Assignment 2: HMM for the weekend

10th August, 2018

1 Introduction:

Several problems in NLP like Named Entity Recognition or Parts-of-Speech (POS) tagging can treated as sequence labeling tasks. Each token of a sequence (generally words) is assigned one particular label. The label of a particular token is often dependent on the labels of its surrounding tokens and thus cannot be treated as i.i.d . This particular assignment explores the task of sequence labeling using two statistical model - generative and discriminative.

2 Generative vs Discriminative

Assume we have a sequence of data points $x_1, x_2..x_n$ along with their corresponding labels $y_1, y_2..y_n$. In the simplest and crudest of definitions, a generative model learns the **joint** probability distribution $p(x, y)$, while the discriminative model learns the **conditional** probability distribution $p(y|x)$. The Figure 1 illustrates the difference between a generative and discriminative model.

Simple example [edit]

Suppose the input data is $x \in \{1, 2\}$, the set of labels for x is $y \in \{0, 1\}$, and there are the following 4 data points: $(x, y) = \{(1, 0), (1, 0), (2, 0), (2, 1)\}$

For the above data, estimating the joint probability distribution $p(x, y)$ from the empirical measure will be the following:

	$y = 0$	$y = 1$
$x = 1$	$1/2$	0
$x = 2$	$1/4$	$1/4$

while $p(y|x)$ will be following:

	$y = 0$	$y = 1$
$x = 1$	1	0
$x = 2$	$1/2$	$1/2$

Figure 1: Example shamelessly copied from Wikipedia to illustrate the basics of probability involved in understanding generative and discriminative models.

Assignment Task 1:

Identify which of the following are generative or discriminative models and justify: Neural networks, Naive Bayes classifier, Logistic regression, Gaussian Mixture model, GANs, LDA(Latent Dirichlet Allocation), SVM, Decision Tree.

3 Hidden Markov Model (HMM)

I am sure you all saw this coming. To spare you the intricate details, however, I would link you to this **tutorial**, which is as uncomplicated and archaic as the HMM model itself.

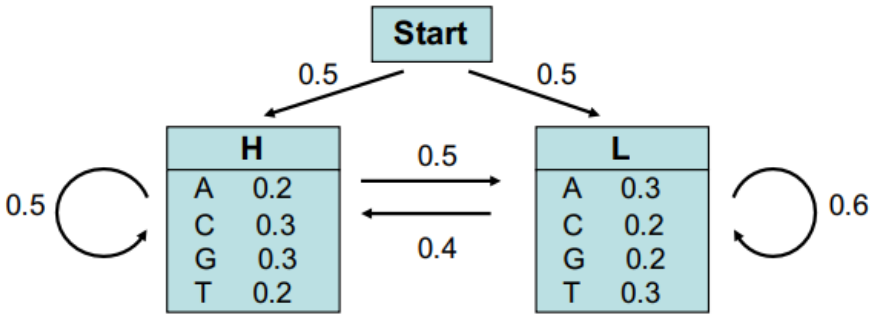

Figure 2: A sample HMM model

The toy HMM model of Figure2 is used to predict the region of coding DNA from a given sequence. It is composed of 2 states, H (high GC content) and L (low GC content).The H state characterizes coding DNA while L state encodes for non-coding DNA.

The transition, starting and emission matrix for the above model is :

$$P = \begin{bmatrix} 0.6 & 0.4 \\ 0.5 & 0.5 \end{bmatrix}$$

$$S = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix}$$

$$O = \begin{bmatrix} 0.3 & 0.2 & 0.2 & 0.3 \\ 0.2 & 0.3 & 0.3 & 0.2 \end{bmatrix}$$

This toy HMM model has been provided in the HMM.ipynb file along with this assignment. It implements an exhaustive search over the state space to identify the best labels for a given sequence. As obvious as it sounds, the search space will blow up with longer sequences. Consequently, you need to implement the Viterbi algorithm that solves this issue. Once, again I refrain from delving into the verbose details here but you would find wonderful illustration of the Viterbi algorithm for the same HMM model in this **link**.

Since a toy-model does not give you the thrill of real-world data, you will have to implement the HMM model on the tagged brown corpus. Further instructions on how to import the brown corpus are already given in the ipynb file.

Assignment Task 2: The crux of this assignment is to assign the POS-TAG to a given sequence of words. To do so, we will train the HMM model on the tagged brown corpus, except for the last but 100 sentences. We will then evaluate the POS-TAGS assigned to the last 10 sentences in the corpus.

- Build the transition, start and emission matrices from the training data. To be explicit, transition matrix is the probability of going from one POS-TAG to another, the start matrix is probability of a tag at the beginning of a sequence, while the emission matrix is the probability of seeing a word given the POS-TAG.
- Formulate the Viterbi algorithm on the given matrices. It is advisable to first check its correctness on the toy-example first.
- Using the HMM model, find the best sequence of POS-TAGs on the test data. Check the accuracy of the model on the actual POS-TAGS and report it.

4 Conditional Random Fields (CRF)

Before the advent of deep neural nets, CRFs were considered to be the state of art in sequence labeling tasks. Thus, it only feels natural to check the veracity of this claim and compare the performance of a simple CRF with respect to the vanilla HMM model.

CRF is a discriminative model and thus you need to cook up features to train on the labeled data. **Link** to learn CRF also follows for the inquisitive. We have also provided the skeleton for the crf-model in the ipynb file to ensure uniformity.

Assignment Task 3:

- Introduce features that into the CRF model (The bias feature is inserted by default). Each feature that you include needs to be supported with justification and should show improvement in performance of the model. For example if you decide to include the length of the word as a feature, I would be even happy with a crude reason like the distribution of length of words for nouns and determiners vary significantly.
- **Bonus:** You are also free to tweak the hyper-parameters in the train function but they should also be justified with experimental tests. If you say the regularizing parameter is to be kept at 0.1, show me a graph which justifies your claim.

5 Deliverables:

Since many of you wish to submit your ipynb file, submit it according to the format Assignment_2_YourRollNo.ipynb .

The ipynb file will not be considered as a report if it lacks the justifications and the answers to all subparts of the question.