Asmt 6B: Graphs

Turn in through GradeScope by 1pm:
Wednesday, December 3
50 points

Overview

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

You will use one data sets for this assignment:

- `http://www.cs.utah.edu/~jeffp/teaching/DM/A6/M.csv`

As usual, it is recommended that you use LaTeX or another method which can properly display mathematical notation 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/`

You could also utilize an LaTeX template specifically created for this assignment. Click here.

1 Finding q_* (50 points)

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

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 ith entry, and then transition to a new state with only a 1 in the jth entry by choosing a new location proportional to the values in the ith 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 `LA.eig(M)` and take the first eigenvector after it has been L_1-normalized.

A (30 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 for estimating q_* as a vector.

B (10 points): Rerun the 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?

D (10 points): Is the Markov chain *ergodic*? Explain why or why not.

2 BONUS: Graph Embedding (2 point)

Use a method of your choice to embed the graph in 2 dimensions to draw it. It should show vertices and edges.