# CSE 325/425 (Spring 2021) Homework 3

Due on 11:55pm, Mar 17, 2021

**Grading:** All questions have the same points (25 each). We will randomly grade some of the questions.

**Submitting:** Only electronic submissions on Coursesite are accepted. You can handwrite your answers on papers and then scan them to images. If you need to plot figures using a computer, the plotted files should be saved and included in the submitted pdf file. Submit a single pdf file named

<Your LIN>HW3.pdf

Other format will not be accepted.

**Questions:**

1. List the probabilities that HMM, MEMM and CRF need to estimated during model training, respectively. Don't just write down symbols or equations from the slides, but explain how to sum these probabilities to the constant 1 (in other words, point out the sample space of each probability distribution).

   [[[ HMM: find transition probabilities ($\Pr(q_t|q_{t-1})$ and $\sum_{q_t}\Pr(q_t|q_{t-1})=1$ for a fixed $q_{t-1}$), emission probabilities ($\Pr(o_t|q_t)$ and $\sum_{o_t}\Pr(o_t|q_t)=1$ for a fixed word $o_t$), and starting probabilities ($\Pr(q_1)$ and $\sum_{q_1}\Pr(q_1)=1$). <==

   MEMM: find the probability of one POS-tag given the privious tag and the current word.

$$\Pr(q_t=c|q_{t-1},o_t;\boldsymbol{\theta}^c)=\frac{1}{Z(q_{t-1},o_t)}\exp\left\{\sum_{i=1}^{d}\theta_i^c f_i(q_{t-1},o_t,q_t=c)\right\} \tag{1}$$

   Both current and previous tags can take any values from the set of all POS-tags.

$$\sum_{c}\Pr(q_t=c|q_{t-1},o_t;\boldsymbol{\theta}^c)=1. \tag{2}$$

   CRF: find the probability of a POS tag sequence conditioned on a given sentence.

$$\Pr(q_1,\ldots,q_T|o_1,\ldots,o_T;\boldsymbol{\theta})=\frac{1}{Z(O)}\exp\left\{\sum_{t=1}^{T}\sum_{i=1}^{d}\theta_i f_i(q_{t-1},q_t,o_t)\right\} \tag{3}$$

$$\sum_{q_1,\ldots,q_T}\Pr(q_1,\ldots,q_T|o_1,\ldots,o_T;\boldsymbol{\theta})=1. \tag{4}$$

   ]]]

2. Design one feature function of the multi-class logistic regression POS-tagger, so that person's names can be tagged as NNP accurately.

   (*Hints: refer to the slides of lecture 9 and consider what characteristics of persons' names in a sentence will distinguish them from other words.*)

   [[[ One simple example is to test if the first letter is capitalized and the word does not appear at the beginning of the sentence (since the first word of a sentence is usually capitalized). ]]] <==

| Secretariat | is | expected to | race tomorrow. | | |
|---|---|---|---|---|---|
| NNP | VBZ | VBN | TO | VB | RB |

| | | $f_1$ | $f_2$ | $f_3$ | $f_4$ | $f_5$ | $f_6$ |
|---|---|---|---|---|---|---|---|
| $c = \text{NN}$ | $\mathbf{f}(o_t, \text{NN})$ | 1 | 0 | 0 | 0 | 0 | 1 |
| | $\boldsymbol{\theta}_{\text{NN}}$ | 0.8 | 1 | -2 | 3 | 0.1 | -1.3 |
| $c = \text{VB}$ | $\mathbf{f}(o_t, \text{VB})$ | 0 | 1 | 0 | 1 | 1 | 0 |
| | $\boldsymbol{\theta}_{\text{VB}}$ | 0.9 | 0.8 | -1 | 0.01 | 0.1 | 0 |
| $c = \text{VBG}$ | $\mathbf{f}(o_t, \text{VBG})$ | 0 | 0 | 0 | 0 | 0 | 0 |
| | $\boldsymbol{\theta}_{\text{VBG}}$ | 1 | 0.3 | 9 | 0.3 | -0.4 | -3.4 |

Figure 1: Features and parameters

3. Based on the feature vectors and parameter vectors and the training sentence in Figure 1, take one gradient descent step to update the parameter vector $\boldsymbol{\theta}_{\text{NN}}$ at the word $o_t = \textit{Race}$.

