Providing Students with Computational Tools for Working with Data

Center for Excellence and Innovation in Teaching, Boston University
January 10, 2013
David G. Sullivan, Ph.D.
CAS Computer Science

Databases Are Everywhere

- Example collections of data:
  - account data: banks, credit-card companies, etc.
  - airline data: flights, reservations, etc.
  - biological data: DNA sequences, protein sequences, etc.
  - socioeconomic data
  - other examples?

- Some are managed by a database management system (DBMS) like Oracle, SQLServer, etc.

- Some are not.
  - text files (CSV files, tab-delimited, etc.)
  - etc.
The Amount of Data Is Exploding!

- Example: the GenBank database of genetic sequences

![Graph showing the amount of data from August 1997 to August 2005.](ftp://ftp.ncbi.nih.gov/pub/FieldGuide/Slides/Current/MTHolyoke.05.10.06/)

The Amount of Data Is Exploding!

- Example: the UN Database (data.un.org)

![Map showing life expectancy and proportion of population with access to improved sanitation.]("An Analysis of Factors Relating to Energy and Environment in Predicting Life Expectancy", CS 105 Final Project by Valerie Belding '12)
The Amount of Data Is Exploding!

- Example: the Google Ngrams Corpus

books.google.com/ngrams

The Amount of Data Is Exploding!

xkcd.com/1140
Data Mining Is Increasingly Pervasive

- Other examples:
  - detecting fraudulent credit-card purchases
  - targeted online advertising
  - retailers mining customer-purchase data
The Problem

- Courses on databases and data mining are typically limited to CS majors and grad students.

Students from other fields are left out.

Our Solution

- CAS CS 105: Intro. to Databases and Data Mining
- Designed for non-majors
- No prereqs
- Topics include:
  - relational databases (4 weeks)
  - programming in Python (4 weeks)
    - to process data stored in text files
  - data graphics/visualization (1 week)
  - data mining basics (4 weeks)
- Provides a data-centric introduction to computer science
Broad Goals of the Course

• Give students computational tools for working with data
  • applicable skills = motivation

• Provide insight into the underlying concepts
  • abstraction
  • mathematical models
  • algorithmic thinking

• Expose them to the discipline of computer science

*Computer science is not so much the science of computers as it is the science of solving problems using computers.*
- Eric Roberts, Stanford

A Delicate Balance

• Allow students from non-technical backgrounds to succeed

• Provide sufficient challenge and coverage
Unit 1, part I: Database Fundamentals

- How data is stored
- Key functions of a DBMS: just the big picture

**Example: transactions**
- balance transfer: $50 from blue to pink
- transaction = series of steps treated as a single operation
  - *ensures all steps happen, or none do*

```
begin transaction
remove $50 from blue
add $50 to pink
end transaction
```

```
$500
$250
```

---

```
begin transaction
remove $50 from blue
add $50 to pink
end transaction
```

```
$500
$250
```

*** CRASH ***

```
restore state!
```

```
$500
$250
```
Unit 1, part II: Data Modeling

- The relational model
  - data is organized into tables
  - example: a table of student info

<table>
<thead>
<tr>
<th>id</th>
<th>name</th>
<th>address</th>
<th>class</th>
<th>dob</th>
</tr>
</thead>
<tbody>
<tr>
<td>12345678</td>
<td>Jill Jones</td>
<td>Warren Towers 100</td>
<td>2013</td>
<td>3/10/95</td>
</tr>
<tr>
<td>25252525</td>
<td>Alan Turing</td>
<td>Student Village A210</td>
<td>2015</td>
<td>2/7/97</td>
</tr>
<tr>
<td>33566891</td>
<td>Audrey Chu</td>
<td>300 Main Hall</td>
<td>2014</td>
<td>10/2/96</td>
</tr>
<tr>
<td>45678900</td>
<td>Jose Delgado</td>
<td>Student Village B300</td>
<td>2016</td>
<td>7/13/98</td>
</tr>
<tr>
<td>66666666</td>
<td>Count Dracula</td>
<td>The Dungeon</td>
<td>2007</td>
<td>11/14/31</td>
</tr>
</tbody>
</table>

- Other data-modeling topics:
  - keys, types, schema, etc.

Example Database

- Data obtained from imdb.com

- Tables with info about:
  - people
  - movies
  - Academy Awards
Unit 1, part III: SQL

• SQL is the *query language* used in relational databases.

• Include fairly advanced topics:
  • joins of two or more tables
  • simple subqueries
  • aggregates, GROUP BY, HAVING
  • outer joins

Unit 1, part III: SQL (cont.)

• We answer (or at least explore) questions like:
  • How many of the top-grossing films have won Oscars?
  • Does the Academy discriminate against older women?
Making It Accessible

• Take a gradual approach:
  • start with queries on a single table
    • example: _Get the names of all movies rated PG-13._
  • then introduce queries that _join_ two or more tables
    • example: _Get the names of all people who won either Best Actor or Best Actress in the 1990s._

• Provide hints as needed in the assignments.

