Advanced Data Management (CSCI 490/680)

Visualization and Databases

Dr. David Koop
pandas 2.0 Released Last Monday

- `dtype_backend` can be `pyarrow`
- Number of changes to `datetime`
  - UTC, `datetime[s]`, parsing
  - [https://pandas.pydata.org/pandas-docs/version/2.0.0/whatsnew/v2.0.0.html#datetimes-are-now-parsed-with-a-consistent-format](https://pandas.pydata.org/pandas-docs/version/2.0.0/whatsnew/v2.0.0.html#datetimes-are-now-parsed-with-a-consistent-format)
Graphs

• In computing, a graph is an abstract data structure that represents set objects and their relationships as vertices and edges/links, and supports a number of graph-related operations

• Objects (nodes): \{A, B, C, D\}

• Relationships (edges):
  \{(D, B), (D, A), (B, C), (B, A), (C, A)\}

• Operation: shortest path from D to A

[K. Salama, 2016]
Graphs with Properties

- Each vertex or edge may have properties associated with it
- May include identifiers or classes

Person

- name = 'Tom Hanks'
- born = 1956

Person

- name = 'Robert Zemeckis'
- born = 1951

Movie

- title = 'Forrest Gump'
- released = 1994

ACTED_IN
- roles = ['Forrest']

DIRECTED
What is a Graph Database?

- A database with an explicit graph structure
- Each node knows its adjacent nodes
- As the number of nodes increases, the cost of a local step (or hop) remains the same
- Plus an Index for lookups

[M. De Marzi, 2012]
Graph Databases Compared to Relational Databases

- Optimized for aggregation
- Optimized for connections

[M. De Marzi, 2012]
Graph Databases Compared to Relational Databases

• Relational Databases (querying is through joins)
  - In effect, the join operation forms a graph that is dynamically constructed as one table is linked to another table.
  - Must be inferred through a series of index-intensive operations

• Graph Databases (querying is through traversal paths)
  - There is no explicit join operation because vertices maintain direct references to their adjacent edges
  - Structures are "hard-wired", not computed at query time

[Rodriguez & Neubauer via Lembo & Rosati]
Example: Friend of Friends Query

- **Relational:**

- **Graph:**

[Lembo & Rosati]
Graph Databases Compared to Key-Value Stores

- Optimized for simple look-ups
- Optimized for traversing connected data

[M. De Marzi, 2012]
Storing and Traversing Graphs

• Storage:
  - Adjacency List: nodes store their neighbors
  - Incidence List: nodes store edges and edges store incident nodes
  - Adjacency Matrix: adjacency list in matrix form (rows & cols are nodes)
  - Incidence Matrix: rows are vertices, columns are edges

• Traversal:
  - Breadth-first Search
  - Depth-first Search
Adjacency List vs. Incidence List

Adjacency List:

- **Properties:**
  - Storage: $O(|V|+|E|+|L|)$
  - $\text{Adjacent}(G,x,y)$: $O(|V|)$
  - $\text{Neighbors}(G,x)$: $O(|V|)$
  - $\text{AdjacentEdges}(G,x,y)$: $O(|V|+|E|)$
  - $\text{Add}(G,x,y,l)$: $O(|V|+|E|)$
  - $\text{Delete}(G,x,y,l)$: $O(|V|+|E|)$

Incidence List:

- **Properties:**
  - Storage: $O(|V|+|E|+|L|)$
  - $\text{Adjacent}(G,x,y)$: $O(|E|)$
  - $\text{Neighbors}(G,x)$: $O(|E|)$
  - $\text{AdjacentEdges}(G,x,y)$: $O(|E|)$
  - $\text{Add}(G,x,y,l)$: $O(|E|)$
  - $\text{Delete}(G,x,y,l)$: $O(|E|)$

Simplified version: each edge has a different label

Source: [Acosta et al.]
Adjacency Matrix vs. Incidence Matrix

**Adjacency Matrix**

<table>
<thead>
<tr>
<th></th>
<th>V1</th>
<th>V2</th>
<th>V3</th>
<th>V4</th>
</tr>
</thead>
<tbody>
<tr>
<td>V1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>V2</td>
<td>{L2}</td>
<td></td>
<td>{L3}</td>
<td></td>
</tr>
<tr>
<td>V3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>V4</td>
<td>{L1}</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Properties:**
- **Storage:** $O(|V|^2)$
- **Adjacent(G,x,y):** $O(1)$
- **Neighbors(G,x):** $O(|V|)$
- **AdjacentEdges(G,x,y):** $O(|E|)$
- **Add(G,x,y,l):** $O(|E|)$
- **Delete(G,x,y,l):** $O(|E|)$

From [ABFRV14]

