Scalable Data Analysis (CIS 602-02)

Data Cleaning

Dr. David Koop
Growth of Data

The Digital Universe: 50-fold Growth from the Beginning of 2010 to the End of 2020

The Untapped Big Data Gap (2012)

- Useful If Tagged and Analyzed: 23%
- Tagged: 3%
- Analyzed: 0.5%

Example Data Sources

- Radio Telescopes
- Twitter
- Wind Turbine Sensors
- Surveillance Cameras
- Cars & Airplanes
- Dog Collars
- Dishwashers
- Traffic Lights
- MRI Scanners
- NFL Football Players
- Farming
Big Data or Small Data?

[Jobs on a Large Analytics Cluster, R. Appuswamy et al., 2013]
Big Data or Small Data?

- Projects with big data concerns that require strategies and infrastructure
- Lots of small data waiting for analysis
Assignment 1

• Due today at 11:59pm (Late Policy)
• http://www.cis.umassd.edu/~dkoop/cis602/assignment1.html
• United Nations High Commissioner for Refugees maintains statistics about population of concern around the world
• Includes data about refugees and asylum seekers in different countries
• Use the provided template
• You must complete the provided functions and not rename them
• Turn in a single .py file via myCourses
Reading Presentations

• Present paper and critique of it
• Goal: help the class better understand a specific paper topic
• Two students per reading
  - Both summarize key points
  - One student focuses on positive points, associated ideas, and past work
  - One student focuses on criticisms, improvements, later work
• After presentations, everyone in the class should be involved in discussing items further
• Complete myCourses Survey about topic interest
• Volunteers for Thursday's paper?
Data Cleaning

- Remove errors, find inconsistencies
- Reshape data
- Format data
- Deal with missing data
Wrangler: Interactive Visual Specification of Data Transformation Scripts

S. Kandel, A. Paepcke, J. Hellerstein, J. Heer
Wrangler

- Data cleaning takes a lot of **time** and **human effort**
- "Tedium is the message"
- Repeating this process on multiple data sets is even worse!
- Solution:
  - interactive interface (mixed-initiative)
  - transformation language with natural language "translations"
  - suggestions + "programming by demonstration"
Previous Work: Potter's Wheel

- V. Raman and J. Hellerstein, 2001
- Defines structure extractions for identifying fields
- Defines transformations on the data
- Allows user interaction
Potter's Wheel: Structure Extraction

<table>
<thead>
<tr>
<th>Example Column Value (Example erroneous values)</th>
<th># Structures Enumerated</th>
<th>Final Structure Chosen (Punc = Punctuation)</th>
</tr>
</thead>
<tbody>
<tr>
<td>-60</td>
<td>5</td>
<td>Integer</td>
</tr>
<tr>
<td>UNITED, DELTA, AMERICAN etc.</td>
<td>5</td>
<td>IspellWord</td>
</tr>
<tr>
<td>SFO, LAX etc. (JFK to OAK)</td>
<td>12</td>
<td>AllCapsWord</td>
</tr>
<tr>
<td>1998/01/12</td>
<td>9</td>
<td>Int Punc() Int Punc() Int</td>
</tr>
<tr>
<td>M, Tu, Thu etc.</td>
<td>5</td>
<td>Capitalized Word</td>
</tr>
<tr>
<td>06:22</td>
<td>5</td>
<td>Int(len 2) Punc(:) Int(len 2)</td>
</tr>
<tr>
<td>12.8.15.147 (ferret03.webtop.com)</td>
<td>9</td>
<td>Double Punc(.) Double</td>
</tr>
<tr>
<td>”GET\b (\b) /postmodern/lecs/xia/sld013.htm</td>
<td>5</td>
<td>Punc(”) IspellWord Punc()</td>
</tr>
<tr>
<td>HTTP</td>
<td>3</td>
<td>AllCapsWord(HTTP)</td>
</tr>
<tr>
<td>/1.0</td>
<td>6</td>
<td>Punc() Double(1.0)</td>
</tr>
</tbody>
</table>

