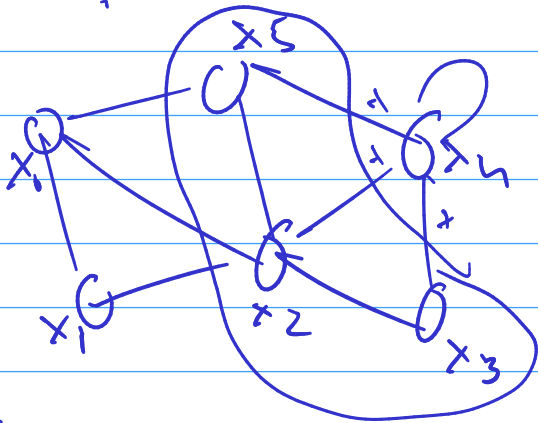
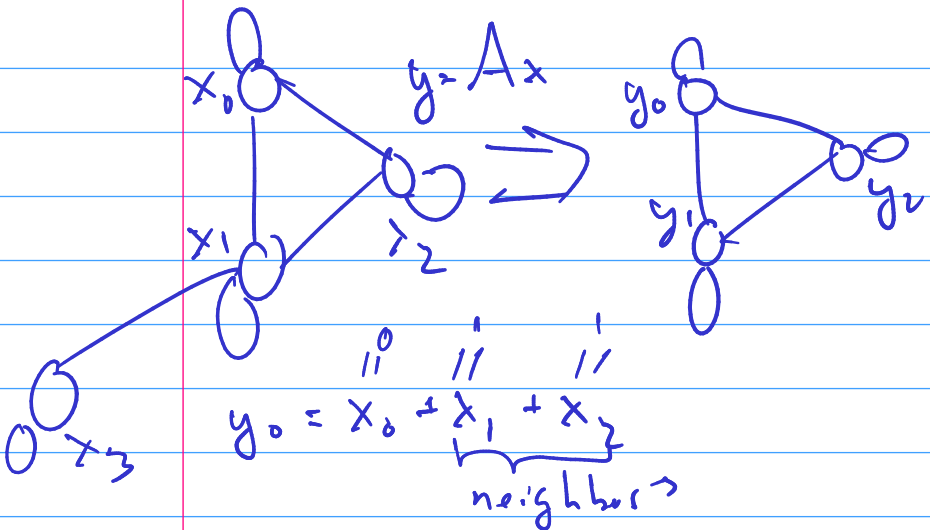


$(Ax)_i \leq 1$  for all  $i$ !



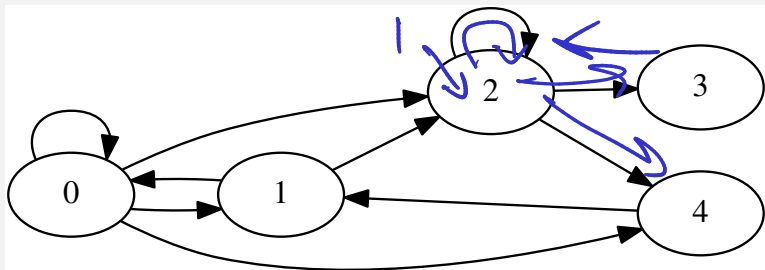
$$y = Ax$$

$$\underline{y = x^T A x}$$



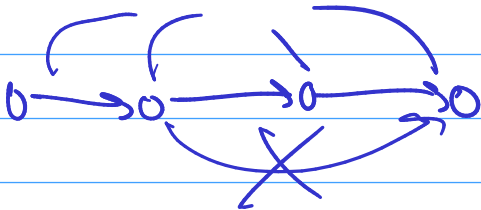
# Graph Matrices and Matrix-Vector Multiplication

