Digraphs and Matlab

David Arnold

August 31, 1996

Abstract

Digraphs are introduced along with their adjacency matrix. An algorithm is proposed for counting paths of length n.

Prerequisites. Matrix multiplication. You also need to have completed the activity on graphs before proceeding to this acitivity on digraphs.

1 Digraphs

A *digraph* is a graph whose edges are directed. In the case of a digraph, you can think of the connections as one-way streets along which traffic can flow only in the direction indicated by the arrow.

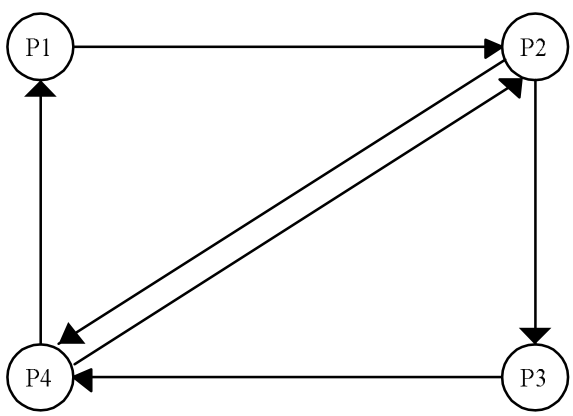

Figure 1. A digraph has *directed* edges.

The *adjacency* matrix for a digraph has a definition similar to the definition of an adjacency matrix for a graph.

$$a_{ij} = \begin{cases} 1 & \text{if there is a } \textit{directed} \text{ edge connecting } P_j \text{ to } P_i. \\ 0 & \text{otherwise} \end{cases}$$

Thus, the adjacency matrix for the digraph in Figure 1 is

$$A = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

Note that matrix A is not symmetric with respect to its main diagonal. This is so because the connections are not two-way streets. A directed edge connecting P_j to P_i does not require the existence of a directed edge connecting P_i to P_j. Use Matlab to show that A^T *does not equal* A.

```
>> A=[0 0 0 1;1 0 0 1;0 1 0 0;0 1 1 0]

A =

     0     0     0     1
     1     0     0     1
     0     1     0     0
     0     1     1     0

>> A'

ans =

     0     1     0     0
     0     0     1     1
     0     0     0     1
     1     1     0     0
```

1.1 Counting the Paths of Length N

The algorithm for counting the number of paths of length N from P_j to P_i in a graph works equally well for digraphs.

Let A be the adjacency matrix for a digraph. If $a_{ij}^{(N)}$ is the element in the ith row, jth column of matrix A^N, then $a_{ij}^{(N)}$ is the number of paths of length N from P_j to P_i.

For example, use Matlab to compute the fourth power of the adjacency matrix for the digraph in Figure 1.

```
>> A^4

ans =

     1     1     1     2
     2     2     1     3
     1     2     1     1
     1     3     2     2
```

Let $a_{24}^{(4)}$ represent the entry in row 2, column 4 of the matrix A^4. Note that $a_{24}^{(4)} = 3$. Three paths of length 4 from P_4 to P_2 follow.

- $P_4 \to P_2 \to P_3 \to P_4 \to P_2$
- $P_4 \to P_2 \to P_4 \to P_1 \to P_2$
- $P_4 \to P_1 \to P_2 \to P_4 \to P_2$

2 Homework

1. All the streets in the downtown area of Trapsville run one-way, either west to east or north to south as indicated in Figure 2. Therefore, traffic can only move to the right and down in the grid of streets (Somebody needs to fire the city manager.). The nodes of the graphs are intersections.

