Asmt 6: Graphs

Turn in a hard copy at the start of class:

Wednesday, April 30

10 points (but you can earn up to 20 points)

This is optional, and will be averaged into your grade only if it improves your grade

Overview

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

You will use two data sets for this assignment:


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:


1 Finding \( q^* \) (10 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 \( q^*_t = (M^t)q_0 \). There are two ways to
create \( M^t \), first we can just let \( M^{t+1} = M^t \times M \), repeating this process \( t - 1 \) times. Alternatively,
(for simplicity assume \( t \) is a power of 2), then in \( \log_2 t \) steps create \( M^{2^i} = M^i \times M^i \).

State Propagation: Iterate \( q_{i+1} = M \times q_i \) for some large enough number \( t \) iterations.

Random Walk: Starting with a fixed state \( q_0 = [00 \ldots 100]^T \) where there is only a 1 at the \( i \)th entry, and then
transition to a new state with only a 1 in the \( i' \)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 \( \text{eig}(M) \) and take the first eigenvector after it has been normalized.

A (4 points): Run each method (with \( t = 500 \), \( q_0 = [100 \ldots 0]^T \) and \( t_0 = 20 \) when needed) and report
the answers.

B (2 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 (4 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.

2 BONUS 1: Taxation (4 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.

3 BONUS 2: Graph Sparsification (6 points)
A (3 points): Consider the adjacency matrix $L$. Run the basic graph sparsification algorithm in L26.1 with $t = 2$. Report the new matrix representing the graph.

B (3 points): Explain how clustering on the new graph may differ from that on the old graph. What problems may occur? Would these persist on a large graph with a large value of $t$, and Why?