1+1+1+1) + (1+1+1+1) + (1+1+1) = 11 points)

1. **(Probability)**

(a)

$$Var(x) = E[(X-\mu)^2] = E[(X^2 - 2\mu_X + \mu_X^2)] = E[X^2] - E[2\mu_X X] + E[\mu_X^2]$$
$$= E[X^2] - 2\mu_X E[X] + \mu_X^2 = E[X^2] - 2\mu_X^2 + \mu_X^2 = E[X^2] - \mu_X^2$$

(b) i. Mean (μ)

$$\sum_{x\epsilon\chi} p_x(x)x = \sum_{x=1}^{6} p_x(x)x = \frac{1}{6}*1 + \frac{1}{6}*2 + \frac{1}{6}*3 + \frac{1}{6}*4 + \frac{1}{6}*5 + \frac{1}{6}*6 = \mathbf{3.5}$$

ii. Variance

$$E[X^2] - \mu_X^2 = \frac{1}{6}*1^2 + \frac{1}{6}*2^2 + \frac{1}{6}*3^2 + \frac{1}{6}*4^2 + \frac{1}{6}*5^2 + \frac{1}{6}*6^2 - (3.5)^2 \approx \mathbf{2.92}$$

iii. Entropy

$$\sum_{x=1}^{6} p_x(x) * -log_2(p_x(x)) = \sum_{x=1}^{6} \frac{1}{6} * -log_2(\frac{1}{6}) = -1 * log_2(\frac{1}{6}) = log_2(6) \approx \mathbf{2.58}$$

(c) i. Mean (μ)

$$\sum_{x\epsilon\chi} p_x(x)x = \sum_{x=1}^{6} p_x(x)x = 1*6 = \mathbf{6}$$

ii. Variance

$$E[X^2] - \mu_X^2 = 1*6^2 - (1)^2 = \mathbf{0}$$

iii. Entropy

$$\sum_{x=1}^{6} p_x(x) * -log_2(p_x(x)) = \sum_{x\epsilon\{6\}} 1 * -log_2(1) = log_2(1) = \mathbf{0}$$

(d) i.

$$-\sum_{x=1}^{6} p_x(x) * log_2(q_x(x)) = (\frac{1}{6} * log_2(0)) * 4 + (\frac{1}{6} * log_2(\frac{1}{2})) * 2 = -(-\infty + 0) = \infty$$

ii.

$$-\sum_{x=1}^{6} p_x(x) * log_2(q_x(x)) = -((\frac{1}{6} * log_2(\frac{1}{5})) * 4 + (\frac{1}{6} * log_2(\frac{1}{10})) * 2) \approx \mathbf{2.655}$$

iii.

$$-\sum_{x=1}^{6} p_x(x) * log_2(q_x(x)) = -((\frac{1}{6} * log_2(\frac{1}{6})) * 6) \approx \mathbf{2.585}$$

2. **(Linear Algebra)**

(a)

$$M = \begin{bmatrix} -3 \end{bmatrix}$$

(b)

$$M = \begin{bmatrix} 13 & 5 & 3 & -2 \\ 28 & 14 & 9 & -8 \end{bmatrix}$$

(c)

$$M = \begin{bmatrix} -176 \\ 988 \\ 51 \\ 82 \\ 135 \end{bmatrix}$$

(d) Invalid

3. **(Optimization)**

(a) The minimum for equation (1) is **3**. The minimum for equation (2) is $-\infty$. The first derivative indicates the slope at each x-value. The second derivative indicates the curvature at each x-value.

(b) $f'(x) = x - 3$
$f''(x) = 1$
One can find the local min/max of a function by finding where the first derivative of the function equals zero. $f'(x) = 0$ when x = 3, which is the minimum point (3,2). Since $f''(x)$ is positive at $x = 3$, we know that this must be a minimum.

