Interactive Data Exploration with Constraints

Alexander Kalinin, Ugur Cetintemel, Stan Zdonik

Brown University

Interactive Data Exploration
- Human-in-the-loop ad-hoc data analysis
- Key for discovery
- Fundamentally a search problem
- Current DBMSs fall short:
  - No exploration-oriented query constructs
  - No interactive operation

Our Approach: Constraint-based Exploration
- Constraint satisfaction problem (CSP):
  - decision variables
  - domains for the variables
  - any additional constraints
- Focus on multidimensional data

Semantic Windows Framework
- First step towards a general constraint-based exploration framework
- “Semantic windows” → rectangular regions specified by constraints on their shapes and contents

Example Queries
1. SDSS: Find areas (sub-arrays) with “interesting” properties, e.g.,
   “Find all 3° by 2° celestial regions where average brightness of all stars is greater than A”

2. NYSE: Find “interesting” time intervals, e.g.,
   “Find all 2-5 months periods when the average tech stock prices were higher than $40/share”

Constraints on Arrays
- Geometric variables → shape/placement of sub-arrays
  - The number of rows/columns in the sub-array
  - The leftmost coordinate.
- Functional variables → contents
  - The total sum of elements
  - Min/Max values of the elements
  - Rank
- Additional constraints → any additional requirements:
  - Similarity between sub-arrays
  - Distance
  - Application-specific requirements

The Search Process
- Use branch-and-cut
- The current state describes all candidate sub-arrays
- Search continues by choosing a sub-array, splitting the search area, binding a variable, etc.
- The algorithm is any-time: search can stop at any sub-array or continue for more results

Approach
- Best-first data-driven graph search
- For each candidate window, estimate:
  - Benefit: distance of the constraint’s function to the required value
  - Cost: cost of reading the window from disk
- Windows are explored in the order of utilities, which combine cost and benefit
- Enables online results
- Optimizations:
  - Sampling, Guides estimations and search
  - Pruning, Eliminates windows fast
  - Caching, Reduces disk reads
  - Prefetching, Reduces overhead of reads

Distributed Architecture

Advanced Query Types
- Anomaly queries, e.g.,
  “The difference between the element and its neighborhood is beyond a threshold”
- Second-order queries, e.g.,
  “Relationships between sub-arrays, e.g., sub-arrays are similar and do not overlap”
- Optimization queries, e.g.,
  “The sub-array has the maximum sum of its elements”

Challenges
- Minimum invasion approach
- Decision builders to steer the search
- Exploring each and every sub-array is inefficient:
  - Use additional structures, like aggregate trees for estimations
- Big arrays are distributed:
  - Split the search to the nodes
  - Explore sub-arrays in parallel

Ongoing Work
- An initial prototype using Google’s OR-tools:
  - Extending the engine with constraints and decision builders
- Putting the search inside SciDB:
  - Additional structures, like aggregate trees for estimations
  - Integrating the SciDB query engine with the constraint satisfaction engine

Experimental Results Highlights

<table>
<thead>
<tr>
<th>Number of nodes</th>
<th>First result, s</th>
<th>All results, s</th>
<th>Completion time, s</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>6</td>
<td>820</td>
<td>1,820</td>
</tr>
<tr>
<td>2</td>
<td>6</td>
<td>470</td>
<td>1,050</td>
</tr>
<tr>
<td>4</td>
<td>8</td>
<td>360</td>
<td>880</td>
</tr>
<tr>
<td>8</td>
<td>7</td>
<td>200</td>
<td>350</td>
</tr>
</tbody>
</table>

• SciDB and PostgreSQL as backend DBMSs

Interactive Data Exploration • Human-in-the-loop ad-hoc data analysis • Key for discovery • Fundamentally a search problem • Current DBMSs fall short: • No exploration-oriented query constructs • No interactive operation