Computational thinking

Devika Subramanian
Rice University
Outline

- Computational thinking – what is it?
- Learning about computational thinking
  - comp140: a freshman course on computational thinking at Rice
- Practicing computational thinking
  - The Houston Storm Risk Calculator
Examples of problems in computational thinking

- Finding a book on your bookshelf
  - Google’s index of Web documents
- Deciding on which line to stand in at a bank, supermarket
  - Queuing theory/task scheduling for Google’s servers
- Packing items in your backpack
  - Knapsack problem for Fedex trucks
- Scheduling day/week activities (or cooking dinner)
  - Traveling salesman problem for Fedex/UPS
Challenges of scaling up

- The number of potential solutions is so large that even a fast computer cannot evaluate all these solutions.
  - How many ways to make a tour of 10 cities? 100 cities?
    - 3 million options for 10 cities, $10^{365}$ options for 100 cities
  - Some solutions are cheap, and others may be expensive or undesirable.
  - How can we find these good solutions?
    - Computational thinking tells us how!
Outline

- Computational thinking – what is it?
- Learning about computational thinking
  - comp140: a freshman course on computational thinking at Rice
- Practicing computational thinking
  - The Houston Storm Risk Calculator
The devil made a proposition to Daniel Webster. The devil proposed paying Daniel for services in the following way: "On the first day, I will pay you $1,000 early in the morning. At the end of the day, you must pay me a commission of $100. At the end of the day, we will both determine your next day's salary and my commission. I will double what you have earned at the end of the day, but you must double the amount that you pay me. Will you work for me for a month?"
Module I

- **Problem**: solving the curious case of Mark V. Shaney
- **Mathematics**: probability theory, generative models, Markov models
- **Computation**: text representation, strings, dictionaries, conditionals and iteration
- **Python**: string, dictionaries, file I/O, modules (random), if, for, while, functions
- **Project**: build a text riffer, analyze Twitter feeds from Houston
- **Riff on Bush’s 2004 speech**


Mr. Chairman, delegates, fellow citizens, I'm honored to aid the rise of democracy in Germany and Japan, Nicaragua and Central Europe and the freedom of knowing you can take them. Tonight, I remind every parent and every school must teach, so we do to improve health care and a more hopeful America. I am in their days of worry. We see that character in our future. We will build a safer world today. The progress we and our friends and allies seek in the life of our work. The terrorists are fighting freedom with all their cunning and cruelty because freedom is not America's gift to every man and woman in this place, that dream is renewed. Now we go forward, grateful for our older workers. With the huge baby boom generation approaching retirement, many of our work. About 40 nations stand beside us in the next four years.
Module 2

- **Problem**: modeling complex systems as networks and analyzing their properties

- **Mathematics**: graph theory, hubs and authorities, paths, cycles, connected components, six degrees of separation

- **Computation**: PageRank computation, graph visualizations, path computation

- **Python**: the networkx module, pylab, recursive functions (findpath), fixpoint computations (page rank)

- **Project**: analyze Rice facebook network, Enron email network, airline route maps
Rice is a tighter world than the world outside the hedges.

