Foundations of Programming

Dictionaries, Markov Models and Files

Announcements

- Enrichment video 3 is up. "Due" this morning, but you can still do it.
- Diversity/inclusion lunch next Monday. Watch for a Google form if you are interested in attending

Learning outcomes/key ideas

- Markov chains
- Representing data in a dictionary
- (Simple) reading data from a file

Algorithmic Authorship... ?

```
I like to run and to go on vacation.  Often I
 run to see things while I am on vacation.  I
will take a vacation this weekend to Seattle.
```

suppose this text represents my "style" ...

How could a *program* author new prose in this same style?!

Algorithmic Authorship... !

```
I like to run and to go on vacation.  Often I
 run to see things while I am on vacation.  I
will take a vacation this weekend to Seattle.
```

suppose this text represents my "style" ...

What would be a reasonable **first word** to start a newly-generated sentence?

What would be a reasonable **next word** to follow the first?

What would be a reasonable **test for sentence-ending**?

Algorithmic authoring *examples*...

'Cause somethin' like he left knee and a harp," said he had to the whole school? The shouting and then some strange and Mrs. "Well, I know Hagrid; they spotted handkerchief and get him get rid of course, had a gigantic beet with her," he knew what to all he's

The club isn't the flames burn on my baby, come on I'm singing like a little while longer When I'm home to say go for hours and smoke Keep watching over Durin's son If I just me in love with the mountain side Desolation comes upon the bassline Are in

Who's the original human author of each of these?

Hint: they're all British...

This is but ourselves. No, faith, My uncle! O royal bed of confession Of your rue for leave to nature; to this time I should weep for thy life is rotten before he is. have sworn 't. Or my blood. I have closely sent for nine; and unprofitable,

The Senators and the date of a written declaration that Purpose, they shall consist of nine States, shall not, when he shall have such Vacancies. The President pro tempore, in the Desire of a Qualification to the Speaker of the Senate. Article 6. When vacancies by the office upon probable

Markov Chains (Models)

Techniques for modeling *any* sequence of natural data

speech, text, sensor data...

1st-order Markov Model
(defining property)

Each item depends ***only*** on the <u>one</u> immediately before it .

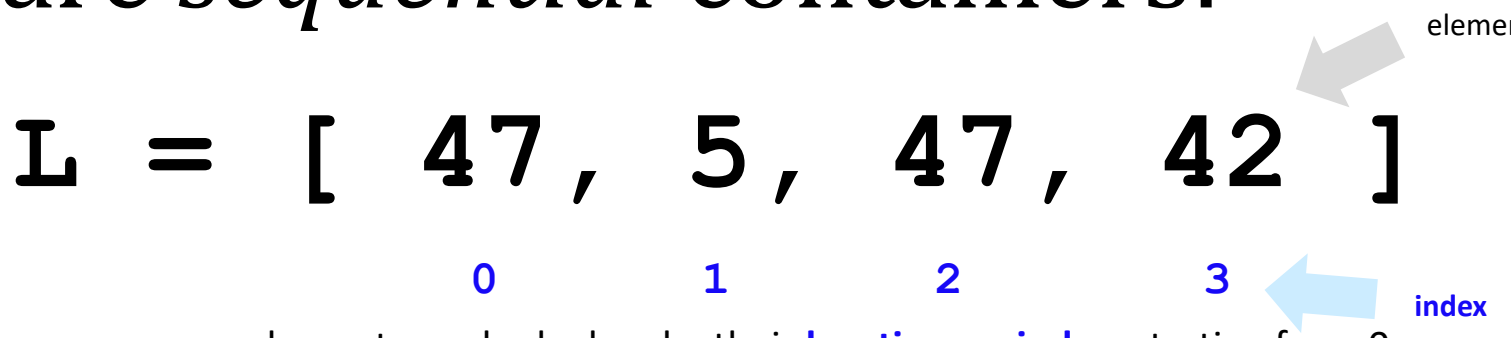
```
I like to run and to go on vacation Often I
run to see things while I am on vacation I
will take a vacation this weekend to Seattle
```

Building a Markov model to generate text:

1. Define the vocabulary (what words have we seen?)
2. Represent a state at a particular time *t*
3. Determine the transition probabilities

