Multi-scale Analysis of Large, Distributed Data Sets in Resource Constrained Environments: Preliminary Results

Robert L. Grossman
University of Illinois at Chicago
&
Open Data Group
Resource-Constrained Data Mining

- Resource constrained data mining develops technology to support decision making involving data mining or analytic models when there are resource constraints.

- Examples of resource constraints:
  - Too much data to process
  - Too few analysts
  - Too little bandwidth
  - Too little time to adequately model the problem

- EOS scale problems vs. single investigator with data that fits on a desk top computer.
Earth Science Examples

- Detecting forest fires (work of Kirk Borne)
- Detecting tornadoes with potential for loss of human life
- Detecting tsunamis
- etc.
Two Cultures: Data Science vs Decision Support

<table>
<thead>
<tr>
<th>Goal</th>
<th>Gain understanding</th>
<th>Take an action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>collected &amp; cleaned until conclusions proven</td>
<td>analyzed until results are due</td>
</tr>
<tr>
<td>Methodology</td>
<td>hypothesis testing</td>
<td>lift of model</td>
</tr>
<tr>
<td>Challenge</td>
<td>data analysis</td>
<td>data access &amp; cleaning; implementation</td>
</tr>
<tr>
<td>Evaluation</td>
<td>results published</td>
<td>ROI, improvement over current decision support process</td>
</tr>
</tbody>
</table>
Three Challenges

1. **Access.** Getting access to the data.

2. **Modeling.** Developing enough analytic models to accurately reflect the data - data is often highly heterogeneous and may contain many different subpopulations and many different scales.

3. **Deployment.** Effectively deploying the analytics in a decision support environment. For example, dealing with false positives, etc.
High Performance Analytics -1990’s

- Data mining is viewed as algorithms over data.
- Models viewed as first class objects – summarized by XML files (Predictive Model Markup Language)
- Need for high performance scoring engines which consume models
- Computational grids of the 1990s are a good infrastructure for the data intensive computing required by data mining.
Attention shifts to derived data and the importance of data transformations.

Requirement for high performance web services to transform, integrate & score data

Early grids (without web services) probably not the best infrastructure, but better today
Attention will shift to data integration and workflow.

Incorporate scalable transport to move data.

Need for scalable web services for integrating and mining data.
Case Study 1: Accessing Remote and Distributed Earth Science Data


Accessing the Data: State of the Art in 2003

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Bandwidth</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP</td>
<td>5 Mbps</td>
<td>Standard as deployed</td>
</tr>
<tr>
<td>TCP</td>
<td>30 Mbps</td>
<td>Tuned 12 MB = 1 Gpbs * 110 ms</td>
</tr>
<tr>
<td>UDT</td>
<td>950 Mbps</td>
<td>Application layer protocol</td>
</tr>
</tbody>
</table>

Goal: Enrich 1 TB earth science data set using data from Amsterdam: NCAR data set
Finding Candidate Brown Dwarfs (SC 05)

- Sloan Digital Sky Survey (SDSS)
  - 82 million stars
  - Visible spectrum

- Two Micro All Sky Survey (2MASS)
  - 208 million stars
  - Infrared spectrum

- Two separate locations - Query at SC 05 in Seattle
  - SDSS in Tokyo & 2MASS in Chicago

- Found 289,283 Candidate Brown dwarfs
  - Common index structure for each cell in sky (metadata)
  - Simple computation - object in both locations; infrared value is >2 degree brighter

- Processed data disk to disk at over 1 Gbps sustained
Key Idea: Network Protocols Matter (Factors of 10x, 100x, 1000x)

2. Developed new application level network protocol - UDT
3. UDT is fair to other high volume data flows
4. UDT is friendly to commodity TCP flows.
5. UDT is easy to deploy since application level.

We developed streaming joins and data mining primitives over UDT.
Lessons Learned

1. Specialized application layer network protocols can effectively provide 10x, 100x or more improvement in end to end analytic applications over networks with high bandwidth delay products.