## Characterization of Rice colleges

<table>
<thead>
<tr>
<th>Entity</th>
<th>#comp</th>
<th>Size of comp</th>
<th>Global clust</th>
<th>Avg degree</th>
<th>dia</th>
<th>Pop deg</th>
<th>Pop bc</th>
</tr>
</thead>
<tbody>
<tr>
<td>comp140</td>
<td>13</td>
<td>77</td>
<td>0.427</td>
<td>4.52</td>
<td>13</td>
<td>Mengqi Jia</td>
<td>Connie Feng</td>
</tr>
<tr>
<td>Baker</td>
<td>3</td>
<td>140</td>
<td>0.544</td>
<td>35.67</td>
<td>4</td>
<td>Christine Cooper</td>
<td>Avinash Ravishankar</td>
</tr>
<tr>
<td>Brown</td>
<td>2</td>
<td>169</td>
<td>0.527</td>
<td>47.30</td>
<td>4</td>
<td>John Mendell</td>
<td>John Mendell</td>
</tr>
<tr>
<td>Duncan</td>
<td>3</td>
<td>181</td>
<td>0.491</td>
<td>40.75</td>
<td>5</td>
<td>Kelvin Pho</td>
<td>Kelvin Pho</td>
</tr>
<tr>
<td>Hanszen</td>
<td>2</td>
<td>169</td>
<td>0.545</td>
<td>56.45</td>
<td>4</td>
<td>Megan Johnson</td>
<td>Megan Johnson</td>
</tr>
<tr>
<td>Lovett</td>
<td>2</td>
<td>188</td>
<td>0.558</td>
<td>54.22</td>
<td>4</td>
<td>Max Chester</td>
<td>Michael Tilghman</td>
</tr>
<tr>
<td>Jones</td>
<td>3</td>
<td>156</td>
<td>0.535</td>
<td>47.58</td>
<td>4</td>
<td>Steven Boswell</td>
<td>Steven Boswell</td>
</tr>
<tr>
<td>Martel</td>
<td>1</td>
<td>175</td>
<td>0.531</td>
<td>46.14</td>
<td>4</td>
<td>Claire Morice</td>
<td>Kevin Tran</td>
</tr>
<tr>
<td>McMurtrty</td>
<td>3</td>
<td>152</td>
<td>0.500</td>
<td>33.61</td>
<td>4</td>
<td>Emily Pyle</td>
<td>Emily Pyle</td>
</tr>
<tr>
<td>SidRich</td>
<td>3</td>
<td>131</td>
<td>0.563</td>
<td>41.76</td>
<td>4</td>
<td>Gustavo Herrera</td>
<td>Gustavo Herrera</td>
</tr>
<tr>
<td>Wiess</td>
<td>3</td>
<td>178</td>
<td>0.510</td>
<td>44.42</td>
<td>4</td>
<td>Alice Fa</td>
<td>Tiffany Ho</td>
</tr>
<tr>
<td>WillRice</td>
<td>5</td>
<td>145</td>
<td>0.540</td>
<td>44.70</td>
<td>3</td>
<td>Nicholas Feltman</td>
<td>Justin Lin</td>
</tr>
</tbody>
</table>
Module 3

- **Problem**: understanding how spam filters, handwritten digit recognizers work
- **Mathematics**: recognition by inverting generative models, Bayes’ Theorem
- **Computation**: recipes with randomization, image representations, multi-level dictionaries, objects (counters), functions, methods
- **Python**: lists, dictionaries, more file I/O, math and random modules
- **Project**: build recognizer for handwritten digits from NIST database, extension to face recognizers

Which digit?
Module 4

- **Problem**: build-a-DJ
- **Mathematics**: digitization of sound by sampling, Nyquist theorem, Fourier analysis, physics of sound
- **Computation**: representations of digitally sampled sound as lists, manipulating digital sounds (fading, mirroring, blending) by list computations, algorithmic composition
- **Python**: reading and writing sound files, map/reduce, elements of functional programming, pyaudio module
- **Project**: build your own ringtone, compose a piece of classical/choral music

Algorithmic composition by Manan Mehta, F08
DJ sounds by Adrian Pellerin F09
Module 5

- **Problem**: End-to-end cryptographically verifiable elections
- **Mathematics**: homomorphic public key cryptography, modular arithmetic, zero-knowledge proofs
- **Computation**: Diffie-Helman, Elgamal crypto algorithms, Chaum-Pederson proofs
- **Python**: modular arithmetic, dictionaries, hash functions
- **Project**: Verifying the Helios election

Module developed with Professor Dan Wallach
Module 6

- **Problem**: understanding how AI bots for interactive games are designed
- **Mathematics**: graphs, path finding in graphs, random walks, decision-making
- **Computation**: search algorithms on graphs, minimax, intelligent agents
- **Python**: module (graphs, search), queues, recursive functions, OOP
- **Project**: build a pacman playing AI bot
Module 6.5

- **Problem**: evolving art
- **Mathematics**: 
- **Computation**: the genetic algorithm, tree representations of mathematical functions
- **Python**: working with tree data structures
- **Project**: building your own computational art generator

Images generated by Ian Arnold, comp140 Fall2010
Module 7

- **Problem**: understand how GoogleMaps calculates routes
- **Mathematics**: path finding algorithms in graphs (Dijkstra, Bellman-Ford, A*)
- **Computation**: graph representations, implementing A*, big O analysis of complexity, scaling
- **Python**: modules (networkx), Javascript, web services, mashups
- **Project**: building a web service to find routes between favorite haunts in Houston

http://www.google.com/help/maps/tour

Module built by my student Mitchell Koch, Rice CS alum, May 2011
You could have fun and learn about problem solving, computation and Python!