(c) $f'(x) = (x - 3)^2$
$f''(x) = 2(x - 3)$
From looking at $f''(x)$, we can tell that there is an inflection point at $x = 3$. The local minimum on the right side of the inflection point is at $x = 3$. However, since the curvature is always negative on the left side of the inflection point ($x < 3$), we know that the global minimum will be at $(-\infty, -\infty)$

Problem 2: n-Gram Models (4 + (2+1+1+1) = 9 points)

1. **(Relative Frequency Lemma)**

(a) Start by simplifying the derivative

$$0 = \frac{\partial}{\partial q_j} \sum_{i \epsilon [n]} c_i * log(q_i) - \lambda(1 - \sum_{i \epsilon [n]} q_i) = \frac{c_i}{q_i * ln2} - \lambda$$

We can solve for λ

$$\lambda = \frac{-c_i}{q_i * ln2} = \frac{-c_i}{\frac{c_i}{N} * ln2} = -\frac{N}{ln2} \quad \forall i \epsilon [n]$$

Also

$$1 = \sum_{j \epsilon [n]} q_j = \sum_{j \epsilon [n]} \frac{-c_j}{\lambda * ln2} = -\frac{1}{\lambda * ln2} * \sum_{j \epsilon [n]} c_j = -\frac{N}{\lambda * ln2}$$

If we substitute λ

$$-\frac{N}{-\frac{N}{ln2} * ln2} = 1$$

Since

$$N = \frac{c_i}{q_i} = 1 \quad \forall i \epsilon [n]$$

$$q_i^* = \frac{c_i}{N}$$

Therefore, we have solved the system and proven lemma 1.

2. **(Maximum Likelihood Estimation (MLE) of the Trigram Language Model)**

(a) Starting with Equation 7:

$$t(x_1 \dots x_m) = \prod_{j=1}^{M+1} t(x_j | x_{j-2}, x_{j-1})$$

By taking the log on both sides we get:

$$log(t(x_1 \dots x_m)) = \sum_{j=1}^{M+1} log(t(x_j | x_{j-2}, x_{j-1}))$$

To maximize $log(t(x_1 \dots x_m))$, we have the following optimization problem:

$$t_x^{MLE} = \mathbf{E}\left[t(x_j | x_{j-2}, x_{j-1})\right] = \arg\max_{t_x \epsilon \theta} \mathbf{E}\left[t(x_j | x_{j-2}, x_{j-1})\right] = \arg\min_{t_x \epsilon \theta} \sum_{(x,x',x'')\epsilon\chi} \#t(x, x', x”) {*} log_{t_x}(x” | x, x')$$

Let $\#(x, x', x”) = c_i$ as in Lemma 1

$$\hat{t}^{MLE}(x” | x, x') = \frac{c_i}{N} = \frac{c_i}{\sum\limits_{i \epsilon N} c_i} = \frac{\#(x, x', x”)}{\sum \#(x, x', x”)} = \frac{\#(x, x', x”)}{\#(x, x')}$$

(b) Format: (x, x’ x”)

(START, START, the) = $\frac{3}{3} = 1$

(START, the, dog) = (START, the, cat) = (START, the, mouse) = $\frac{1}{3}$

(the, cat, STOP) = (the, cat, ate) = (the, mouse, STOP) = (the, mouse, screamed) = $\frac{1}{2}$

(the, dog, ignored) = (dog, ignored, the) = (ignored, the, cat) = (cat, ate, the) = (ate, the, mouse) = (mouse, screamed, STOP) = (cat, STOP, STOP) = (mouse, STOP, STOP) = (screamed, STOP, STOP) = $\frac{1}{1} = 1$

(c) Perplexity = $e^{Entropy} = e^{-(ln(1/3)+2*ln(1/2))} = 12$ nats

(d) Perplexity = $e^{CrossEntropy} = e^{-(2/3*ln(1/3)+2/3*ln(1/3)+2*ln(1/2))} = 17.3$ nats

Problem 3: Programming

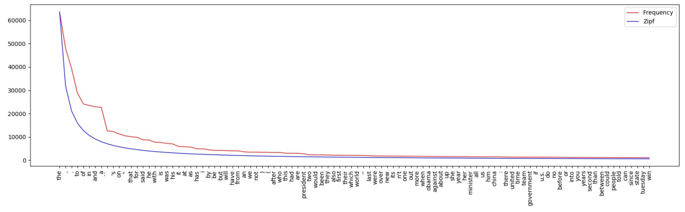
(2 + 1 + 1 + 1 + 2 + 1 + 3 + 1 = 12 points)

(Code must be submitted as well, with unambiguous commands for replicating reported results.)