• Be judicious in coverage of mathematical underpinnings.

Beyond Relational Databases

• Example: DNA sequence data
  >gi|49175990|ref|NC_000913.2| Escherichia coli 912, complete genome
  AGCTTTTCTTATTCTGACTGCAACGGGCAATATGTCTCTGTGTGGATTAAAAAAAGAGTGTCTGATAGCAGCTTCTGAACTGGTTACCTGCCGTGAGTA
  AATTAAAATTTTATTGACTTAGGTCACTAAATACTTTAACCAATATAGGCATAGCGCACAGACAGATAAAAATTACAGAGTACACAACATCCATGAA
  ACGCATTAGCACCACCATTACCACCACCATCACCATTACCACAGGTAACGGTGCGGGCTGACGCGTACAGGAAACACAGAAAAAAGCCGACACCTGA
  CAGTGCGGGCTTTTTTTTTCGACCAAAGGTAACGAGGTAACAACCATGCGAGTGTTGAAGTTCGGCGGTACATCAGTGGCAAATGCAGAACGTTTTC
  TGCGTGTTGCCGATATTCTGGAAAGCAATGCCAGGCAGGGGCAGGTGGCCACCGTCCTCTCTGCCCCCGCCAAAATCACCAACCACCTGGTGGCAGAT
  GATTGAAAAAACCATTAGCGGCCAGGATGCTTTACCCAATATCAGCGATGCCGAACGTATTTTTGCCGAACTTTTGACGGGACTCGCCGCCGCCCAG
  CCGGGGGTTCGCGCTCGAGCA

• Common queries involve looking for similarities or patterns.
  • what genes in mice are similar to genes in humans?
  • SQL can't do this!
Unit 2: Programming in Python

• Main goal: to be able to process data stored in text files

• Python makes it easier.

Making It Accessible

• Take advantage of Python:
  • simplified syntax
  • list and file processing

• Build on concepts learned in SQL

• Structure the assignments carefully
  • start by modifying an existing program
  • later, write programs similar to ones from lecture
  • provide "scaffolding"
Example Problem Without Scaffolding

Body Mass Index
A person's body mass index (BMI) is equal to the person's weight in pounds, multiplied by 720, and then divided by the square of the person's height in inches. 19-25 is the range of healthy BMI values. Write a program that reads a person's weight and height, computes and prints the person's BMI to the nearest integer, and prints a message indicating whether they are below, above, or within the healthy range. You may assume that both inputs are positive.

Example Problem With Scaffolding

Body Mass Index
Body mass index (BMI) is a measure of body fat that is based on a person's weight and height. 19-25 is the range of healthy BMI values. Write a program named bmi.py that can be used to compute a person's BMI, and to determine whether it is below, above, or within the healthy range.

Step 1: The program should begin by getting the following inputs from the user:
- the person's weight, storing it in a variable named weight
- the person's height, storing it in a variable named height

Step 2: The program should then use the values of the variables weight and height to compute and print the person's BMI as a real number using the following formula:

\[
\text{BMI} = \frac{720 \times \text{weight}}{\text{height} \times \text{height}}
\]

...
Unit 3: Data Visualization

• A shorter unit taught by Wayne Snyder
• Based on the work of Edward Tufte
• Principles for creating data graphics that combine:
  • simplicity of design
  • complexity of data
• Show the value that computational tools can add

Unit 4: Data Mining

• The process of finding patterns in data.
  • "hidden knowledge"
  • vs. the "shallow", factual knowledge given by SQL queries
• Data mining applies *machine-learning* algorithms that:
  • operate on a set of *training data*
  • learn some type of *model*
Classification Learning

- One type of machine learning
- Learns a model that can classify/categorize
- Something that human beings have always done!
  - example: how do we learn to identify a dog?

Example: Medical Diagnosis

- Goal: diagnose a patient with cold-like symptoms
  - classify as: Strep throat, Allergy, or Cold

- Sample training data (Roiger & Geatz):

<table>
<thead>
<tr>
<th>Patient ID#</th>
<th>Sore Throat</th>
<th>Fever</th>
<th>Swollen Glands</th>
<th>Congestion</th>
<th>Headache</th>
<th>Diagnosis</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Strep throat</td>
</tr>
<tr>
<td>2</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Allergy</td>
</tr>
<tr>
<td>3</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Cold</td>
</tr>
<tr>
<td>4</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Strep throat</td>
</tr>
<tr>
<td>5</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Cold</td>
</tr>
<tr>
<td>6</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Allergy</td>
</tr>
<tr>
<td>7</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Strep throat</td>
</tr>
<tr>
<td>8</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Allergy</td>
</tr>
<tr>
<td>9</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Cold</td>
</tr>
<tr>
<td>10</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Cold</td>
</tr>
</tbody>
</table>

- Can you see any patterns that would help the diagnosis?
Different algorithms learn different types of models.

