Author recognition

Prof. Noah Snavely
CS1114
http://cs1114.cs.cornell.edu

Cornell University
Computer Science

Administrivia

- Quiz 5 this Thursday, 4/23
  - Focus on Markov chains
- A6 released, due on Friday
  - There will be demo sessions
  - You will also turn in your code this time
- Prelim 3 next Thursday, 4/30 (last lecture)
  - Will be comprehensive, but focus on most recent material

Administrivia

- Final projects
  - Due on Friday, May 8 (one big demo session)
  - Other CS faculty may come by
- The proposals look great!

What's the difference...

... between

```
% A is a cell array
A(1)
```

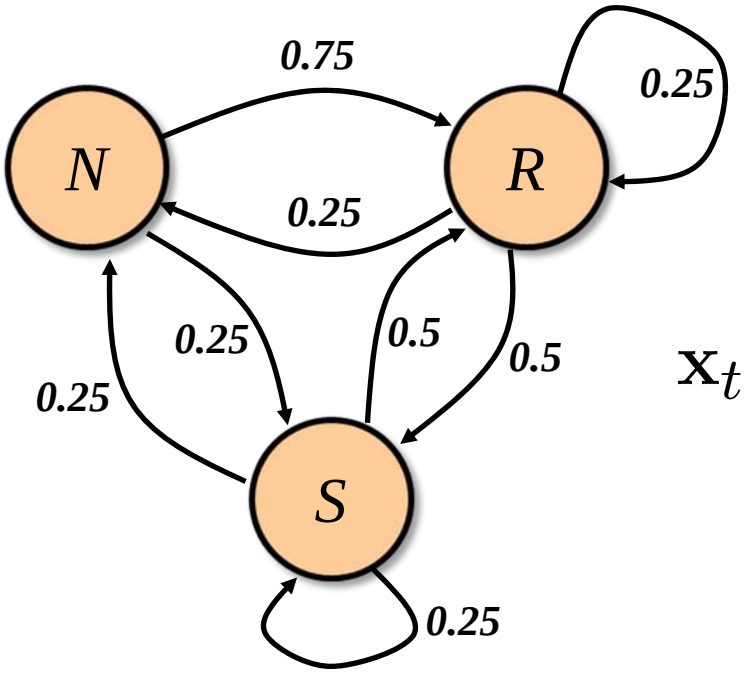
... and ...

```
A{1}
```

?

Markov chains

- Example: Springtime in Ithaca
- We can represent this as a kind of graph
- (N = Nice, S = Snowy, R = Rainy)



$$
\begin{array}{c} \\ \mathbf{x}_{t-1} \end{array}
\begin{array}{c}
\begin{array}{ccc} \boldsymbol{N} & \boldsymbol{R} & \boldsymbol{S} \end{array} \\
\begin{array}{c} \boldsymbol{N} \\ \boldsymbol{R} \\ \boldsymbol{S} \end{array}
\begin{bmatrix} 0.0 & 0.75 & 0.25 \\ 0.25 & 0.25 & 0.5 \\ 0.25 & 0.5 & 0.25 \end{bmatrix}
\end{array}
\quad \mathbf{x}_t
$$

Transition probabilities

Author recognition

- Simple problem:

  Given two Markov chains, say *Austen* (*A*) and *Dickens* (*D*), and a string *s* (with *n* words), how do we decide whether *A* or *D* wrote *s*?

- Idea: For both *A* and *D*, compute the probability that a random walk of length *n* generates *s*

Probability of a sequence

- What is the probability of a given *n*-word sequence *s*?

$$s = s_1\ s_2\ s_3\ \ldots\ s_n$$

- Probability of generating *s* = the product of transition probabilities:

$$\Pr(S_1 = s_1)\Pr(S_2 = s_2|S_1 = s_1)\Pr(S_3 = s_3|S_2 = s_2)\ldots\Pr(S_n = s_n|S_{n-1} = s_{n-1})$$

Probability that a sequence starts with s_1

Transition probabilities

(we'll ignore this for now)

Likelihood

- Compute this probability for *A* and *D*

$$\Pr(s|A)$$

"likelihood" of A

$$\Pr(s|D)$$

"likelihood" of D

$$\Pr(s|A) > \Pr(s|D)$$

Jane Austen wrote *s*

$$\Pr(s|A) < \Pr(s|D)$$

Charles Dickens wrote *s*

$$\Pr(s|A) = \Pr(s|D)$$

???