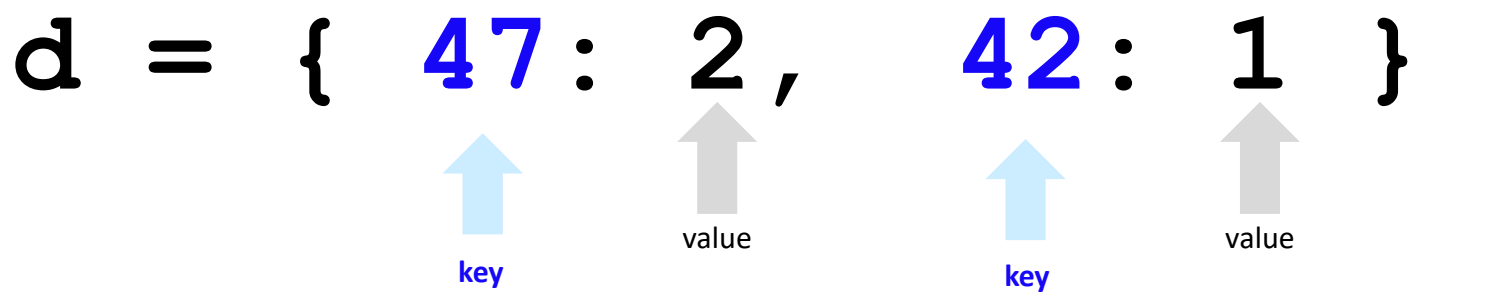
But how to do this in Python…?

Lists are *sequential* containers:



elements are looked up by their **location**, or **index**, starting from 0

Dictionaries are *arbitrary* containers:



elements (or *values*) are looked up by a **key** starting anywhere you want! **Keys** don't have to be ints!

Dictionaries are *arbitrary* containers:

```
yr = {'rabbit':1999, 'ox':1997}
```

elements (or *values*) are looked up by a **key** starting anywhere you want! **Keys** don't have to be ints!

What's up with `yr`'s data here?

Dictionaries are *arbitrary* containers:

```
yr = {'rabbit': 1999, 'ox':1997}
```

key — value — key — value

elements (or *values*) are looked up by a **key** starting anywhere you want! **Keys** don't have to be ints!

Rat	Feb 19 **1996**–Feb 06 1997
Ox	Feb 07 **1997**–Jan 27 1998
Tiger	Jan 28 **1998**–Feb 15 1999
Rabbit	Feb 16 **1999**–Feb 04 2000
Dragon	Feb 05 **2000**–Jan 23 2001
Snake	Jan 24 **2001**–Feb 11 2002
Horse	Feb 12 **2002**–Jan 31 2003
Goat	Feb 01 **2003**–Jan 21 2004
Monkey	Jan 22 **2004**–Feb 08 2005
Rooster	Feb 09 **2005**–Jan 28 2006

12-year zodiac...

Dictionaries are *arbitrary* containers:

```
yr = {'rabbit':[1999,1987,1975,...],
      'ox':[1997,1985,1973,...],
      'tiger':[1998,2010,...], ... }
```

What type are the *keys*?

A. String
B. List
C. int
D. Dictionary

Dictionaries are *arbitrary* containers:

```
yr = {'rabbit':[1999,1987,1975,...],
      'ox':[1997,1985,1973,...],
      'tiger':[1998,2010,...], ... }
```

What type are the *keys*?

What type are the *values*?

A. String
B. List
C. int
D. Dictionary

zi

Dictionaries are *arbitrary* containers:

```
z = {'rabbit':[1999,1987,1975,...],
     'ox':[1997,1985,1973,...],
     'tiger':[1998,2010,...], ... }
```

Is 'dragon' a key in **z**?

```
if 'dragon' in z
```

Is 1969 in **z['dragon']**?

```
if  1969 in z['dragon']
```

```
LoW = [ 'spam', 'spam', 'poptarts', 'spam' ]
```

Write the code that will build a dictionary where the keys are the words in the list and the values are their counts:

```
counts = {}   # empty dictionary
for word in LoW:
    if ____________________________:



    else:
```

Useful info!

```
# Add the key 'hello'
# and set its value to 0
counts['hello'] = 0

# Get the value associated
# with 'hello'
counts['hello']

# check if 'hello' is a key
'hello' in counts
```

```
LoW = [ 'spam', 'spam', 'poptarts', 'spam' ]
```

Oldenborg's menu!

