TTIC 31210: Advanced Natural Language Processing

Homework 4:

Yiyang Ou

June 5, 2019

1 Gibbs Sampling for HMMs (40 points)

1.1

$$LHS = \frac{P(Y_t = y, Y_{-t} = y_{-t}, X = x)}{P(Y_{-t} = y_{-t}, X = x)} \tag{1}$$

$$= \frac{p_\tau(< /s > |y_T)\Pi_{k=1}^{T} p_\tau(y_k|y_{k-1})p_\eta(x_k|y_k)}{P(Y_{-k} = y_{-k}, X = x)} \tag{2}$$

$$= p_\tau(y|y_{t-1})p_\tau(y_{t+1}|y)p_\eta(x_t|y) \cdot C \tag{3}$$

where C is some constant that doesn't contain the variable Y_t. Thus $LHS \propto p_\tau(y_t|y_{t-1})p_\tau(y_{t+1}|y_t)p_\eta(x_t|y_t)$.

1.2

$$p(Y_1 = y|Y_{-1} = y_{-1}, X = x) \propto P(X_1 = x_1|Y_1 = y)P(Y_1 = y| < s >)P(Y_2 = y_2|Y_1 = y)$$

$$p(Y_T = y|Y_{-T} = y_{-T}, X = x) \propto P(X_T = x_T|Y_T = y)P(Y_T = y|Y_{T-1} = y_{T-1})P(< /s > |Y_T = y)$$

1.3

See jupyter notebook implementation.

1.4

See jupyter notebook for the reports.

1.5

I found for small $\beta = 0.5$, the accuracy is worse and it converges very slowly. This is likely due to β flattens the distribution thus making it hard to choose high possibility word. Yet these high possibiility and correct words can be picked up by large $\beta = 2$, and we see it converges quickly. But note that $\beta = 5$ gives similar accuracy as $\beta = 0.5$. This is likely due to the bottleneck is the model's high probability word isn't the correct word (so model misses some true patterns), thus sharpening more won't help.

1.6

I tried with other schedule: increasing 0.2 and 0.05 at each iteration but the accuracy is similar. And $k = 0.05, K = 1000$ is slightly better than $k = 0.1, K = 1000$ (0.8899403578528827 vs 0.889860834990059)

2 2. Gibbs Sampling for Minimum Bayes Risk Inference (20 points)

2.1

When using 0-1 cost, since $cost(y, y') = 0$ for $y' \neq y$. Equation 3 reduces to:

$$\hat{(}y) = argmin_y \sum_{y' \neq y} P(Y = y'|X = x) = argmin_y(1 - P(Y = y|X = x)) = 1 - argmax_y P(Y = y|X = x)$$

so it gives the same reuslt as equation 2.

2.2

$$P(Y_t = y|X = x) = \sum_{y_{-t}} P(Y_t = y, Y_{-t} = y_{-t}|X) \approx \sum_{y_{-t}} \frac{1}{K} \sum_{i=1}^{K} \mathbb{I}[\tilde{y}^{(i)} = y_{-t} \cup y] = \frac{1}{K} \sum_{i=1}^{K} \mathbb{I}[\tilde{y}_t^{(i)} = y]$$

2.3

See experiments and reports in jupyter notebook. I found that unlike the previous case, small $\beta = 0.5$ already gives as good accuracy as larger β if we allow it to fully converge (for larger K). This difference from the case where we used only the final sample can be explained by that: when we do MBR, we smooth out probability by averaging, so it would return the word with higher probability, in other words it gives a good approximation to argmax even when the distribution is relatively flat; yet if we just pick the last one,since the distribution isn't sharp, there's high chance due to randomness that we deviate from argmax and pick an incorrect word .