   [[[ According to the multi-class logistic regression, the negative log-likelihood loss for this word and tag is <==

$$-\log \Pr(q_t = VB | o_t; \boldsymbol{\theta}_{\text{NN}}, \boldsymbol{\theta}_{\text{VB}}, \boldsymbol{\theta}_{\text{VBG}}) = -\log \frac{\exp(\boldsymbol{\theta}_{\text{VB}}^\top \mathbf{f}(o_t, \text{VB}))}{\sum_{q \in \{\text{NN,VB,VBG}\}} \exp(\boldsymbol{\theta}_q^\top \mathbf{f}(o_t, q))}. \tag{5}$$

   The gradient of the loss with respect to the inner product $\boldsymbol{\theta}_{\text{NN}}^\top \mathbf{f}(o_t, \text{NN})$ is

$$\frac{\exp(\boldsymbol{\theta}_{\text{NN}}^\top \mathbf{f}(o_t, \text{NN}))}{\sum_{q \in \{\text{NN,VB,VBG}\}} \exp(\boldsymbol{\theta}_q^\top \mathbf{f}(o_t, q))} = \Pr(q_t = NN | o_t; \boldsymbol{\theta}_{\text{NN}}, \boldsymbol{\theta}_{\text{VB}}, \boldsymbol{\theta}_{\text{VBG}}), \tag{6}$$

   (using $\log \frac{a}{b} = \log a - \log b$, $\log \exp(a) = a$, and the gradient of $\boldsymbol{\theta}_{\text{VB}}^\top \mathbf{f}(o_t, \text{VB})$ with respect to $\boldsymbol{\theta}_{\text{NN}}^\top$ is 0.)
   Using the chain rule, the partial derivative of the loss with respect to $\boldsymbol{\theta}_{\text{NN}}$ is

$$\Pr(q_t = NN | o_t; \boldsymbol{\theta}_{\text{NN}}, \boldsymbol{\theta}_{\text{VB}}, \boldsymbol{\theta}_{\text{VBG}}) \mathbf{f}(o_t, \text{NN}). \tag{7}$$

   The gradient update is

$$\boldsymbol{\theta}_{\text{NN}}^{(t+1)} \leftarrow \boldsymbol{\theta}_{\text{NN}}^{(t)} - \eta \Pr(q_t = NN | o_t; \boldsymbol{\theta}_{\text{NN}}, \boldsymbol{\theta}_{\text{VB}}, \boldsymbol{\theta}_{\text{VBG}}) \mathbf{f}(o_t, \text{NN}), \tag{8}$$

   where $\eta$ is a learning rate and $\mathbf{f}(o_t, \text{NN})$ is from Figure 1. The gradient update has an intuitive explanation: if the current model thinks that $\Pr(q_t = NN | o_t; \boldsymbol{\theta}_{\text{NN}}, \boldsymbol{\theta}_{\text{VB}}, \boldsymbol{\theta}_{\text{VBG}})$ is high and makes the wrong prediction, then $\boldsymbol{\theta}_{\text{NN}}^{(t)}$ is pushed away from the feature vector $\mathbf{f}(o_t, \text{NN})$ for the wrongly predicted tag NN. ]]]

4. For an RNN with

$$\begin{aligned}
\mathbf{a}^{(t)} &= \mathbf{b} + W\mathbf{h}^{(t-1)} + U\mathbf{x}^{(t)} && (9)\\
\mathbf{h}^{(t)} &= \tanh(\mathbf{a}^{(t)}) && (10)\\
\mathbf{o}^{(t)} &= \mathbf{c} + V\mathbf{h}^{(t)} && (11)\\
\hat{\mathbf{y}}^{(t)} &= \text{softmax}(\mathbf{o}^{(t)}) && (12)
\end{aligned}$$

   Assume that $\mathbf{o}^{(t)}$ is of length 2, $\mathbf{h}^{(t)}$ is of length 3, and $\mathbf{x}^{(t)}$ is of length 2. Write down the dimensionalities of the parameters $\mathbf{b}$, $W$, $U$, $\mathbf{c}$, and $V$.

   [[[ $\mathbf{b} \in \mathbf{R}^3$, $W \in \mathbf{R}^{3\times 3}$, $U \in \mathbf{R}^{3\times 2}$, $\mathbf{c} \in \mathbf{R}^2$, $V \in \mathbf{R}^{2\times 3}$. ]]] <==

5. Based on your answer to the above question, populate the parameters with all 1's (that is, all parameters are matrices/vectors of all 1's). Then execute the above equations for the two steps on two input vectors $\mathbf{x}^{(1)} = [1, 1]^\top$ and $\mathbf{x}^{(2)} = [2, 2]^\top$.

   [[[ You can modify the RNN example codes in the IPython notebook "Recurrent Neural Networks.ipynb" <==

to do the calculations.

$$\mathbf{a}^{(1)} = [6., 6., 6.] \tag{13}$$
$$\mathbf{h}^{(1)} = [0.99998771, 0.99998771, 0.99998771] \tag{14}$$
$$\mathbf{o}^{(1)} = [4., 4.] \tag{15}$$
$$\hat{\mathbf{y}}^{(1)} = [0.5, 0.5] \tag{16}$$

$$\mathbf{a}^{(2)} = [7.99996313, 7.99996313, 7.99996313] \tag{17}$$
$$\mathbf{h}^{(2)} = [0.99999977, 0.99999977, 0.99999977] \tag{18}$$
$$\mathbf{o}^{(2)} = [3.99996313, 3.99996313] \tag{19}$$
$$\hat{\mathbf{y}}^{(2)} = [0.5, 0.5] \tag{20}$$

]]]