2. Data mining middleware is emerging that layers network protocols, storage and data services, and data mining primitives.

<table>
<thead>
<tr>
<th>Analytic applications</th>
<th>Data mining primitives</th>
<th>Storage and data services</th>
<th>Network protocols</th>
<th>Analytic stack</th>
</tr>
</thead>
</table>
Case Study 2: Modeling Highly Heterogeneous Data
Motivation

- Large, complex data sets often have structures at various levels – e.g. geospatial data sets
- One reason - processes at different scales may best be modeled by different physical laws
Dyadic Decompositions

- We enclose the data set in a cube.
- Recursively bisect one or more edges of cube at scale $k$ to define cubes at scale $k+1$. 
Basic Definitions

- We define cubes noise cubes, neighbors and outlier cubes, and clusters.
- Fix a threshold $N$ noise. A cube is a noise cube in case it contains fewer than $N$ noise points and is adjacent to some other cube with more than $N$ noise points.
- Two points are neighbors iff:
  1. They belong to the same $k$-cube or
  2. They belong to adjacent $k$-cubes and neither cube is a noise cube.
Forest Cover Data

- This data contained forest cover type for 30 x 30 meter cells obtained from US Forest Service (USFS) Region 2 Resource Information System (RIS) data.
- Number of instances (observations) 581012
- Number of Attributes 54 : 12 measures, but 54 columns of data (10 quantitative variables, 4 binary wilderness areas and 40 binary soil type variables)
- Forest Cover Types and Their Distribution:
## Preliminary Experimental Results

<table>
<thead>
<tr>
<th>Noise Threshold</th>
<th>Min Size</th>
<th>% Correct Labels</th>
<th>% Incorrect Labels</th>
<th>% Outlier Labels</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>200</td>
<td>81.32</td>
<td>11.36</td>
<td>7.31</td>
</tr>
<tr>
<td>1</td>
<td>500</td>
<td>80.12</td>
<td>12.62</td>
<td>7.24</td>
</tr>
<tr>
<td>1</td>
<td>800</td>
<td>78.42</td>
<td>14.44</td>
<td>7.12</td>
</tr>
<tr>
<td>2</td>
<td>200</td>
<td>81.44</td>
<td>11.24</td>
<td>7.31</td>
</tr>
<tr>
<td>2</td>
<td>500</td>
<td>79.39</td>
<td>13.48</td>
<td>7.12</td>
</tr>
<tr>
<td>2</td>
<td>800</td>
<td>76.54</td>
<td>17.32</td>
<td>6.15</td>
</tr>
<tr>
<td>4</td>
<td>200</td>
<td>81.83</td>
<td>11.50</td>
<td>6.65</td>
</tr>
<tr>
<td>4</td>
<td>500</td>
<td>80.38</td>
<td>12.97</td>
<td>6.64</td>
</tr>
<tr>
<td>4</td>
<td>800</td>
<td>78.32</td>
<td>15.31</td>
<td>6.36</td>
</tr>
</tbody>
</table>
Case Study 3: Deployment Challenges
Improving Alerts Using Doppler Radar Data

- Is the traffic speed and volume today (Tuesday, Nov. 15, 3 pm, convention event, no rain) different than the baseline?
- If so, send an alert to a PDA.

- 833 road sensors
- weather data (images, xml)
- text data about special events

Working with your heterogeneous terabytes.
Key Idea 1: Build $10^4+$ Models

1. Divide & conquer data (segment) using multidimensional data cubes
2. For each distinct cube, estimate parameters for separate statistical model
3. Detect changes from baselines and send alerts in real time
Drill Down/Aggregate to Balance Meaningful/Manageable Tradeoff

- More alerts
- Alerts more meaningful
- To increase alerts, add breakpoint to split cubes, order by number of new alerts, & select one or more new breakpoints

- Fewer alerts
- Alerts more manageable
- To decrease alerts, remove breakpoint, order by number of decreased alerts, & select one or more breakpoints to remove

One model for each cell in data cube