APML - Ex 2

Rhea Chowers, 204150643

December 8, 2019

Theoretical Questions

0.1 MEMM contains HMM - Cancelled

0.2 Higher Order Markov Model - Cancelled

0.3 Energy Based Model Gradient

1. We saw that for the MEMM

$$\mathbb{P}\left(y_{1:n}|x_{1:n}\right)=\prod_{t}\frac{e^{w^{T}\phi\left(x_{1:n},y_{t-1},y_{t},t\right)}}{Z}=\prod_{t}p\left(y_{t-1\rightarrow t}\right)$$

Thus $-E\left(x;\theta\right)=-E\left(y_{t}|y_{t-1},x_{1..n};\theta\right)=w^{T}\phi\left(x_{1:n},y_{t-1},y_{t},t\right)$. Therefore the energy is minus the product of the weights and the feature function.

2. For a general energy based model:

$$\log p\left(x\right)=\log\left(\frac{1}{Z}e^{-E\left(x;\theta\right)}\right)=-E\left(x;\theta\right)-\log Z=-\left(E\left(x;\theta\right)+\log\left(\sum_{x'}e^{-E\left(x';\theta\right)}\right)\right)\Rightarrow$$

$$\frac{\partial}{\partial\theta}\left(\log p\left(x\right)\right)=-\left(\frac{\partial E\left(x;\theta\right)}{\partial\theta}\right)-\frac{\partial}{\partial\theta}\left(\log\left(\sum_{x'}e^{-E\left(x';\theta\right)}\right)\right)=-\left(\frac{\partial E\left(x;\theta\right)}{\partial\theta}\right)-\frac{1}{\sum_{x'\sim P_{\theta}}e^{-E\left(x';\theta\right)}}\cdot\left(\frac{\partial}{\partial\theta}\left(\sum_{x'\sim P_{\theta}}e^{-E\left(x';\theta\right)}\right)\right)=$$

$$=-\left(\frac{\partial E\left(x;\theta\right)}{\partial\theta}\right)+\frac{1}{Z}\sum_{x'\sim P_{\theta}}\left(\frac{\partial E\left(x';\theta\right)}{\partial\theta}\right)e^{-E\left(x';\theta\right)}$$

Inputting the calculation into an expectation:

$$\underset{x\sim D}{\mathbb{E}}\left[\frac{\partial\log\left(p\left(x\right)\right)}{\partial\theta}\right]=\sum_{x\in D}\left(\frac{\partial\log\left(p\left(x\right)\right)}{\partial\theta}\cdot p\left(x\right)\right)=\sum_{x\in D}\left(\left(-\left(\frac{\partial E\left(x;\theta\right)}{\partial\theta}\right)+\frac{1}{Z}\sum_{x'\sim P_{\theta}}\left(\frac{\partial E\left(x';\theta\right)}{\partial\theta}\right)e^{-E\left(x';\theta\right)}\right)\cdot p\left(x\right)\right)=$$

$$\sum_{x\in D}\left(\frac{p\left(x\right)}{Z}\sum_{x'\sim P_{\theta}}\left(\frac{\partial E\left(x';\theta\right)}{\partial\theta}\right)e^{-E\left(x';\theta\right)}\right)-\sum_{x\in D}\left(p\left(x\right)\cdot\frac{\partial E\left(x;\theta\right)}{\partial\theta}\right)=\sum_{x\in D}p\left(x\right)\sum_{x'\sim P_{\theta}}\left(\frac{\partial E\left(x';\theta\right)}{\partial\theta}\right)\underbrace{\frac{e^{-E\left(x';\theta\right)}}{Z}}_{=p\left(x'\right)}-\underset{x\sim D}{\mathbb{E}}\frac{\partial E\left(x;\theta\right)}{\partial\theta}=$$

$$= \sum_{x\in D} p\left(x\right) \sum_{x'\sim P_\theta} \left(\frac{\partial E\left(x';\theta\right)}{\partial\theta}\right) p\left(x'\right) - \mathop{\mathbb{E}}_{x\sim D}\frac{\partial E\left(x;\theta\right)}{\partial\theta} = \sum_{x\in D} p\left(x\right)\left(\mathop{\mathbb{E}}_{x'\sim P_\theta}\frac{\partial E\left(x';\theta\right)}{\partial\theta}\right) - \mathop{\mathbb{E}}_{x\sim D}\frac{\partial E\left(x;\theta\right)}{\partial\theta}$$

$$= \mathop{\mathbb{E}}_{x'\sim P_\theta}\frac{\partial E\left(x';\theta\right)}{\partial\theta}\underbrace{\sum_{x\in D}p\left(x\right)}_{=1} - \mathop{\mathbb{E}}_{x\sim D}\frac{\partial E\left(x;\theta\right)}{\partial\theta} = \mathop{\mathbb{E}}_{x\sim P_\theta}\frac{\partial E\left(x;\theta\right)}{\partial\theta} - \mathop{\mathbb{E}}_{x\sim D}\frac{\partial E\left(x;\theta\right)}{\partial\theta}$$

as required.