Problems with likelihood

1. Most strings of text (of significant length) have probability zero
   - Why?

2. Even if it's not zero, it's probably extremely small
   - What's 0.01 * 0.01 * 0.01 * … (x200) … * 0.01?
   - According to Matlab, zero

- How can we fix these problems?

	a	dog	is	man's	best	friend	it's	eat	world	out	there	.
a		**2/3**		**1/3**								
dog			**1/3**					**1/3**	**1/3**			
is	**1**											
man's					**1**							
best						**1**						
friend												**1**
it's	**1**											
eat		**1**										
world										**1**		
out											**1**	
there												**1**
.							**1**					

Pr("is dog man's best friend") = 0

Bigger example

	it	was	the	best	of	times	worst	…	birthday	…	far	better
it	0.004	0.17	0.005		0.002							0.002
was	0.004		0.06		0.004						0.001	
the				0.003			0.002					0.002
best					0.26							
of	0.017		0.23			0.001						
times	0.04					0.04						
worst					0.47							
…												
birthday					0.5							
…												
far											0.025	0.025
better					0.036							

13253 rows

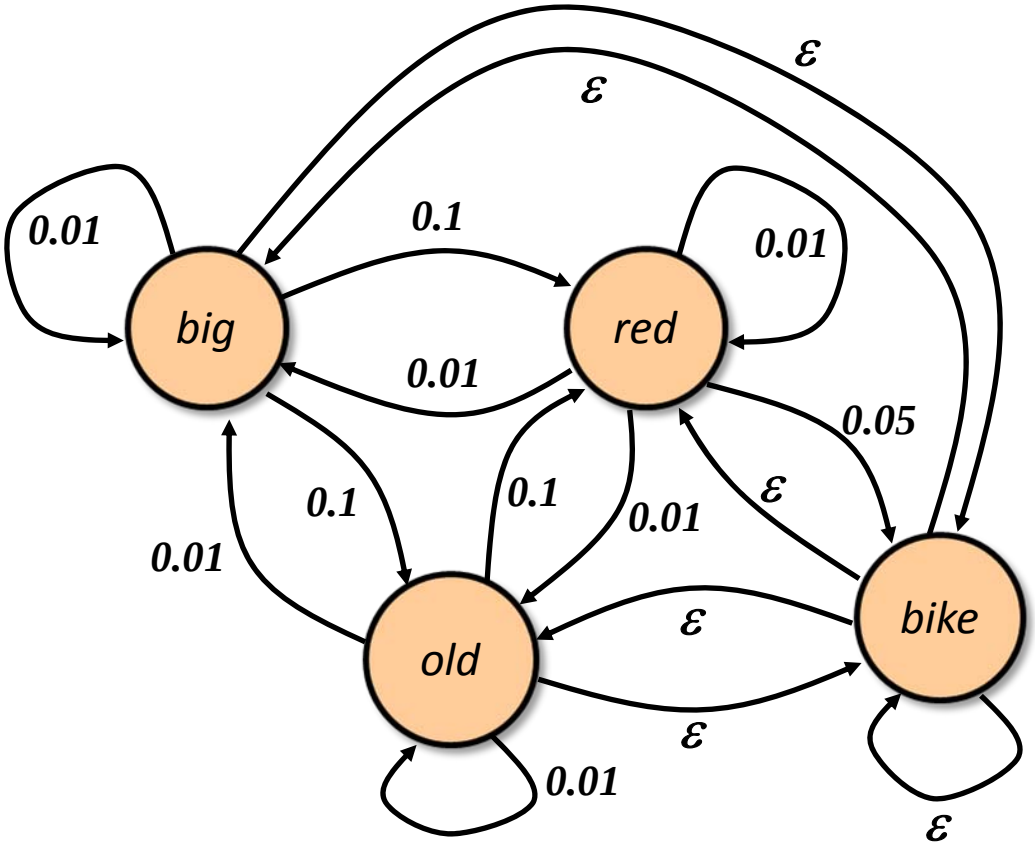
13253 cols

Handling zeroes

- We don't want to give every string with a new word / transition zero probability