`counts` will be...

```
counts = {}

for word in LoW:

  if word not in counts:

    counts[word] = 1

  else:

    counts[word] += 1
```

```
{}

{'spam':1}

{'spam':2}

{'spam':2, 'poptarts':1}

{'spam':3, 'poptarts':1}
```

final `d`

```
LoW = [ 'spam', 'spam', 'poptarts', 'spam' ]
```

but where to get so many words?

FILES !

Oldenborg's menu!

```
coun
                                    {}
                                    {'spam':1}
                                    {'spam':2}
                                    {'spam':2, 'poptarts':1}
                                    {'spam':3, 'poptarts':1}
    in counts:
        counts[word] = 1
    else:
        counts[word] += 1
```

final **d**

Files...

a.txt

```
I like to run and to go on vacation.
Often I run to see things while I am on
vacation.  I will take a vacation this
weekend to Seattle.
```

```
f = open( 'a.txt' )
```

opens the file and calls it **f**

```
text = f.read()
```

reads the whole file into the string **text**

```
f.close()
```

closes the file (optional)

```
text
'I like to run and to go on vacation. …
LoW = text.split()
[ 'I', 'like', to', ... ]
```

text.split() returns a list of each "word"

```
def word_count( filename ):

    f = open( filename )
    text = f.read()
    f.close()

    LoW = text.split()
    print("There are",len(LoW),"words")
```

file handling

What if we wanted the number of ***different*** words in the file?

This would be the author's ***vocabulary count***, instead of the total word count.

```
from filename import defaultdict

def vocab_count( filename ):

    f = open( filename )
    text = f.read()
    f.close()

    LoW = text.split()
    print "There are", len(LoW), "words."

    d = defaultdict( int )

    for w in LoW:
      if w not in d:
        d[w] = 1
      else:
        d[w] += 1

    print "There are", len(d), "_distinct_ words.\n"

    return d        # return d for later use by other code...
```

file handling

word counting

Tracking the number of occurrences of each word with a dictionary, d.

most/least common?

Vocabulary, anyone?

Shakespeare used ***31,534 different words*** -- and a grand total of 884,647 words, counting repetitions across all of his works....

http://www-math.cudenver.edu/~wbriggs/qr/shakespeare.html

Shakespearean coinages

gust
besmirch
unreal
superscript
watchdog
swagger

successful

affined
rooky
attasked
out-villained

unsuccessful

There's also a ***contemporary*** British author in the Oxford English Dictionary…

Who?

What word?

http://www.pathguy.com/shakeswo.htm
http://www.shakespeare-online.com/biography/wordsinvented.html

Vocabulary, anyone?

Shakespeare used ***31,534 different words*** -- and a grand total of 884,647 words, counting repetitions across all of his works....

http://www-math.cudenver.edu/~wbriggs/qr/shakespeare.html

Shakespearean coinages

gust
besmirch
unreal
superscript
watchdog
swagger

successful

affined
rooky
attasked
out-villained

unsuccessful

http://www.pathguy.com/shakeswo.htm
http://www.shakespeare-online.com/biography/wordsinvented.html

muggle

'Muggle' goes into Oxford English Dictionary

JK Rowling's word for non-wizards - "muggle" - has made it into the new edition of the Oxford English Dictionary (OED).

The draft definition according to the dictionary's website says:

- ***Muggle:*** *invented by JK (Joanne Kathleen) Rowling (b. 1965), British author of children's fantasy fiction (see quot. 1997).*

  In the fiction of JK Rowling: a person who possesses no magical powers. Hence in allusive and extended uses: a person who lacks a particular skill or skills, or who is regarded as inferior in some way.

J. K. Rowling

Our Markov Model

Try it!

```
I like to run and to go on vacation.
Often I run to see things while I am on
vacation.  I will take a vacation this
weekend to Seattle.
```

Original file

keys values

Markov Model

A dictionary!

{

What are the keys?

What are the values?

}

Markov Models are *generative!*

A key benefit of Markov Models is that they can ***generate*** feasible data!

Original file:

```
I like to run and to go on vacation.
Often I run to see things while I am on
vacation.  I will take a vacation this
weekend to Seattle.
```

demo…

Markov Models are *generative!*

A key benefit of Markov Models is that they can ***generate*** feasible data!