**Incidence Matrix**

<table>
<thead>
<tr>
<th></th>
<th>L1</th>
<th>L2</th>
<th>L3</th>
</tr>
</thead>
<tbody>
<tr>
<td>V1</td>
<td>destination</td>
<td>destination</td>
<td></td>
</tr>
<tr>
<td>V2</td>
<td>source</td>
<td>source</td>
<td></td>
</tr>
<tr>
<td>V3</td>
<td></td>
<td></td>
<td>destination</td>
</tr>
<tr>
<td>V4</td>
<td>source</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Properties:**
- **Storage:** $O(|V| \times |E|)$
- **Adjacent(G,x,y):** $O(|E|)$
- **Neighbors(G,x):** $O(|V| \times |E|)$
- **AdjacentEdges(G,x,y):** $O(|E|)$
- **Add(G,x,y,l):** $O(|V|)$
- **Delete(G,x,y,l):** $O(|V|)$

From [Acosta et al.]
Property Graph Model (Cypher in neo4j)

- Directed, labelled, attributed multigraph
- Properties are **key/value pairs** that represent metadata for nodes and edges

![Property Graph Diagram]

[R. Angles and C. Gutierrez, 2017]
Graph Query Languages: Cypher

- Implemented by neo4j system
- Expresses reachability queries via path expressions
  
  \[- p = (a)-[\text{:knows}\ast]\rightarrow(b): \text{nodes from } a \text{ to } b \text{ following } \text{knows} \text{ edges}\]

- \text{START } x=\text{node:person(name="John")}
  \text{MATCH } (x)-[\text{:friend}]->(y)
  \text{RETURN } y.\text{name}\]
Graph DBMS Building Blocks

- Property graph data model
- Graph query language
- Graph visualization
- Subgraph matching
- Relational queries
- Path queries
- Stored procedures

[Graph DBMS Building Blocks diagram by P. Boncz, 2022]
Graph DBMS Problems

- **performance**
  - Slow loading speeds
  - Query speeds over magnitude slower than RDBMS
- **scalability**
  - Low datasize limit, typically << RAM
  - Little benefit from parallelism
- **reliability**
  - Loads never terminate
  - Query run out of memory or crash
  - Bugs

[D. Koop, CSCI 680/490, Spring 2022]
Quiz Monday

- Read Beast paper
- Quiz at the beginning of class
Assignment 5

- Last assignment, coming soon
- Graphs, visualization, spatial data, time series
Data Exploration through Visualization
# MTA Fare Data Exploration

<table>
<thead>
<tr>
<th>REMOTE</th>
<th>STATION</th>
<th>FF</th>
<th>SEN/DIS</th>
<th>7-D AFAS UNL</th>
<th>D AFAS/RMF</th>
<th>JOINT RR TKT</th>
<th>7-D UNL</th>
<th>30-D UNL</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>R011 42ND STREET &amp; 8TH AVENUE</td>
<td>00228985</td>
<td>00008471</td>
<td>00000441</td>
<td>00001455</td>
<td>00000134</td>
<td>0003341</td>
<td>00071255</td>
</tr>
<tr>
<td>2</td>
<td>R170 14TH STREET-UNION SQUARE</td>
<td>00224603</td>
<td>0011051</td>
<td>00000827</td>
<td>00003026</td>
<td>00000660</td>
<td>00089367</td>
<td>00199841</td>
</tr>
<tr>
<td>3</td>
<td>R046 42ND STREET &amp; GRAND CENTRAL</td>
<td>00207758</td>
<td>00007908</td>
<td>00000323</td>
<td>00001183</td>
<td>00003001</td>
<td>00040759</td>
<td>00096613</td>
</tr>
<tr>
<td>4</td>
<td>R012 34TH STREET &amp; 8TH AVENUE</td>
<td>00188311</td>
<td>00006490</td>
<td>00000498</td>
<td>00001279</td>
<td>00003622</td>
<td>00035527</td>
<td>00067483</td>
</tr>
<tr>
<td>5</td>
<td>R293 34TH STREET - PENN STATION</td>
<td>00168768</td>
<td>00006155</td>
<td>00000523</td>
<td>00001065</td>
<td>00005031</td>
<td>00030645</td>
<td>00054376</td>
</tr>
<tr>
<td>6</td>
<td>R033 42ND STREET/TIMES SQUARE</td>
<td>00159382</td>
<td>00005945</td>
<td>00000378</td>
<td>00001205</td>
<td>00000690</td>
<td>00058931</td>
<td>00078644</td>
</tr>
<tr>
<td>7</td>
<td>R022 34TH STREET &amp; 6TH AVENUE</td>
<td>00156008</td>
<td>00006276</td>
<td>00000487</td>
<td>00001543</td>
<td>00000712</td>
<td>00058910</td>
<td>00110466</td>
</tr>
<tr>
<td>8</td>
<td>R084 59TH STREET/COLUMBUS CIRCLE</td>
<td>00155262</td>
<td>00009484</td>
<td>00000589</td>
<td>00002071</td>
<td>00000542</td>
<td>00053397</td>
<td>00113966</td>
</tr>
<tr>
<td>9</td>
<td>R020 47-50 STREETS/ROCKEFELLER</td>
<td>00143500</td>
<td>00006402</td>
<td>00000384</td>
<td>00001159</td>
<td>00000723</td>
<td>00037978</td>
<td>00090745</td>
</tr>
<tr>
<td>10</td>
<td>R179 86TH STREET-LEXINGTON AVE</td>
<td>00142169</td>
<td>0010367</td>
<td>00000470</td>
<td>00001839</td>
<td>00000271</td>
<td>00050328</td>
<td>00125250</td>
</tr>
<tr>
<td>11</td>
<td>R023 34TH STREET &amp; 6TH AVENUE</td>
<td>00134052</td>
<td>00005005</td>
<td>00000348</td>
<td>00001112</td>
<td>00000649</td>
<td>00031531</td>
<td>00075040</td>
</tr>
<tr>
<td>12</td>
<td>R029 PARK PLACE</td>
<td>00121614</td>
<td>00004311</td>
<td>00000287</td>
<td>00000931</td>
<td>00000792</td>
<td>00025404</td>
<td>00065362</td>
</tr>
<tr>
<td>13</td>
<td>R047 42ND STREET &amp; GRAND CENTRAL</td>
<td>00100742</td>
<td>00004273</td>
<td>00000185</td>
<td>00000704</td>
<td>00001241</td>
<td>00022808</td>
<td>00068216</td>
</tr>
</tbody>
</table>
MTA Fare Data Exploration
MTA Fare Data Exploration
MTA Fare Data Exploration
MTA Fare Data Exploration
MTA Fare Data Exploration