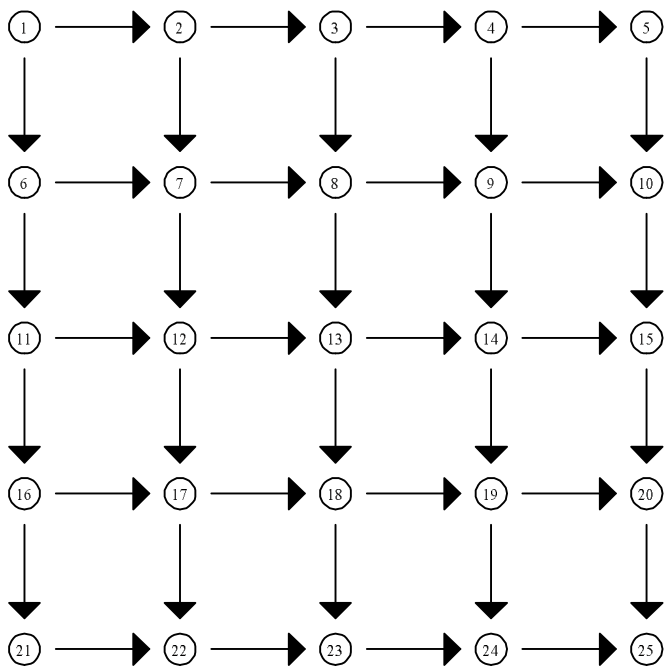

Figure 2. Downtown Trapsville.

Your task is to find the number of paths of length 8 from intersection #1 to intersection #25.

(a) Create the adjacency matrix A for the digraph in Figure 2. Note that A has 25 rows and 25 columns (There are 25 nodes.).

(b) Enter the adjacency matrix A in Matlab. *Hint:* The adjacency matrix in this case is an example of a *sparse* matrix. Most of the elements of a sparse matrix are zeros with comparatively few non-zero entries.

It is probably easiest to create an M-file that builds the adjacency matrix. Click on File-New-Mfile to open the editor. Save the file in a directory in your Matlab path as `digraphs.m`.

Your M-file should first create a matrix of zeros. Then the file should edit the adjacency matrix: if there is an edge connecting intersection $\#j$ to intersection $\#i$, place a 1 in the ith row and jth column. Copy the following statements into the file `digraphs.m`. Note the use of Matlab's transpose operator (').

```
A=zeros(25);
A([2 6],1)=[1 1]';
A([3 7],2)=[1 1]';
A([4 8],3)=[1 1]';
A([5 9],4)=[1 1]';
A(10,5)=1;
A([7 11],6)=[1 1]';
A([8 12],7)=[1 1]';
A([9 13],8)=[1 1]';
A([10 14],9)=[1 1]';
A(15,10)=1;
A([12 16],11)=[1 1]';
A([13 17],12)=[1 1]';
A([14 18],13)=[1 1]';
A([15 19],14)=[1 1]';
A(20,15)=1;
A([17 21],16)=[1 1]';
A([18 22],17)=[1 1]';
A([19 23],18)=[1 1]';
A([22 24],19)=[1 1]';
A(25,20)=1;
A(22,21)=1;
A(23,22)=1;
A(24,23)=1;
A(25,24)=1
```

Save the file as `digraphs.m`. Return to the Matlab command window and type the following at the Matlab prompt.

```
>>digraphs
```

(c) Use Matlab to compute A^8. How many paths of length 8 are there from intersection #1 to intersection #25?

2. Suppose that there is a riot taking place on the street connecting intersection #8 to intersection #13. Police have blocked off access to this street

so that no one may use it. How many paths of length 8 are there now from intersection #1 to intersection #25?

3. It is interesting to count the number of floating point operations[1] required to compute A^8.

   (a) Enter the following commands at the Matlab prompt.

   ```
   >> digraphs
   >> flops(0)
   >> A^8
   >> flops
   ```

   How many floating point operations are required to compute A^8.

   (b) Because the matrix A is sparse (mostly zeros), you can probably guess that there is a great deal of wasted resources when computing A^8. Matlab can economize resources when working with sparse matrices. You can greatly speed up the computation of A^8 by entering the following commands at the Matlab prompt[2].

   ```
   >> digraphs
   >> S=sparse(A)
   >> flops(0)
   >> S^8
   >> flops
   ```

   How many floating point operations are required to compute S^8? Surprised?

   (c) Repeat parts (a) and (b) for A^{1000} and S^{1000}.

[1] Without getting too technical, addition, subtraction, multiplication, division, and exponentiation each require a certain number of cycles of the CPU (central processing unit) on your computer. These are often referred to as floating point operations. Type `help flops` at the Matlab prompt to get a short summary of how floating point operations are counted.

[2] The true value of sparse matrices is realized when you have a matrix of such large size that the calculation of A^8 eats up all of your memory before completing.