Applied Machine Learning T1 Practical: HMMs

Date: 26th Feb, 2016 (Saturday)

Time: 1.00 pm to 5:00 pm IST

Problem : Build a classifier using HMMs to detect Cancer from Gene Expressions

The goal of this lab experiment is to help you learn how to build and use HMMs as classifiers for practical problems that are based on sequences as observables.

You are provided with a dataset which contains Gene Expressions that signify a possibility of procuring a certain type of Cancer (type is the file name). It also contains a file for testing the HMMs where each entry is of the form (tag, expression sequence) which you need to use to test accuracy. Along with this dataset, you are given an implementation of HMM in the form of a class along with some utility functions that will allow you focus on building the core classifier within the provided time.

The high level steps to build the classifier are as follows:

1. Write a file called ‘train.py’ that will create one HMM per disease. Select a number of internal states and initialize the HMM model while satisfying probability rules.
2. Make use of the util function to obtain the training data. Pickle files will be created which will need to be used when testing.
3. Since each of the inputs are streams of decimal values, our discrete HMMs would have to have each possibly infinite output symbols in the emission matrix which isn’t feasible. Therefore, we use a procedure called ‘Vector Quantization’ that does k-means clustering and gives a single integer for a specified length of input. (Note that the training data itself had varying sequence lengths for various types of cancer, so there is a dictionary that maps size of vector to their respective codebooks for conversion). Read the utils file carefully and use these accordingly when testing. When training, get the VQ sequence and pass it to the HMM training algorithm.
4. Pickle the trained HMMs. Take a look at the documentation of the pickle module in python, specifically pickle.dump and pickle.load.
5. Write a file called ‘test.py’ that loads the trained HMM pickles and codebooks. Read the test sequences and convert them to VQ sequences as follows:

```
import scipy.cluster.vq as sp

vq_seq = map(str,sp.vq(np.reshape(vecs,(n,size)),
codebooks[size])[0])
```

（Do the input conversion to VQ for all sizes and use the corresponding size for type of cancer obtained in the size mapping pickle to fill up probabilities over all types using the appropriate HMM algorithm)

Finally, pick the one with highest probability as the possible type and report the number of correct predictions made by your HMM based architecture.

There are only 15 samples in the test file. The challenge in this paper is not to achieve the highest accuracy, but to understand why you get these numbers from what you have implemented. Reason out in terms of the hyperparameters or data or whatever you may feel is a possible rationale. This exercise is to ensure that focus is on understanding very well whatever you implement.

Deliverables:

1. Source code (all the python files)
2. A post on the Facebook group that reports your observations in terms of correct predictions and how hyperparameters such as number of states for each HMM, number of distinct values that you configured to be output by the VQ phase (bins) affected this if it did at all. Articulate your learnings from the exercise.
3. Feedback on the exercise on a separate Facebook poll that will be put up.

Please zip your source code and mail to aml2016@googlegroups.com on time. Late Submissions will be penalized.

Best wishes from your faculty ☺