Come play and discover with us!
Outline

- Computational thinking – what is it?
- Learning about computational thinking
  - comp140: a freshman course on computational thinking at Rice
- Practicing computational thinking
  - The Houston Storm Risk Calculator
Building and validating geographically refined hurricane wind risk models for residential structures

- Devika Subramanian, Rice University
- Josue Salazar, Rice University
- Leonardo Duenas-Osorio, Rice University
- Bob Stein, Rice University

Building and validating geographically refined hurricane wind risk models for residential structures, Natural Hazards Review 10.1061/(ASCE)NH.1527-6996.0000130 (Sep. 12, 2013).
Motivation

- No information about hurricane induced risk is made available to people in the white zone.
- The white zone is where the bulk of the Harris County population resides.

Real evacuee = at risk and decided to evacuate

Shadow evacuee = not at risk and decided to evacuate

Stayed correctly = not at risk and decided to shelter

Stayed incorrectly = at risk and decided to shelter
Perspective

- How do we exploit the massive amounts of data we gather, using new instrumentation and algorithms we have developed, to help answer questions that matter to us?
- Artificial intelligence and machine learning are the primary technological enablers for data-driven decision-making!
What if we could calculate risk of damage to residential structures due to surge, rainfall and wind at the level of city blocks? Effectively communicate that risk through a website?
State of the art prediction at the census tract level for Ike
Observed Ike wind damage

Field damage assessments to roofs, exteriors, foundation, windows, landscaping, overall damage to 700,000+ residences in Harris County.

Source: Harris County Housing Authority, 2008
Side by side comparison

Predicted

Actual

(c) Devika Subramanian 4/10/14
Prediction errors at the tract level

Correct = 59.24%
Underpredicted = 31.28%
Over-predicted = 9.48%

Prediction Type
- Under
- Correct
- Over

(c) Devika Subramanian 4/10/14
Prediction errors at 1 km² level

Correct = 51.03%
Under = 25.63%
Over = 23.34%

Prediction Type
- Under
- Correct
- Over
Error analysis using machine learning

What are the characteristics of the tracts/blocks where the fragility curve based model makes errors?

explanatory variables

Learning algorithm

function to predict outcome

Training data

(c) Devika Subramanian 4/10/14
Structure of model

- **Features averaged at census tract/block level**
  - **Terrain**: Roughness length, HGAC landcover, tree100m, highway 1 km
  - **Building properties**: land value, building value, extra features value, exterior wall, floors, age, remodeled age, construction quality, garage type
  - **Construction code enforcement**: in an incorporated area, incorporation year, built after incorporation
  - **Hazard characteristics**: Max wind, wind swath, wind direction, wind duration, wind steadiness, hazus wind gust, hazus sustained wind
Explanatory variables

HCAD Land Value

HCAD Building Value

HCAD Number of Stories

HCAD Age

HCAD Remodeled Age

HCAD Quality
Results of analysis

Tract Level

Feature Type

Building
Wind
Terrain

Block Level
Interpreting results

- Variable importance plots tell us which factors are critical in designing the next generation of risk prediction models.
- The learning algorithms reveal how to modify decision criteria to predict wind risk better.

<table>
<thead>
<tr>
<th>Old parameters</th>
<th>New parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frame</td>
<td>Building value</td>
</tr>
<tr>
<td>Floors</td>
<td>Land value</td>
</tr>
<tr>
<td>Roof shape</td>
<td>Quality of construction</td>
</tr>
<tr>
<td>Roof-deck attachment</td>
<td>Age</td>
</tr>
<tr>
<td>Nailing pattern</td>
<td>Remodeled Age</td>
</tr>
<tr>
<td>Garage type</td>
<td>Incorporation Year</td>
</tr>
<tr>
<td>Terrain</td>
<td>Exterior wall</td>
</tr>
<tr>
<td>Wind speed</td>
<td></td>
</tr>
</tbody>
</table>
Direct prediction from data

Accuracy = 89%

Machine learning model

Prediction errors

http://houstonstormrisk.org
Houston Storm Risk Calculator

- http://houstonstormrisk.org/
Summary

- Computational thinking – what is it?
- Learning about computational thinking
  - comp140: a freshman course on computational thinking at Rice
- Practicing computational thinking
  - The Houston Storm Risk Calculator