- Several possibilities to consider:
  1. Transition from a known word to an new word
  2. Transition from a new word to a new word
  3. Transition from a new word to a known word
  4. Transition from a known word to a known word (unseen transition)

Handling zeroes



Trained Markov chain (in part)

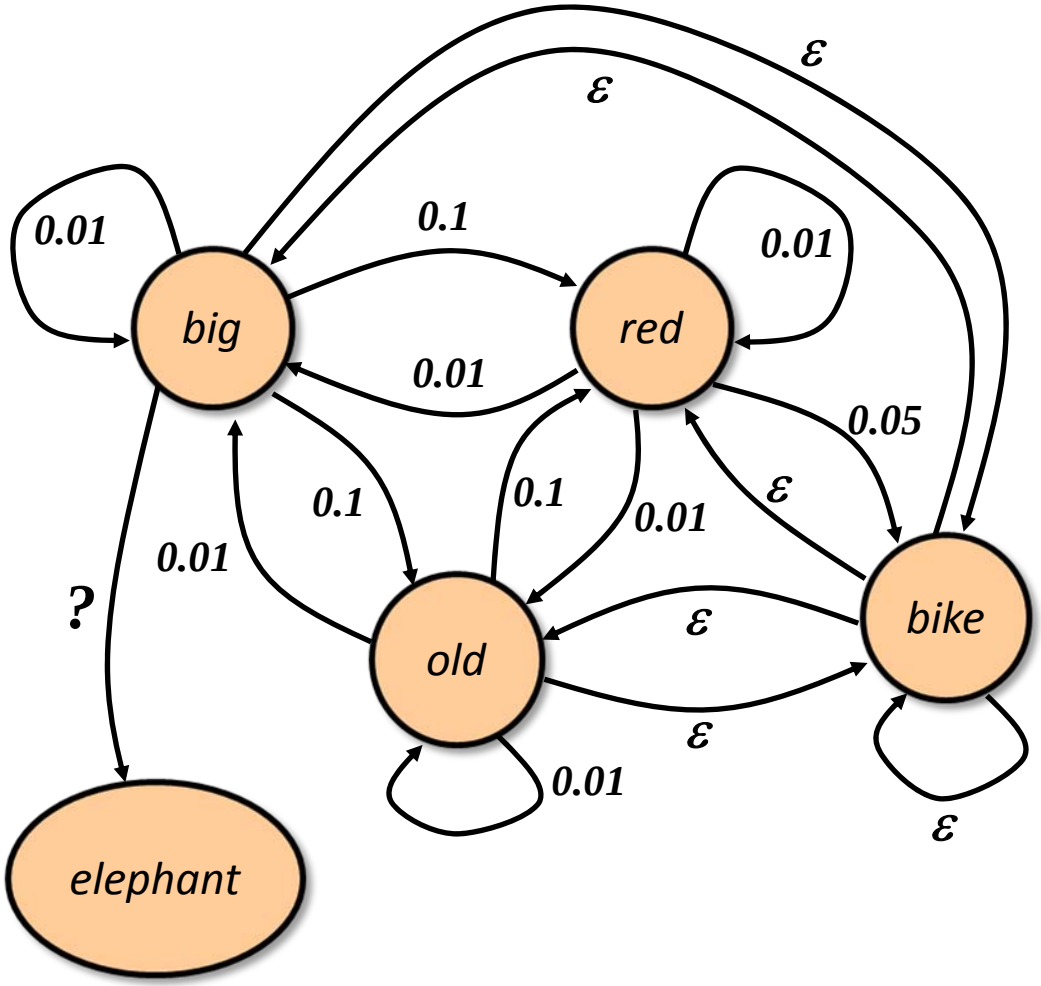
Test text: ***"… big bike …"***

The probability of generating this string with this Markov chain is zero

Idea: we'll add a small probability ε of any unobserved transition

(Reminiscent of PageRank)

