Seth Ely
seel6470
CSPB 5753, Fall 2024

Lab 8

Markov Chains

You have been given a lot of flexibility for your implementation of a predictive paging algorithm for PA4. In this lab, we will introduce *Markov chains*, a stochastic model that you might find useful for deciding which pages to page in/out. Regardless of whether you decide to use a Markov chain for PA4, they are of wide interest in both theory and practice.

Matrix Multiplication

Let A be a $\ell \times m$ matrix and B be a $m \times n$ matrix:

$$A = \begin{pmatrix} a_{1,1}, a_{1,2}, \cdots, a_{1,m} \\ a_{2,1}, a_{2,2}, \cdots, a_{2,m} \\ \vdots \;, \;\vdots \;, \ldots, \;\vdots \\ a_{\ell,1}, a_{\ell,2}, \cdots, a_{\ell,m} \end{pmatrix}, \qquad B = \begin{pmatrix} b_{1,1}, b_{1,2}, \cdots, b_{1,n} \\ b_{2,1}, b_{2,2}, \cdots, b_{2,n} \\ \vdots \;, \;\vdots \;, \ldots, \;\vdots \\ b_{m,1}, b_{m,2}, \cdots, b_{m,n} \end{pmatrix}.$$

The multiplication AB gives a $\ell \times n$ matrix defined such that the (i, j)-entry is

$$(AB)_{i,j} = \sum_{k=1}^{n} a_{i,k} b_{k,j}.$$

Convince yourself that you essentially already wrote multithreaded code for performing AB. In particular, extend your code to perform AB. What quantity should each thread should be in charge of computing?

Walks and Matrix Multiplication

Recall from your course on algorithms that we can represent a directed graph $G = (V, E)$ as a $n \times n$ adjacency matrix A such that $(A)_{i,j} = 1$ if (i, j) is an arc, and is 0 otherwise.

1. What does the (i, j)-entry of A^2 compute? (i, j) of A^2 represents how many paths from node i to node j exist that use 2 steps to get there (with one node between i and j).
2. For any positive integer k, what does the (i, j)-entry of A^k compute?
   (i, j) of A^k represents how many paths exist from i to j that use k steps to get from i to j.

Markov Chains

A *Markov chain* M can be seen as a generalization of our adjacency matrix. It is defined such that $M_{i,j}$ is the *probability* of going from i to j in a single step. We weight the arcs of our digraph $G = (V, E)$ with non-negative real numbers such that

$$\sum_{j \in V} M_{i,j} = 1 \quad \text{for each } i \in V \quad (\textit{why?})$$

where $M_{i,j} = 0$ if $(i, j) \notin E$. In other words, a $n \times n$ matrix is a Markov chain if its entries are all non-negative and each row sums to 1.

1. For any positive integer k, is M^k a Markov chain?
2. What does the (i, j)-entry of M^k compute?

Finally, think about how you could use a Markov chain as part of a predictive paging algorithm. How do you define M? What are its probabilities? How do we compute them? What do powers of M tell us?

1. Yes, since the definition of a Markov chain requires the rows to sum to 1, we know that each dot product between a row of M and a column of M must also sum to one. This is because the row of M (which sums to 1) is used to find the dot product of a column (which doesn't necesarrily sum to 1). Since this is performed for each column of M for a given row, the weighted multiplication will also yield a row that sums to 1.

Additionally, since all probabilities will alway be between 0 and 1 inclusive, the multiplication between them will always yield a nonnegative value between 0 and 1 inclusive.

2. The (i, j) entry of M^k represents the probability of state i transitioning to state j in k steps.

For the paging algorithm, we could use a Markov chain M where each index (i, j) represents the probability of a process transitioning from page i to page j. We could calculate each probability by first initializing the matrix to have all equal probabilities. After this, any page that is loaded for the process would update the row and column representing the previous page and current page since a transition has occured from page i to page j. This would require a way to track the previous page, assigning the current page as the previous page after updating the matrix, allow the next page call to be able to reference what page was used before it.

Doing so, each row would represent the ratio of pages used after the current page and we could predictively swap in a page to anticipate what this process might need by finding the max value in the row. Furthermore, we could create a cutoff value where only pages with a high probability would be swapped in to minimize our chances of paging in a page that will go unused by the process.

For this scenario, powers of M would represent a Markov matrix with probabilities showing for any M^k and index (i, j) what the probabilities will be to transition from page i to page j in k steps. This could be useful for predicting the next two or three pages that might be used, or conversely the next two or three pages that might not be as useful (taking the minimum probability in the row).