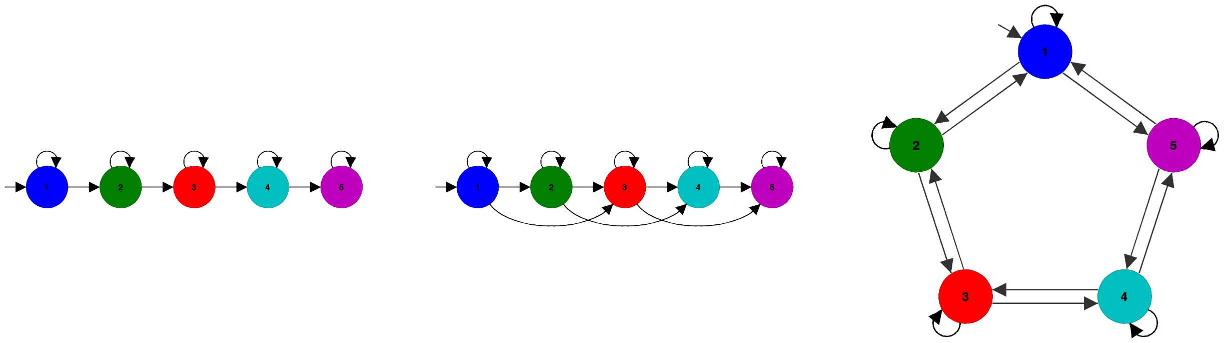
EE613 - Hidden Markov models - Exercises - Oct. 28, 2015

The main folder contains four examples `demo_HMM01.m`, `demo_HMM_Viterbi01.m`, `demo_HSMM01.m` and `demo_trajHSMM01.m`. These codes can be run either from Matlab or from GNU Octave. Each example loads a dataset and fits an HMM, an HSMM or a trajectory-HSMM to the data with an EM algorithm. First run the four examples, visualize the results and try to change the model parameters.

Exercise 1: Initialization of HMM parameters



- Based on `demo_HMM01.m`, initialize the transition probabilities and initial state distribution with parameters corresponding to the three different structures in the figure above. The probabilities can be set arbitrarily, by verifying that the specified values form a valid HMM with parameters that satisfy the required constraints.

- Use the function `m_fcts/EM_HMM.m` to re-estimate the HMM parameters, and display in the command prompt the resulting transition probabilities (stored as a matrix) and initial state distribution (stored as a vector). Do the models keep the same structure as the one set at initialization? Can you explain why?

Exercise 2: Likelihood computation in HMM

- Select a dataset (e.g., 5-10 motion samples to draw one of the alphabet letters in the `data` folder), and train an HMM.

- Create a function evaluating the likelihood that one of the trajectory in the training set belongs to the HMM. To do this, you can use the forward variable. One option is to exploit the forward variable at the end of the trajectory. You can take inspiration from `m_fcts/EM_HMM.m`, which is also using the forward variable in the E step of the EM procedure. However, note that the code uses a rescaled version of the forward and backward variables to avoid numerical underflow when estimating the parameters of the model, and that this rescaling should not be used when evaluating the likelihood of a trajectory. As expected, the likelihoods will become very low for long sequences (you can can consider shorter sequences if you reach the numerical limit of Matlab/Octave).

Exercise 3: Influence of HMM initialization for EM parameters estimation

- Train a left-right HMM as in the first image of the figure above, with Gaussian emission parameters initialized by

    1. k-means clustering (`init_GMM_kmeans` function in `demo_HMM01.m`),
    2. Temporally-ordered clusters with equal bin sizes (`init_GMM_kbins` function in `demo_HMM01.m`),

- Use the function created in *Exercise 2* to evaluate the average likelihood of the trajectories in the training set. What can we conclude? What happens if the null elements in the initial state distribution and transition probabilities are replaced by very small probabilities (e.g., 1E-10)?