**ECE 8527: Introduction to
Machine Learning and Pattern Recognition**

# HW No. 11: Discrete Markov Models

Develop a discrete hidden Markov model (DHMM) that predicts the next letter in English text. Use the data here to train your DHMM:

*https://www.isip.piconepress.com/courses/temple/ece\_8527/resources/data/set\_06/*

Treat all whitespace (e.g., tab, space) as one symbol called SP. Ignore case (e.g., lowercase everything). Map numbers (e.g., 0-9) to a symbol called NM. Treat the remaining non-alphabetic characters as one symbol: if it is not in the range [a-z], map it to a symbol called PC. Therefore, you will have 26 letters plus three non-alphabetic symbols in your list of output symbols.

Start with a simple single-state model. Train the associated probabilities. Comment on what you observe. Discuss how you decided to terminate training.

Next, train a two-state fully ergodic model. Comment on the probabilities that the model learns. Does this better explain the data? How can you test that assertion?

Train three-state to ten-state ergodic models. Plot the log likelihood of the data given the model as a function of the number of states. Comment on what you observe.

Depending on your programming environment, there are various toolboxes available to do this. In MATLAB, we have used this toolbox in the past:

[*https://www.isip.piconepress.com/courses/temple/ece\_8527/resources/hmm\_matlab/*](https://www.isip.piconepress.com/courses/temple/ece_8527/resources/hmm_matlab/)

In Python, we have used these tools the general machine learning library *scikit-learn*. More information on this can be found at:

*http://scikit-learn.sourceforge.net/stable/modules/hmm.html*

As your models get more complex, be careful how you initialize them and what training recipe you use. Convergence of the model might not be so easy to achieve.