0.4 MLE for HMM

S - num words, T - time = n

We'll use Lagrange multipliers to solve this problem. We wish to estimate $t_{i,j} = argmax_t\left\{l\left(S,\theta\right)\right\}$ subject to $\forall i \in \left[|P|\right] : \sum_{j=1}^{|P|} t_{i,j} = 1$ for P being the set of all possible PoS tags. This is because t is a probability function, and the sum of all transitions from i is 1. Define our new function: $1 - \sum_{j=1}^{|P|} t_{i,j}$ and install it into a new log likelihood function:

$$l'\left(S,\theta\right) = \sum_{i=1}^{|S|}\sum_{j=1}^{\left|x^i\right|}\log t_{y_{j-1},y_j} + \sum_{k=1}^{|P|}\lambda_k\left(1-\sum_{j=1}^{|P|}t_{y_k,y_j}\right) \Rightarrow \frac{\partial l'\left(S,\theta\right)}{\partial t_{y_{j-1},y_j}} = 0 \Rightarrow$$

$$\sum_{i=1}^{|S|}\frac{1}{t_{y_{j-1},y_j}} - \lambda_k = 0 \Rightarrow \lambda_k^{-1}\sum_{i=1}^{|S|}1_{y_{j-1}\rightarrow y_j} = t_{y_{j-1},y_j}$$

Where $1_{y_{j-1}\rightarrow y_j}$ is an indicator.

$$\frac{\partial l'\left(S,\theta\right)}{\partial\lambda_k} = 0 \Rightarrow \left(1-\sum_{j=1}^{|P|}t_{y_k,y_j}\right) = 0 \Rightarrow \sum_{j=1}^{|P|}t_{y_k,y_j} = 1$$

Substituting what we got from the first partial derivative:

$$\sum_{j=1}^{|P|}t_{y_k,y_j} = \sum_{j=1}^{|P|}\left(\lambda_k^{-1}\sum_{i=1}^{|S|}1_{y_{j-1}\rightarrow y_j}\right) = 1 \Rightarrow \lambda_k = \sum_{j=1}^{|P|}\sum_{i=1}^{|S|}1_{y_{j-1}\rightarrow y_j}$$

Installing the phrase above back into the first partial derivative:

$$\lambda_k^{-1}\sum_{i=1}^{|S|}1_{y_{j-1}\rightarrow y_j} = t_{y_{j-1},y_j} \Rightarrow t_{y_{j-1},y_j} = \frac{\sum_{i=1}^{|S|}1_{y_{j-1}\rightarrow y_j}}{\sum_{j=1}^{|P|}\sum_{i=1}^{|S|}1_{y_{j-1}\rightarrow y_j}} = \frac{\#\left(y_{j-1}\rightarrow y_j\right)}{\#\left(y_{j-1}\right)}$$

Since $j-1$ is the 'time' and not a specific we get that the phrase is the number of times we saw a transition $y_i \rightarrow y_j$ divided by the number of times we saw y_i transitioning to something, getting as required:

$$\hat{t}_{i,j} = \frac{\#\left(y_i\rightarrow y_j\right)}{\sum_k\#\left(y_i\rightarrow y_k\right)}$$

Practical Part

Sampling the HMM

Eventhough the sampling function returns sentences without any logic, we can still recognize a gramatical structure, for example: "The RARE_WORD Steinhardt has than opposition of cash on those football", or "they says the president exporter" - replacing words in these sentences with the same PoS taggings could make perfect sense ("They saw the president".

Testing the models

When training the baseline model on 10%, 25%, 90% of the data and testing on 10% I get an accuracy rate of 85%~, 88%~, 91%~ accordingly.
When training the HMM model on 10%, 25%, 90% of the data and testing on 10% I get an accuracy rate of 87%~, 91%~, 94%~ accordingly.
When training the MEMM model I encounter a problem - the runtime of it is way too long. Unfortunately, I managed to only train it on 10% of the data and get 33% accuracy, and failed to do so on the rest. However, while running the Perceptron algorithm I test the model every 1000 samples on 64 random samples. Doing so we can see that the model improves (not drasticly, since it is tested on only 64 samples), but still learns.