1. **see code**

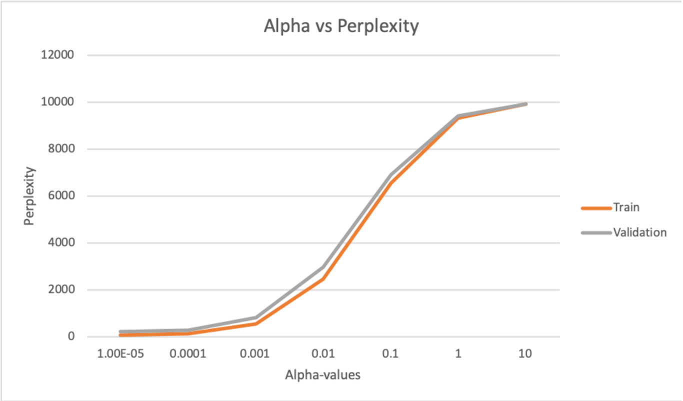
2. Basic - 69148
   NLTK - 41844
   WP - 15716
   BPE - 22581

3. As you can see from the plot, the data approximately maps the Zipf regression line.

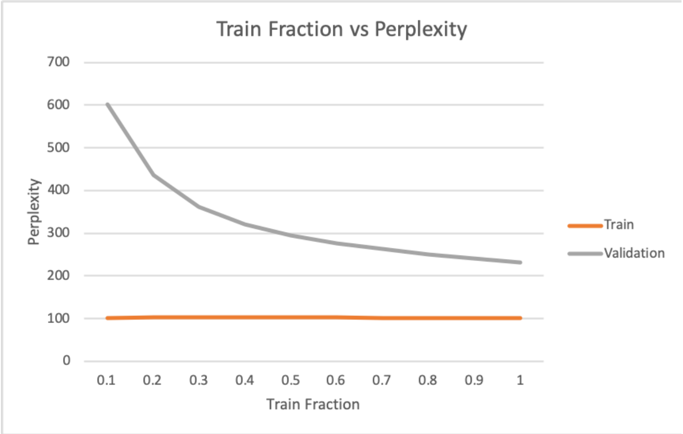


4. Training Perplexity: 70.967555
   Validation Perplexity: ∞
   The Validation perplexity is calculated by taking the inverse probability of the test set and normalizing by the number of words. Since some vocabulary words do not appear, the number of occurrences is 0. As such, our model is assigning a zero probability to unknown word. When we are dividing by zero, this results in an infinite perplexity.

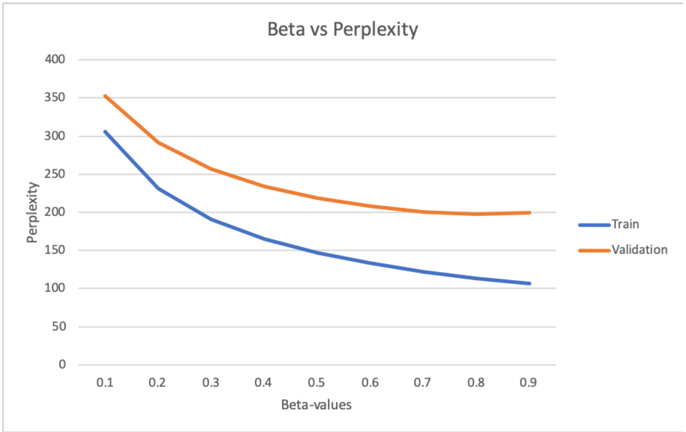
5. Laplace smoothing re-normalizes the perplexity and predict each vocabulary word to appear with a positive, non-zero value. When we plot the perplexity on both train and validation sets as a function of α, their appears to be a logistic regression.



6. As expected, the perplexity decreases as more training data is used in the model. This is intuitive because one would expect the probability model to better predicts a sample if the the training data is more encompassing.



7. As the beta values increased for the interpolation smoothing, the training and validation perplexity decreased in a regression that models exponential decay.



8. Minimum Validation Perplexity: 184.7
   Parameters:

$\textit{train fraction} = 0.9$
$\textit{smoothing} = \text{interpolation}$
$\alpha = 0.001$
$\beta = 0.8$