CS 510 Assignment 3:

Daniel Campos

April 11th,2021

1 Problem 1

1.1 Write down the likelihood for log p(D— θ)

$logp(D|\theta) = \sum_{w \in D} log((1-\eta)p(w|\theta) + (\eta)p(w|C))$ where C is the background language model. Since we do not care about the background language model we can set $\eta = 0$ since no words will be generated by the background language model. Thus, the like hood becomes $logp(D|\theta) = \sum_{w \in D} log(p(w|\theta))$

1.2 Derive the E-step and M-step for estimating the unknown parameter λ in iteration t, for t = 1, 2

Since the background model never occurs we can just set $p(z_{d,w} = B) = 0$ and only focus on updating $p(z_{d,w} = \theta) = \frac{\pi^{(}_{d,\theta} n)(w|\theta)}{\pi_{d,\theta'} p^{(}n)(w|\theta')}$ for the E-Step and for M-Step $\pi^{(}_{d,j} n{+}1) = \frac{\sum_{w \in V} c(w,d)(1-p(z_{d,w}=\theta')p(z_{w,d}=\theta)}{\sum_{w \in V} c(w,d)(1-p(z_{d,w}=\theta')p(z_{w,d}=\theta')}$

2 Derive the E-step and M-step for estimating p(w—H).

First off, we can treat this as having two language models, for writer h given by θ_h and writer t given by θ_t. We know $\theta_h = 0.9$, $\theta_t = 0.1$ and we know the $p(w|t)$ but not $p(w|h)$. We initialize $p(w|\theta_h)$ to a random value and use EM. For our optimization we set our E-step to :
$p(z = 1|\theta_h) = \frac{\theta_h p^{(n)} p(w|\theta_h)}{\theta_h p^{(n)} p(w|\theta_h) + p(\theta_t) p^{(n)} p(w|\theta_t)}$ since we already know θ_h we and the true distribution for writer t we can replace their probability with a constant which allows us to simplify to e-step to$p(z = 1|\theta_h) = \frac{0.9 p^{(n)} p(w|\theta_h)}{0.9 p^{(n)} p(w|\theta_h) + c_w}$ where c_w is the constant for $\theta_t * p(w|\theta_t)$. For the M-Step modify the formula to update the word distribution. $p^{(}n+1)(w|\theta_h) = \frac{c(w,d)(1-p(z=1|\theta_h))p(w|D)}{\sum_{w' \in V} \sum_{d \in D} c(w',d)(1-p(z=h))p(z=t)}$

3 Question 3

To estimate the remaining 900 documents we apply the EM for PLSA as covered in lecture. First, we initialize all unknown parameters(true topic mapping of word distributions) randomly. Next we repeat the E-Step and the M step (covered shortly) until convergence.
E-step: Our hidden variable in this question is our topic indicator by word denoted by $z_{d,w}$ where $z_{d,w} \in B, 1, 2$ where B is background, 1 is Seattle and 2 is Chicago. We run $(z_{d,w} = 1) = \frac{\pi^{(}_{d,j} n) p^{(}n)(w|\theta_1)}{\sum_{j'=2} \pi^{(}_{d,j} n) p^{(}n)(w|\theta_2)}$, $(z_{d,w} = 2) = \frac{\pi^{(}_{d,j} n) p^{(}n)(w|\theta_2)}{\sum_{j'=1} \pi^{(}_{d,j} n) p^{(}n)(w|\theta_1)}$ and $p(z_{d,w} = B) = \frac{\lambda_b p(w|\theta_b)}{\lambda_b p(w|\theta_b) + (1-\lambda_b) \sum_{j=1}^{2} \pi^{(}_{d,j} n) p^{(}n) w|\theta_j)}$.

The M-step re-estimates the probability of doc d covering a topic θ_j where: $\pi^{(}_{d,j}n{+}1) = \frac{\sum_{w\in V} c(w,d)(1-p(z_{d,w}=B))p(z_d}{\sum_{j'}\sum_{w\in V} c(w,d)(1-p(z_{d,w}=B))p(}$
and re-estimates the probability of word w for topic θ_j by: $p^{(}n{+}1)(w|\theta_j) = \frac{\sum_{w\in V} c(w,d)(1-p(z_{d,w}=B))p(z_{d,w}=j)}{\sum_{w'\in V}\sum_{d\in D} c(w',d)(1-p(z_{d,w'}=B))p(z_{d,w'}=j)}$

4 Topic Estimation

4.1 Question 1.1

Implemented

4.2 Question 1.2

The first sequence, sampleseq1 and samplemod1 tags every a with 0 and b with 1 since output probabilities for 0 have been set to be A 0.9999 and for 1 have set to be b for 0.99999 and the transmission probabilities bet weens states are equal. This mean that it is most unlikely that any B receive a 0, a receive a 1 and transferring between 0 to 1 is as common as staying.
For the second sequnce, sampleseq2 and samplemod2 we produce eight zeros followed by 8 ones because of the different mix in transmission probabilities and output probabilities. Our first 8 outputs are 0 because it is dominated by As and as As are unlikley to be produced by 1 the sequence is likely zeros. Once the sequence transitions mostly to b, it is more probable that we had the minor odds of transitioning from 0 to 1 than producing that many B with state 0 so the second eight states are 1.

4.3 Question 1.3

See zip.

4.4 Question 2.1

See zip.

4.5 Question 2.2

See zip.

4.6 Question 3.1

The probabilities are slighly off as the model has 0.85583, 0.156617 which should be 0.8, 0.2 and 0.144053, 0.843271 which should be 0.2, 0.8.
The learned tagging is the same as the tagging produced by samplemod2.

4.7 Question 3.2

Yes this is able to identify DNA and amino acids.
If I insert one P it is able to identify as amino because there are no DNA sequences with P.
If I insert six P after the second A they are also all correctly identified as Aminos. If we look at the trained model we see for P 0.0736726 5.70029e-07 meaning P is never a DNA sequence and always

an amino. Even when it is unlikely to go from DNA to amino (0 to 1 0.960808 0.0701715 and 1 to 0 0.0391916 0.929828) since P is never DNA it doesn't change.