SciCSM: Novel Contrast Set Mining over Scientific Datasets Using Bitmap Indices

Gangyi Zhu, Yi Wang, Gagan Agrawal

The Ohio State University
Outline

- Introduction
- SciCSM Algorithm
- Pruning and Combination
- Bitmap-Based Optimization
- Experiment Results
- Conclusion
Introduction

• Goal
  – Identify all the interesting contrast sets, which are described as conjunctions of attributes value pairs and have significant difference in different groups

• Example
  – Given two datasets of climate simulation for the same spatial area but different time periods, the key differences between two datasets, which can be taken as climate changes, is of great interest

• Challenges for Array-based Scientific Data
  – Both value-based and dimension-based attributes are involved
  – The attributes are mostly numeric for scientific datasets
  – Scientific array dataset of interest to us can be extremely large
SciCSM Algorithm

• Search Strategy
  – Discretize each attribute range into a number of bins
  – A set-enumeration tree is adopted as foundation of search strategy
  – All attributes are enumerated in a certain order, and every node represents a contrast set

• Exhaustive but Efficient Search
  – Level-wise (top-down)
  – Pruning + combination
  – Each node is visited only once or not visited at all (if pruned)
Pruning and Combination

• Pruning Strategy
  – **Minimum Absolute Quality**: sufficient quality
  – **Minimum Relative Quality**: sufficient quality improvement compared with its parent

• Combination Strategy
  – **Existence of Adjacency**: only adjacent nodes can be combined
  – **Combined Quality**: combined node should have sufficient quality
  – **Difference Homogeneity**: support/mean difference in two combining nodes should have the same trend (both positive or negative)
  – **Distribution Purity**: introduce a measure **Continuous Weighted Entropy (CWE)** to control the granularity of output contrast sets

\[
CWE = -\sum_{i=1}^{n} p(i) \times \log p(i) \times \frac{|v_i - M|}{v_n - v_1}
\]
Bitmap-Based Optimization

- Our algorithm entirely operates on bitmap indices
- Key Insights
  - Each attribute-range pair can be represented by a disjunction of bitvectors
  - Each contrast set can be represented by a conjunction of bitvectors
  - All the statistics like mean, CWE, and quality can be calculated entirely based on bitmap

A Bitmap Indexing Example in SciCSM

<table>
<thead>
<tr>
<th>Index</th>
<th>Val</th>
<th>Bitvector1</th>
<th>Bitvector2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>1100</td>
<td>1100</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>0011</td>
<td>0001</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>1010</td>
<td>1010</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>0101</td>
<td>0101</td>
</tr>
<tr>
<td>Vector for Dimension-based</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Vector for Value-based A</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Vector for Value-based B</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(a) Conventional Indices

(b) Indices for Dimension-based Attributes and Value-based Attributes Bitvectors
Experiment Results

• Hardware: A machine with 48 GB of main memory and Intel® Xeon(R) CPU X5650 2.67 GHz CPU

• Quality Evaluation

<table>
<thead>
<tr>
<th>Rank</th>
<th>Contrast Set</th>
<th>Quality</th>
<th>Mean Diff</th>
<th>Sup. Diff</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>14.47 ≤ TEMP ≤ 15.00 AND 35.25 ≤ SALT ≤ 35.37</td>
<td>1.056</td>
<td>0.55%</td>
<td>7.00%</td>
</tr>
<tr>
<td>2</td>
<td>14.47 ≤ TEMP ≤ 15.00 AND 34.91 ≤ SALT ≤ 35.15</td>
<td>1.056</td>
<td>0.47%</td>
<td>6.70%</td>
</tr>
<tr>
<td>3</td>
<td>14.47 ≤ TEMP ≤ 15.00 AND 35.90 ≤ SALT ≤ 34.10</td>
<td>1.039</td>
<td>3.55%</td>
<td>1.17%</td>
</tr>
<tr>
<td>4</td>
<td>21.46 ≤ TEMP ≤ 37.74</td>
<td>1.034</td>
<td>0.94%</td>
<td>2.44%</td>
</tr>
<tr>
<td>5</td>
<td>34.92 ≤ SALT ≤ 34.94 AND -179.50 &lt; lon &lt; -120.50</td>
<td>1.033</td>
<td>3.26%</td>
<td>0.00%</td>
</tr>
<tr>
<td>6</td>
<td>-3.92 ≤ TEMP ≤ 1.25</td>
<td>1.031</td>
<td>1.87%</td>
<td>1.23%</td>
</tr>
</tbody>
</table>

• Performance Evaluation
Conclusion

• Functionality
  – Capable of handling arrays comprising value-based attributes and dimension-based attributes
  – Can effectively process numeric attributes

• Bitmap Acceleration
  – Compact input leads to I/O and storage advantage
  – Fast bitwise operations support highly efficient computation
  – Provide ability of processing larger datasets

• Efficiency and Effectiveness
  – Both high efficiency and effectiveness are demonstrated by extensive experiments on multiple real-life datasets