[V. Raman and J. Hellerstein, 2001]
Potter's Wheel:

<table>
<thead>
<tr>
<th>Transform</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Format</td>
<td>$\phi(R, i, f) = {(a_1, \ldots, a_{i-1}, a_{i+1}, \ldots, a_n, f(a_i)) \mid (a_1, \ldots, a_n) \in R}$</td>
</tr>
<tr>
<td>Add</td>
<td>$\alpha(R, x) = {(a_1, \ldots, a_n, x) \mid (a_1, \ldots, a_n) \in R}$</td>
</tr>
<tr>
<td>Drop</td>
<td>$\pi(R, i) = {(a_1, \ldots, a_{i-1}, a_{i+1}, \ldots, a_n) \mid (a_1, \ldots, a_n) \in R}$</td>
</tr>
<tr>
<td>Copy</td>
<td>$\kappa((a_1, \ldots, a_n), i) = {(a_1, \ldots, a_n, a_i) \mid (a_1, \ldots, a_n) \in R}$</td>
</tr>
<tr>
<td>Merge</td>
<td>$\mu((a_1, \ldots, a_n), i, j, \text{glue}) = {(a_1, \ldots, a_{i-1}, a_{i+1}, \ldots, a_{j-1}, a_{j+1}, \ldots, a_n, a_i \oplus \text{glue} \oplus a_j) \mid (a_1, \ldots, a_n) \in R}$</td>
</tr>
<tr>
<td>Split</td>
<td>$\omega((a_1, \ldots, a_n), i, \text{splitter}) = {(a_1, \ldots, a_{i-1}, a_{i+1}, \ldots, a_n, \text{left}(a_i, \text{splitter}), \text{right}(a_i, \text{splitter})) \mid (a_1, \ldots, a_n) \in R}$</td>
</tr>
<tr>
<td>Divide</td>
<td>$\delta((a_1, \ldots, a_n), i, \text{pred}) = {(a_1, \ldots, a_{i-1}, a_{i+1}, \ldots, a_n, a_i, \text{null}) \mid (a_1, \ldots, a_n) \in R \land \text{pred}(a_i)} \cup {(a_1, \ldots, a_{i-1}, a_{i+1}, \ldots, a_n, \text{null}, a_i) \mid (a_1, \ldots, a_n) \in R \land \neg\text{pred}(a_i)}$</td>
</tr>
<tr>
<td>Fold</td>
<td>$\lambda(R, i_1, i_2, \ldots i_k) = {(a_1, \ldots, a_{i_1-1}, a_{i_1+1}, \ldots, a_{i_2-1}, a_{i_2+1}, \ldots, a_{i_k-1}, a_{i_k+1}, \ldots, a_n, a_{i_1}) \mid (a_1, \ldots, a_n) \in R \land 1 \leq l \leq k}$</td>
</tr>
<tr>
<td>Select</td>
<td>$\sigma(R, \text{pred}) = {(a_1, \ldots, a_n) \mid (a_1, \ldots, a_n) \in R \land \text{pred}(a_1, \ldots, a_n)}$</td>
</tr>
</tbody>
</table>

Notation: $R$ is a relation with $n$ columns. $i, j$ are column indices and $a_i$ represents the value of a column in a row. $x$ and glue are values. $f$ is a function mapping values to values. $x \oplus y$ concatenates $x$ and $y$. splitter is a position in a string or a regular expression, left($x$, splitter) is the left part of $x$ after splitting by splitter. pred is a function returning a boolean.

