NLP Homework1

Xiangan Liu xl2699@columbia.edu

September 27, 2017

Problem 1.

Consider the computation of perplexity P.

$$\begin{aligned} l &= \frac{1}{M}\sum_{i=1}^{m} logp(x^{(i)}) \\ &= \frac{1}{M}\sum_{i=1}^{m}\sum_{j=1}^{n} logq(w_j^i|w_{j-1}^i, w_{j-2}^i) \end{aligned}$$

$\sum_{i=1}^{m}\sum_{j=1}^{n} logq(w_j^i|w_{j-1}^i, w_{j-2}^i)$ is the sum of the log probability of all the trigrams that exist in all the sentences.If we have the count of each trigram,we can simplify the equation.Since we have the count $c^{'}(u,v,w)$,which is the number of times that (u,v,w) is seen in valid-set.

$$\begin{aligned} l &= \frac{1}{M}\sum_{i=1}^{m}\sum_{j=1}^{n} logq(w_j^i|w_{j-1}^i, w_{j-2}^i) \\ &= \frac{1}{M}\sum_{w,u,v} c'(u,v,w) logq(v|u,w) \end{aligned}$$

We observe that $L(\lambda_1, \lambda_2, \lambda_3) = M * l$ and $P = 2^{-l}$,if we maximize $L(\lambda_1, \lambda_2, \lambda_3)$,we will also get max($l$) since M is a constant.Thus,we will have min(P) when we maximize $L(\lambda_1, \lambda_2, \lambda_3)$.

Problem 2.

The problem will be that $\sum_{w,u,v\in V} q(v|w,u)$ may not equal to 1,which violates the basic rule of probability distribution.In the original interpolation,we have:

$$\begin{aligned} \sum_{v\in V} q(v|w,u) &= \sum_{v\in V}(\lambda_1 * q_{ML}(v|w,u) + \lambda_2 * q_{ML}(v|u) + \lambda_3 * q_{ML}(v)) \\ &= \lambda_1 * \sum_{v\in V} q_{ML}(v|w,u) + \lambda_2 * \sum_{v\in V} q_{ML}(v|u) + \lambda_3 * \sum_{v\in V} q_{ML}(v) = \lambda_1 + \lambda_2 + \lambda_3 = 1 \end{aligned}$$

In the previous bigram method,all the λ changed according to the times we see the bigram.In $\sum_{v \in V} q(v|w,u)$,because the (w,u) is fixed and we can get the fixed λ.Therefore,it's same as the original interpolation and we can ensure the sum is one.However, in this trigram based interpolation method,we will have 12 parameters and we cannot ensure the sum of q(v—w,u) is still one.

Problem 3.

Input: a sentence $x_1......x_n$
Definitions:Define K to be the set of possible tags.Define $K[0] = K[1] = *$,and $K_k = \{y|y \in T(x_k)\}$,for k=1...n and $x = x_k$.

- For k=1....n,
  - For $u \in K_{k-1}, v \in K_k$,
    - $\pi(k,u,v) = \max_{w \in K_{k-2}}(\pi(k-1), w, u) * q(v|w,u) * e(x|v))$
    - $bp(k,u,v) = arg \max_{w \in K_{k-2}}(\pi(k-1,u,v) * q(v|w,u) * e(x_k|y))$
- Set $(y_{n-1}, y_n) = argmax_{u \in K_{n-1}, v \in K_n}((\pi(n,u,v)) * q(STOP|u,v))$
- For k=(n-2).....1,

$$y_k = bp(k+2, y_{k+1}, y_{k+2})$$

- Return the tag sequence $y_1....y_n$

The algorithm runs in $O(nK^3)$ since it is linear in the length of the sequence, and cubic in the number of tags.