![Bar chart showing fare data for East 161st Street and River Avenue]

- x-axis: Date
- y-axis: Full Fares Purchased
- Data is available from 08-02 to 11-01
MTA Fare Data Exploration

East 161st Street and River Avenue

New York Yankees

ALL GAMES ARE EASTERN TIME.

2013 REGULAR SEASON SCHEDULE
Definition of Visualization

“Computer-based visualization systems provide visual representations of datasets designed to help people carry out tasks more effectively” — T. Munzner
Why do we visualize data?

Total Bandwidth
(millions of bits per second)

Why Graphics?

Figures are richer; provide more information with less clutter and in less space.

Figures provide the gestalt effect: they give an overview; make structure more visible.

Figures are more accessible, easier to understand, faster to grasp, more comprehensible, more memorable, more fun, and less formal.

list adapted from: [Stasko et al. 1998]

D. Koop, CSCI 680/490, Spring 2022
Why Visual?

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>II</td>
<td>III</td>
<td>IV</td>
</tr>
<tr>
<td>x</td>
<td>y</td>
<td>x</td>
<td>y</td>
</tr>
<tr>
<td>10.0</td>
<td>8.04</td>
<td>10.0</td>
<td>9.14</td>
</tr>
<tr>
<td>10.0</td>
<td>8.04</td>
<td>10.0</td>
<td>9.14</td>
</tr>
<tr>
<td>10.0</td>
<td>8.04</td>
<td>10.0</td>
<td>9.14</td>
</tr>
<tr>
<td>10.0</td>
<td>8.04</td>
<td>10.0</td>
<td>9.14</td>
</tr>
<tr>
<td>10.0</td>
<td>8.04</td>
<td>10.0</td>
<td>9.14</td>
</tr>
</tbody>
</table>

[F. J. Anscombe]
Why Visual?

<table>
<thead>
<tr>
<th>I</th>
<th>II</th>
<th>III</th>
<th>IV</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>y</td>
<td>x</td>
<td>y</td>
</tr>
<tr>
<td>10.0</td>
<td>8.04</td>
<td>10.0</td>
<td>9.14</td>
</tr>
<tr>
<td>8.0</td>
<td>6.95</td>
<td>8.0</td>
<td>8.14</td>
</tr>
<tr>
<td>13.0</td>
<td>7.58</td>
<td>13.0</td>
<td>8.74</td>
</tr>
<tr>
<td>9.0</td>
<td>8.81</td>
<td>9.0</td>
<td>8.77</td>
</tr>
<tr>
<td>11.0</td>
<td>8.33</td>
<td>11.0</td>
<td>9.26</td>
</tr>
<tr>
<td>14.0</td>
<td>9.96</td>
<td>14.0</td>
<td>8.10</td>
</tr>
<tr>
<td>6.0</td>
<td>7.24</td>
<td>6.0</td>
<td>6.13</td>
</tr>
<tr>
<td>4.0</td>
<td>4.26</td>
<td>4.0</td>
<td>3.10</td>
</tr>
<tr>
<td>12.0</td>
<td>10.84</td>
<td>12.0</td>
<td>9.13</td>
</tr>
<tr>
<td>7.0</td>
<td>4.82</td>
<td>7.0</td>
<td>7.26</td>
</tr>
<tr>
<td>5.0</td>
<td>5.68</td>
<td>5.0</td>
<td>4.74</td>
</tr>
</tbody>
</table>

Mean of x: 9
Variance of x: 11
Mean of y: 7.50
Variance of y: 4.122
Correlation: 0.816

[F. J. Anscombe]
Why Visual?
Why Visual?

Mean of $x$ 9
Variance of $x$ 11
Mean of $y$ 7.50
Variance of $y$ 4.122
Correlation 0.816
Visual Pop-out
Visual Pop-out
Visual Pop-out
Visual Perception Limitations
Visual Perception Limitations

[Diagram of visual perception limitations]

[C. G. Healey]
Databases and Visualization?
Scalable Visualization

J. Heer