[V. Raman and J. Hellerstein, 2001]
Potter's Wheel: Example

![Diagram showing Potter's Wheel example with splits and merges]

[Source: V. Raman and J. Hellerstein, 2001]
**Potter's Wheel: Inferring Structure from Examples**

<table>
<thead>
<tr>
<th>Example Values Split By User</th>
<th>Inferred Structure</th>
<th>Comments</th>
</tr>
</thead>
</table>
| Taylor, Jane | $52,072  
Blair, John | $73,238  
Tony Smith | $1,00,533 |
|                           | $(<\xi^* > <', Money >) | Parsing is doable despite no good delimiter. A regular expression domain can infer a structure of $[0-9,]*$ for last component. |
| MAA | to | SIN  
JFK | to | SFO  
LAX | - | ORD  
SEA | / | OAK |
|                           | $(<\text{len 3 identifier}> <\xi^* >  
<\text{len 3 identifier}> ) | Parsing is possible despite multiple delimiters. |
| 321 Blake #7 | , Berkeley | , CA 94720  
719 MLK Road | , Fremont | , CA 95743 |
|                           | $(<\text{number} \ \xi^* > <', word>  
<', (2 \text{ letter word}) (5 \text{ letter integer})>) | Parsing is easy because of consistent delimiter. |

[V. Raman and J. Hellerstein, 2001]
Wrangler Transformation Language

- Based on Potter's Wheel
- Map: Delete, Extract, Cut, Split, Update
- Lookup/join: Use external data (e.g. from zipcode→state)
- Reshape: Fold and Unfold (aka pivot)
- Positional: Fill and lag
- Sorting, aggregation, key generation, schema transforms
Interface

- Automated Transformation Suggestions
- Editable Natural Language Explanations
  - Fill Bangladesh by **copying** values from above
  - Fill Bangladesh by **averaging** the 5 values from above
- Visual Transformation Previews
- Transformation History

---

[S. Kandel et al., 2011]
Automation from past actions

• Infer parameter sets from user interaction

• Generating transforms

  (a) **Reported crime in Alabama**

  before: {'in', ' '} ‘Alabama’ → {'Alabama', word}
  after: {()} {('Alabama', word),()}

  (b) selection: {'Alabama'} ‘in’ → {'in', word, lowercase}
       after: {()} {('Alabama'),()} {('in'), ('Alabama'),()}

  (c) selection: {'Alabama'}, (word)
       after: {()} {('Alabama'),()}

  (d) {('Alabama'),(),()}
       {('Alabama'),(word),()}
       {('Alabama'),(),()} {('in'), ('Alabama'),(),()}
       {('in'), ('Alabama'),(word),()}

  (e) {('in'), ('Alabama'),(),()} {('in'), ('Alabama'),(word),()}

• Ranking and ordering transformations:
  - Based on user preferences, difficulty, and corpus frequency
  - Sort transforms by type and diversify suggestions
Evaluation

• Compare with Excel

• Tests:
  - Extract text from a single string entry
  - Fill in missing values with estimates
  - Reshape tables

• Allowed users to ask questions about Excel, not Wrangler

• Found significant effect of tool and users found previews and suggestions helpful

• Complaint: No manual fallback, make implications of user choices more obvious for users
Task Completion Times

![Task Completion Times Chart]

Figure 11. Task completion times. Black bars indicate median values.

[S. Kandel et al., 2011]
Data Wrangler Demo

- http://vis.stanford.edu/wrangler/app/

**Transform Script**

- Split **data repeatedly** on **newline** into rows
- Split **split repeatedly** on ','
- Promote **row 0** to header

**Table**

<table>
<thead>
<tr>
<th>Year</th>
<th>Property_crime_rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Reported crime in Alabama</td>
</tr>
<tr>
<td>1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>2004</td>
</tr>
<tr>
<td>3</td>
<td>2005</td>
</tr>
<tr>
<td>4</td>
<td>2006</td>
</tr>
<tr>
<td>5</td>
<td>2007</td>
</tr>
<tr>
<td>6</td>
<td>2008</td>
</tr>
<tr>
<td>7</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Reported crime in Alaska</td>
</tr>
<tr>
<td>9</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>2004</td>
</tr>
<tr>
<td>11</td>
<td>2005</td>
</tr>
<tr>
<td>12</td>
<td>2006</td>
</tr>
</tbody>
</table>
Pandas Demo

Reminders

• Assignment 1 due tonight (Late policy)
• Submit presentation preferences on myCourses
• Volunteers for Thursday's presentation
• Reading response for Thursday's class