Original file:

```
I like to run and to go on vacation.
Often I run to see things while I am on
vacation.  I will take a vacation this
weekend to Seattle.
```

Generated text:

'I like to Seattle. I like to go on vacation. Often I am on vacation. Often I like to go on vacation. Often I am on vacation. I am on vacation. I run and to Seattle. I will take a'

demo…

WMSCI 2005

Rooter: A Methodology for the Typical Unification of Access Points and Redundancy

Jeremy Stribling, Daniel Aguayo and Maxwell Krohn

http://pdos.csail.mit.edu/scigen/

Markov-generated submission accepted to WMSCI 2005

*Not a first-order model … but a **third-order** model*

Rooter: A Methodology for the Typical Unification of Access Points and Redundancy

Jeremy Stribling, Daniel Aguayo and Maxwell Krohn

Abstract

Many physicists would agree that, had it not been for congestion control, the evaluation of web browsers might never have occurred. In fact, few hackers worldwide would disagree with the essential unification of voice-over-IP and public-private key pair. In order to solve this riddle, we confirm that SMPs can be made stochastic, cacheable, and interposable.

I. Introduction

Many scholars would agree that, had it not been for active networks, the simulation of Lamport clocks might never have occurred. The notion that end-users synchronize with the investigation of Markov models is rarely outdated. A theoretical grand challenge in theory is the important unification of virtual machines and real-time theory. To what extent can web browsers be constructed to achieve this purpose?

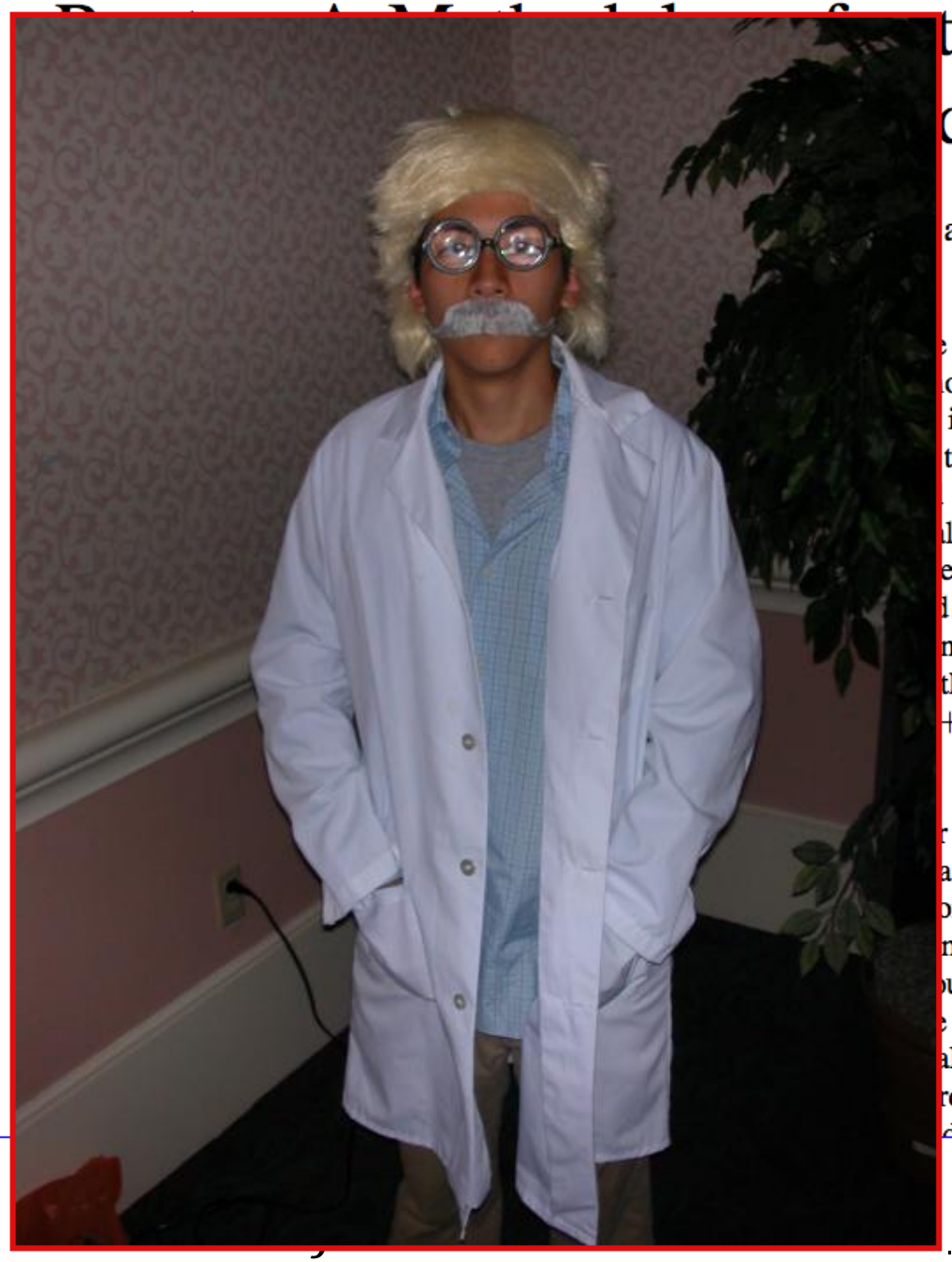
Certainly, the usual methods for the emulation of Smalltalk that paved the way for the investigation of rasterization do not apply in this area. In the opinions of many, despite the fact that conventional wisdom states that this grand challenge is continuously answered by the study of access points, we

