# Asmt 8: Graphs

Turn in through Canvas by 5pm:
Wednesday, April 24
50 points

## Overview

In this assignment you will explore different approaches to analyzing Markov chains.

You will use one data sets for this assignment:

- `http://www.cs.utah.edu/~jeffp/teaching/cs5140/A8/M.dat`

These data sets are in matrix format and can be loaded into MATLAB or OCTAVE. By calling
`load filename` (for instance `load M.dat`)
it will put in memory the the data in the file, for instance in the above example the matrix `M`. You can then display this matrix by typing `M`

*As usual, it is highly recommended that you use LaTeX for this assignment. If you do not, you may lose points if your assignment is difficult to read or hard to follow. Find a sample form in this directory:* `http://www.cs.utah.edu/~jeffp/teaching/latex/`

## 1 Finding $q_*$ (100 points)

We will consider four ways to find $q_* = M^t q_0$ as $t \to \infty$.

**Matrix Power:** Choose some large enough value $t$, and create $M^t$. Then apply $q_* = (M^t)q_0$. There are two ways to create $M^t$, first we can just let $M^{i+1} = M^i * M$, repeating this process $t-1$ times. Alternatively, (for simplicity assume $t$ is a power of 2), then in $\log_2 t$ steps create $M^{2i} = M^i * M^i$.

**State Propagation:** Iterate $q_{i+1} = M * q_i$ for some large enough number $t$ iterations.

**Random Walk:** Starting with a fixed state $q_0 = [0, 0, \ldots, 1, \ldots, 0, 0]^T$ where there is only a 1 at the $i$th entry, and then transition to a new state with only a 1 in the $j$th entry by choosing a new location proportional to the values in the $i$th column of $M$. Iterate this some large number $t_0$ of steps to get state $q_0'$. (This is the *burn in period*.)

Now make $t$ new step starting at $q_0'$ and record the location after each step. Keep track of how many times you have recorded each location and estimate $q_*$ as the normalized version (recall $\|q_*\|_1 = 1$) of the vector of these counts.

**Eigen-Analysis:** Compute `eig(M)` and take the first eigenvector after it has been $L_1$-normalized.

**A (40 points):** Run each method (with $t = 1024$, $q_0 = [1, 0, 0, \ldots, 0]^T$ and $t_0 = 100$ when needed) and report the answers.

**B (20 points):** Rerun the Matrix Power and State Propagation techniques with $q_0 = [0.1, 0.1, \ldots, 0.1]^T$. For what value of $t$ is required to get as close to the true answer as the older initial state?

**C (24 points):** Explain at least one **Pro** and one **Con** of each approach. The **Pro** should explain a situation when it is the best option to use. The **Con** should explain why another approach may be better for some situation.

**D (8 points):** Is the Markov chain *ergodic*? Explain why or why not.

**E (8 points):** Each matrix $M$ row and column represents a node of the graph, label these from $1$ to $10$ starting from the top and from the left. What nodes can be reached from node $4$ in one step, and with what probabilities?

## 2 BONUS: Taxation (5 points)

Repeat the trials in part **1.A** above using taxation $\beta = 0.85$ so at each step, with probability $1 - \beta$, any state jumps to a random node. It is useful to see how the outcome changes with respect to the results from Question 1. Recall that this output is the *PageRank* vector of the graph represented by `M`.

Briefly explain (no more than 2 sentences) what you needed to do in order to alter the process in question 1 to apply this taxation.