Handling zeroes

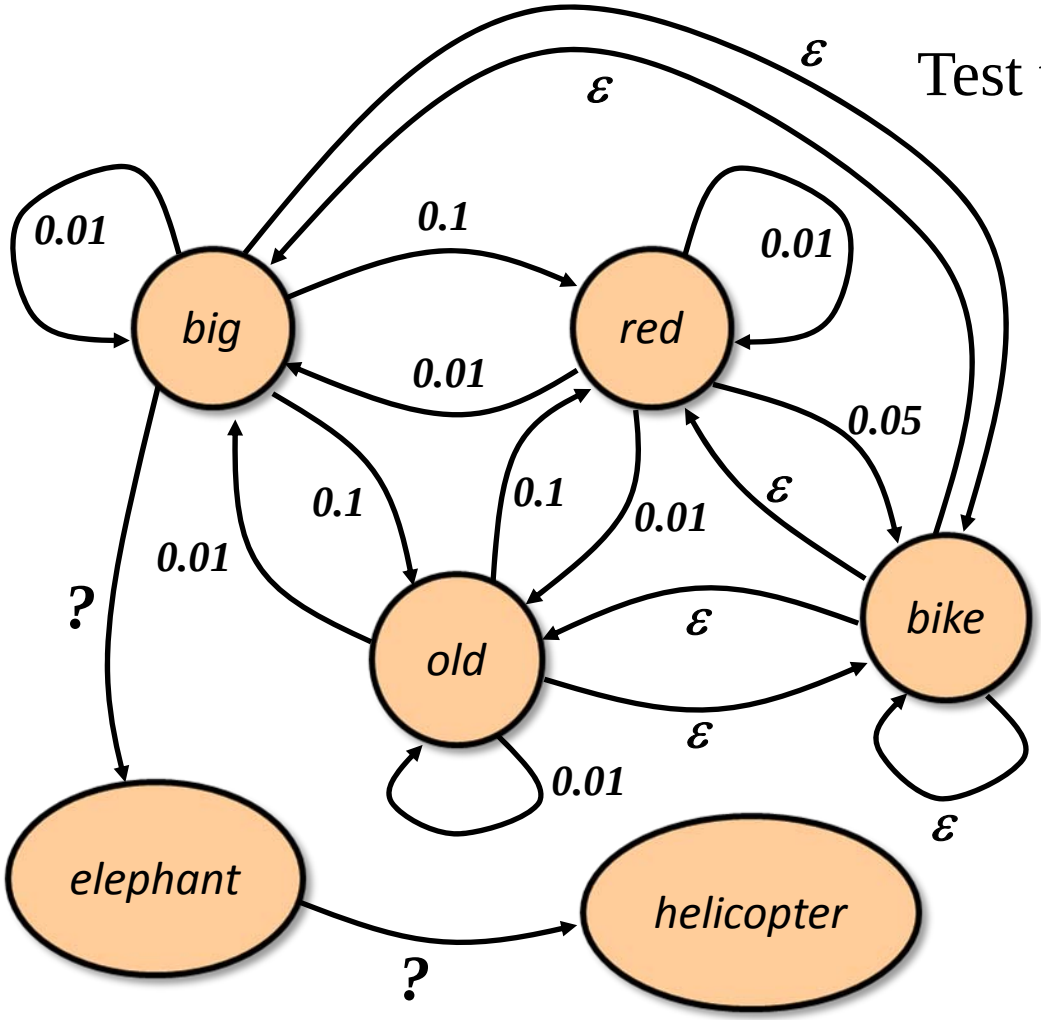


Test text: **"… *big elephant* …"**

We didn't see "elephant" in the training text

What should be the probability of a transition from "big" → "elephant"?