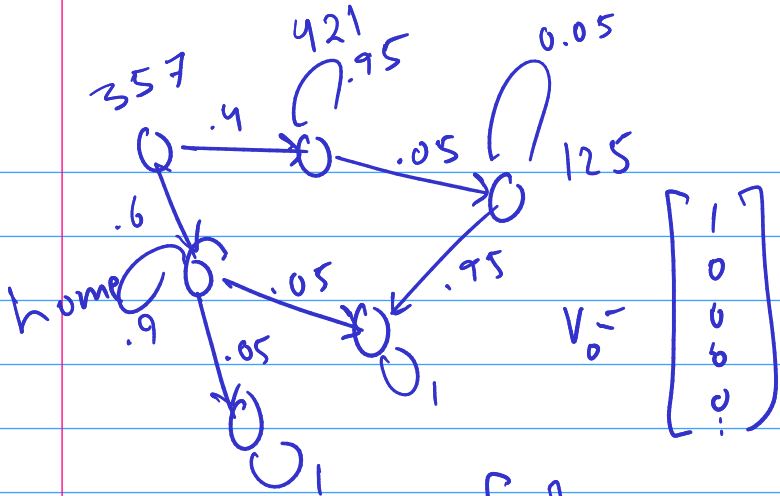
If we multiply a graph matrix by the  $i$ th unit vector, what happens?



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

Markov processes  
events





$$V_1 = A \cdot V_0 \quad V_1 = \begin{bmatrix} 0 \\ .5 \\ .6 \\ 0 \\ 0 \end{bmatrix} \Rightarrow V_2 = A V_1$$

$$V_i = A V_{i-1}$$

## Demo: Matrices for Graph Traversal

# Outline

Python, Numpy, and Matplotlib  
Making Models with Polynomials  
Making Models with Monte Carlo

Error, Accuracy and Convergence  
Floating Point

## Modeling the World with Arrays

The World in a Vector

What can Matrices Do?

Graphs

Sparsity

Norms and Errors  
The 'Undo' Button for Linear Operations: LU

LU: Applications

Linear Algebra Applications

Interpolation

Repeating Linear Operations:  
Eigenvalues and Steady States

Eigenvalues: Applications

Approximate Undo: SVD and Least Squares

SVD: Applications

Solving Funny-Shaped Linear Systems

Data Fitting

Norms and Condition

Numbers

Low-Rank Approximation

Iteration and Convergence

Solving One Equation

Solving Many Equations

Finding the Best: Optimization in 1D

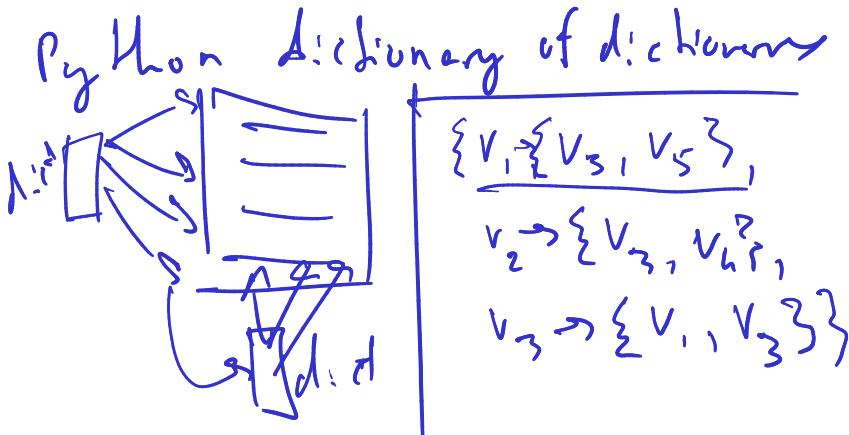
Optimization in  $n$  Dimensions

# 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?





Coordinate format (COO)

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

# Storing Sparse Matrices Using Arrays

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

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

CSR: Compressed Sparse Row

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

$$v = A \cdot w$$

for  $i$  in  $0$  to  $n-1$   
for  $j$  in  $0$  to  $\text{row\_st}[i] - \text{row\_st}[i-1]$

$$v[i] = v[i] + \text{values}[\text{row\_st}[i] + j] \cdot w[\text{cols}[i] + j]$$

$$v[i] += \text{values}[\text{row\_st}[i] + j]$$

$m$  - values  
 $n$  - rows

$$\text{storage} = 2m + n$$

$$\text{values}^{\text{rows}} \cdot w^{\text{row-st}}$$

## Demo: Sparse Matrices in CSR Format