One possible model is a set of rules:

if Swollen Glands == Yes
then Diagnosis = Strep Throat

if Swollen Glands == No and Fever == Yes
then Diagnosis = Cold

if Swollen Glands == No and Fever == No
then Diagnosis = Allergy

Another possible model is known as a decision tree:

What diagnosis would it give for patient 11 above?
### Example: Medical Diagnosis (cont.)

<table>
<thead>
<tr>
<th>Patient ID#</th>
<th>Sore Throat</th>
<th>Fever</th>
<th>Swollen Glands</th>
<th>Congestion</th>
<th>Headache</th>
<th>Diagnosis</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Strep throat</td>
</tr>
<tr>
<td>2</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Allergy</td>
</tr>
<tr>
<td>3</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Cold</td>
</tr>
<tr>
<td>4</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Strep throat</td>
</tr>
<tr>
<td>5</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Cold</td>
</tr>
<tr>
<td>6</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Allergy</td>
</tr>
<tr>
<td>7</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Strep throat</td>
</tr>
<tr>
<td>8</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Allergy</td>
</tr>
<tr>
<td>9</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Cold</td>
</tr>
<tr>
<td>10</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Cold</td>
</tr>
<tr>
<td>11</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Allergy</td>
</tr>
</tbody>
</table>

- Another possible model is known as a **decision tree**:

![Decision Tree Diagram]

- **What diagnosis would it give for patient 11 above?**

### Unit 4: Data Mining (cont.)

- Teach two simple classification-learning algorithms
  - students apply them by hand

- Other topics include:
  - two other types of machine learning
  - preparing data for mining
  - assessing the goodness of a learned model
  - the possibility of overfitting

- Introduce students to Weka
  - freely available toolkit for data mining
Making It Accessible

• Stick with simple algorithms and straightforward math

• Apply the algorithms to "toy" datasets
  • provides concrete illustrations of the key concepts

• Throughout the course, hold weekly lab sessions
  • hands-on practice, assisted by a TF

• Use the Piazza online learning environment

Student Assessment

• Nine problem sets

• Three 50-minute "quizzes"

• Final exam

• Final project
  • choose a dataset of interest
  • analyze it using techniques learned from the course
  • written report
  • brief in-class presentation
  • work alone or in pairs

• Hall of Fame on course website
  http://cs-people.bu.edu/dgs/courses/cs105/hall_of_fame/
Enrollment fall and spring semesters combined

Most Common Majors of Enrollees

<table>
<thead>
<tr>
<th>Major</th>
<th>Number of Students</th>
</tr>
</thead>
<tbody>
<tr>
<td>Economics</td>
<td>32</td>
</tr>
<tr>
<td>Business Admin / Management</td>
<td>30</td>
</tr>
<tr>
<td>Computer Science</td>
<td>21</td>
</tr>
<tr>
<td>International Relations</td>
<td>19</td>
</tr>
<tr>
<td>Archeology</td>
<td>16</td>
</tr>
<tr>
<td>Mathematics</td>
<td>16</td>
</tr>
<tr>
<td>Anthropology</td>
<td>14</td>
</tr>
<tr>
<td>Undeclared</td>
<td>13</td>
</tr>
<tr>
<td>Political Science</td>
<td>13</td>
</tr>
<tr>
<td>English</td>
<td>11</td>
</tr>
</tbody>
</table>

fall 2007 – spring 2011
End-of-Semester Evaluations

(averages of ratings on a 1-5 scale)

Follow-Up Survey

- Online survey of alums of the course from 2007-2011
- Received 96 replies (35% response rate)
- Very positive responses overall
CS 105 Has Been Useful...

- ...in subsequent classes
  - strongly agree
  - agree
  - uncertain/neutral
  - disagree
  - strongly disagree

- ...in non-academic pursuits like internships and jobs
  - strongly agree
  - agree
  - uncertain/neutral
  - disagree
  - strongly disagree

Computer Science Is Relevant to My Pursuits

- I thought that **before** taking the course.
  - strongly agree
  - agree
  - uncertain/neutral
  - disagree
  - strongly disagree

- I think that now, **after** taking the course.
  - strongly agree
  - agree
  - uncertain/neutral
  - disagree
  - strongly disagree
CS 105 Increased My Interest
In Learning More About Computer Science

Lessons Learned

• Take steps to keep the material accessible.
  • gradually increase the level of difficulty
  • provide hints/scaffolding in the assignments

• Be willing to experiment and adapt.
  • list of topics
  • sequencing of topics
  • number and difficulty of assignments

• You can teach non-majors practical tools for working with data.
  • introduce key concepts at the same time
  • show them a way of thinking and solving problems
    that underlies much of the modern world