Handling zeroes

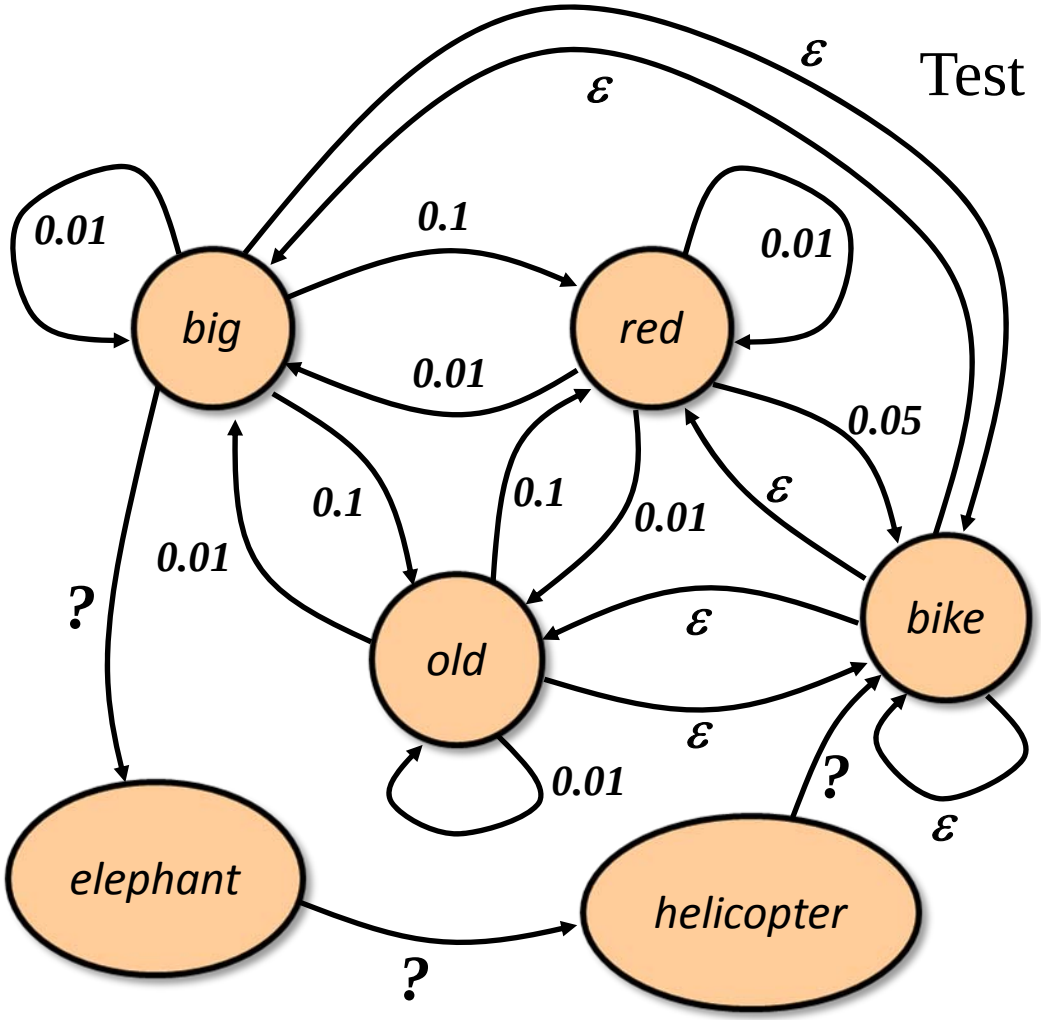


Test text: **"… *elephant helicopter*…"**

We didn't see "elephant" or "helicopter" in the training text

What should be the probability of a transition from "elephant" → "helicopter"?

Handling zeroes



Test text: ***"… helicopter bike …"***

We didn't see "helicopter" in the training text

What should be the probability of a transition from "helicopter" → "bike"?

Handling very low probabilities

- There's a smallest (positive) number that Matlab can store (why?)

```
>> realmin
ans =
   2.2251e-308
```

- Pretty small (the size of an electron is 10^{-15} m)
- The probability of generating a given long string can easily be *less* than this (but still > 0)

Handling very low probabilities

0.01 * 0.01 * 0.01 * … (200 times) … * 0.01 = 0

- How can we fix this?
- We'll compute the *log* of the probability instead

log(0.01 * 0.01 * 0.01 * … (200 times) … * 0.01)

= log(0.01) + log(0.01) + … (200 times) … + log(0.01)

= -2 - 2 - … (200 times) - 2

= -400

Handling very low probabilities

log(0.01 * 0.01 * 0.01 * … (x200) … * 0.01)

= log(0.01) + log(0.01) + … (x200) … + log(0.01)

= -2 - 2 - … (x200) - 2

= -400

- I.e., we're compute the *exponent* of the probability (roughly speaking)
- If log(P) > log(Q), then P > Q

Testing authorship

- In A6, you'll train Markov chains for several authors
- Given several new test sequences, you'll guess who wrote which sequence
  - By finding the chain with the highest log-likelihood

- You're free to extend this in any way you can think of (treat periods and other punctuation differently, higher-order Markov models, etc)
- The best performing code (on our tests) will get two points of extra credit