The rest of this paper is organized as follows. For starters, we motivate the need for fiber-optic cables. We place our work in context with the prior work in this area. To address this obstacle, we disprove that even though the much-tauted autonomous algorithm for the construction of digital-to-analog converters by Jones [10] is NP-complete, object-oriented languages can be made signed, decentralized, and signed. Along these same lines, to accomplish this mission, we concentrate our efforts on showing that the famous ubiquitous algorithm for the exploration of robots by Sato et al. runs in $\Omega((n + \log n))$ time [22]. In the end, we conclude.

II. Architecture

Our research is principled. Consider the early methodology by Martin and Smith; our model is similar, but will actually overcome this grand challenge. Despite the fact that such a claim at first glance seems unexpected, it is buffetted by previous work in the field. Any significant development of secure theory will clearly require that the acclaimed real-time algorithm for the refinement of write-ahead logging by Edward Feigenbaum et al. [15] is impossible; our application is no different. This may or may not actually hold in reality

*Not a first-order model … but a **third-order** model*



the Typical Unification
d Redundancy

and Maxwell Krohn

rest of this paper is organized as follows. For starters, otivate the need for fiber-optic cables. We place our in context with the prior work in this area. To ad- this obstacle, we disprove that even though the much- autonomous algorithm for the construction of digital- log converters by Jones [10] is NP-complete, object- ed languages can be made signed, decentralized, and d. Along these same lines, to accomplish this mission, we ntrate our efforts on showing that the famous ubiquitous thm for the exploration of robots by Sato et al. runs in $+ \log n)$) time [22]. In the end, we conclude.

II. Architecture

research is principled. Consider the early methodology artin and Smith; our model is similar, but will actually ome this grand challenge. Despite the fact that such m at first glance seems unexpected, it is buffetted by us work in the field. Any significant development of theory will clearly require that the acclaimed real- algorithm for the refinement of write-ahead logging by rd Feigenbaum et al. [15] is impossible; our application different. This may or may not actually hold in reality

third-order wardrobe?

Name that author… ?

'Cause somethin' like he left knee and a harp," said he had to the whole school? The shouting and then some strange and Mrs. "Well, I know Hagrid; they spotted handkerchief and get him get rid of course, had a gigantic beet with her," he knew what to all he's

All the sky with the sun in the sun in the church where you're gone Lucy in my eyes. There beneath the girl with an hourglass And then the banker never wears a lot to hold your hand. Can't buy me tight, tight Owww! Love is love I can't hide,

Who is the author? What is the work? What is going on?

HAMLET sh.

This is but ourselves. No, faith, My uncle! O royal bed of confession Of your rue for leave to nature; to this time I should weep for thy life is rotten before he is. have sworn 't. Or my blood. I have closely sent for nine; and unprofitable,

CONSTITUTION

The Senators and the date of a written declaration that Purpose, they shall consist of nine States, shall not, when he shall have such Vacancies. The President pro tempore, in the Desire of a Qualification to the Speaker of the Senate. Article 6. When vacancies by the office upon probable