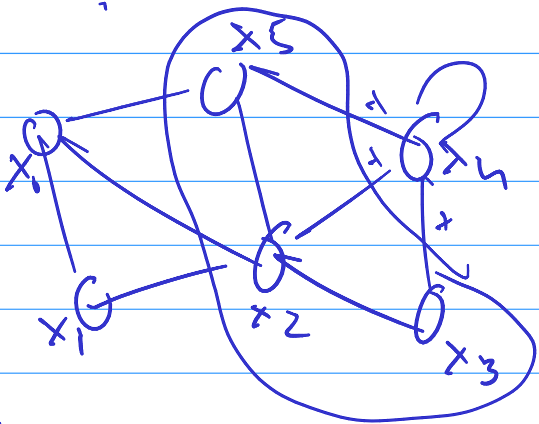
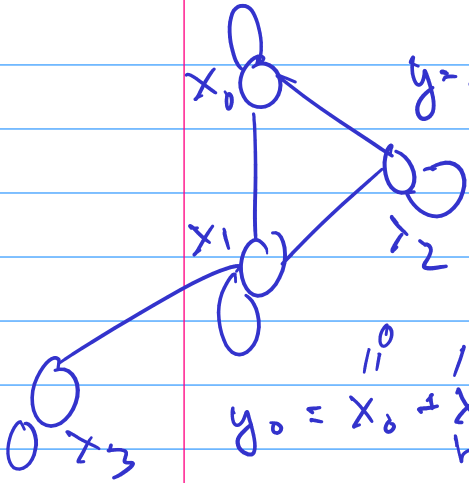
$(Ax)_i \leq 1$ for all i

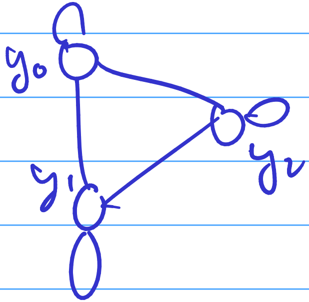


$y = Ax$

$y = x + Ax$



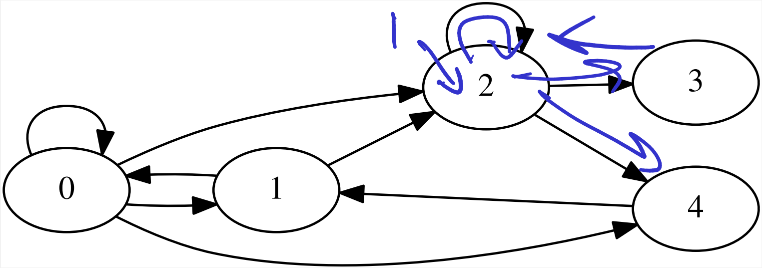
$y = Ax$ $\Rightarrow$



$$y_0 = \overset{0}{\overset{\shortparallel}{x_0}} + \underbrace{\overset{1}{\overset{\shortparallel}{x_1}} + \overset{1}{\overset{\shortparallel}{x_2}}}_{\text{neighbors}}$$

Graph Matrices and Matrix-Vector Multiplication

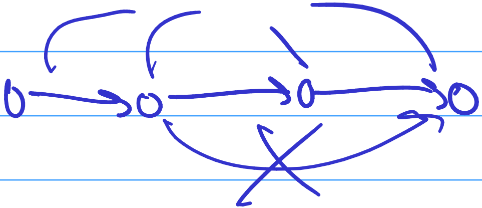
If we multiply a graph matrix by the ith unit vector, what happens?

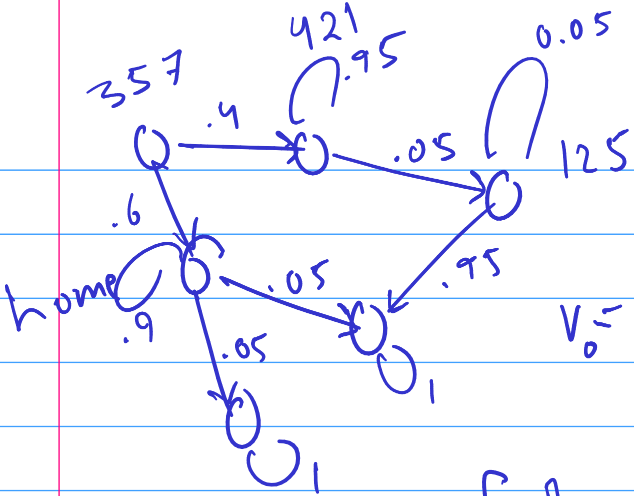


$$\begin{bmatrix} 1 & 1 & & & \\ 1 & 0 & & & 1 \\ 1 & 1 & 1 & & \\ & & 1 & 0 & \\ 1 & & 1 & & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

Markov processes

events





$$V_0 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ \vdots \end{bmatrix}$$

$$V_1 = A \cdot V_0$$

$$V_1 = \begin{bmatrix} 0 \\ .4 \\ .6 \\ 0 \\ 0 \end{bmatrix} \Rightarrow V_2 = Av_1$$

$$V_i = Av_{i-1}$$

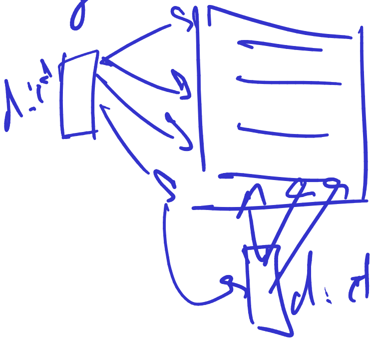
Demo: Matrices for Graph Traversal

Outline

Storing Sparse Matrices

Some types of matrices (including graph matrices) contain many zeros.
Storing all those zero entries is wasteful.
How can we store them so that we avoid storing tons of zeros?

Python dictionary of dictionary



$\{v_1 \to \{v_3, v_5\},$
$v_2 \to \{v_3, v_4\},$
$v_3 \to \{v_1, v_3\}\}$

Coordinate format (COO)

$$\big(\underbrace{(i,j)}_{\text{coordinate}},\ \underbrace{A_{ij}}_{\text{value}}\big)\ \forall A_{ij} \neq 0$$

Storing Sparse Matrices Using Arrays

How can we store a sparse matrix using just arrays? For example:

$$
\begin{array}{c c}
0\text{th} & \multirow{4}{*}{$\begin{pmatrix} 0 & 2 & 0 & 3 \\ 1 & 4 & & \\ & & 5 & \\ 6 & & & 7 \end{pmatrix}$} \\
1\text{st} & \\
2\text{nd} & \\
 &
\end{array}
$$

CSR: Compressed Sparse Row

values: $\{0, 2, 0, 3, 1, 4, 5, 6, 7\}$

cols: $\{0, 1, 2, 3, 0, 1, \boxed{2}, 0, 3\}$

row_st: $\{0, 4, 6, 7\}$

$$V = A \cdot W$$

```
for i in 0 to n-1
    for j in 0 to row_st[i] - row_st[i-1]
        V[i] = V[i] + values[row_st[i-1] + j] · W[cols[j]]
```

$$V[i] \mathrel{+}= \text{values}[\text{row\_st}[i-1] + j] \cdot W[\text{cols}[\text{row\_st}[i-1] + j]]$$

m - values

n - rows

$$\text{storage} = \underbrace{2m}_{\text{values \& cols}} + \underbrace{n}_{\text{row\_st}}$$

Demo: Sparse Matrices in CSR Format