Assignment 5: Binomial and Normal Distribution; Markov Model

Assigned: Nov. 16
Due: Nov. 30

Problem 1

A. Write a function \texttt{CoinFlip(P,N,M)} that conducts \(M\) experiments as follows: There is an unfair coin, that comes up heads with probability \(P\) and tails with probability \(1 - P\). In each experiment, the coin is flipped \(N\) times, and the number of times it comes up heads is recorded. The function \texttt{CoinFlip} returns a vector \(\vec{V}\) of length \(N + 1\) where \(\vec{V}[I]\) is the fraction of experiments in which the coin came up heads \(I - 1\) times (this inconvenient shift of 1 is necessary because MATLAB uses 1-based indexing on arrays).

For example if \(M = 100\), \(N = 2\), \(P = 0.5\) and there are 22 experiments that come up two tails, 59 that come up one head and one tail, and 19 that come up two heads, return the vector \([0.22, 0.59, 0.19]\)

Programming hint: Do not carry out \(M \cdot N\) calls to \texttt{rand}. Rather, begin by constructing a \(M \times N\) random array of 1’s and 0’s using the call \texttt{rand(M,N) < P}, and then use matrix operations as far as possible.

B. Write a function \texttt{PlotCoinFlip(P,N,M)} which shows, on a single plot:

- The result of a call to \texttt{CoinFlip(P,N,M)}
- The binomial distribution \(C(N, I - 1)P^{I-1}(1 - P)^{N+1-I}\)
  - Note: The Matlab function \texttt{nchoosek(N,K)} computes \(C(N, K) = N!/K!(N - K)!\)
- The normal distribution
  \[
  \frac{1}{\sqrt{2\pi}\sigma}e^{-\left(x-(NP+1)\right)^2/2\sigma^2}
  \]
  where \(\sigma^2 = NP(1 - P)\)
  from \(x = 1\) to \(x = N + 1\).

Problem 2

Write a function \texttt{PageRank(A,E)} that computes page rank based on links, as described in the class notes. The input parameter \(A\) is an \(N \times N\) adjacency matrix; \(A(I,J) = 1\) if page \(I\) has a link to page \(J\). The input parameter \(E\) corresponds to the probabilistic parameter \(\epsilon\) as described in the notes.

Ignore any self-loops. Treat a page with no outlinks as if it had an outlink to every other page.

You should implement algorithm PageRank2 (p. 6 of the notes). Do as much as possible with matrix operations, keeping the use of explicit loops to a minimum.