Scorpion

Explaining Away Outliers in Aggregate Queries

Eugene Wu
Samuel Madden
VLDB 2013

Presented by: Sanaz
<table>
<thead>
<tr>
<th>Tuple id</th>
<th>Time</th>
<th>SensorID</th>
<th>Voltage</th>
<th>Humidity</th>
<th>Temp.</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1</td>
<td>11AM</td>
<td>1</td>
<td>2.64</td>
<td>0.4</td>
<td>34</td>
</tr>
<tr>
<td>T2</td>
<td>11AM</td>
<td>2</td>
<td>2.65</td>
<td>0.5</td>
<td>35</td>
</tr>
<tr>
<td>T3</td>
<td>11AM</td>
<td>3</td>
<td>2.63</td>
<td>0.4</td>
<td>35</td>
</tr>
<tr>
<td>T4</td>
<td>12PM</td>
<td>1</td>
<td>2.7</td>
<td>0.3</td>
<td>35</td>
</tr>
<tr>
<td>T5</td>
<td>12PM</td>
<td>2</td>
<td>2.7</td>
<td>0.5</td>
<td>35</td>
</tr>
<tr>
<td>T6</td>
<td>12PM</td>
<td>3</td>
<td>2.3</td>
<td>0.4</td>
<td>100</td>
</tr>
<tr>
<td>T7</td>
<td>1PM</td>
<td>1</td>
<td>2.7</td>
<td>0.3</td>
<td>35</td>
</tr>
<tr>
<td>T8</td>
<td>1PM</td>
<td>2</td>
<td>2.7</td>
<td>0.5</td>
<td>35</td>
</tr>
<tr>
<td>T9</td>
<td>1PM</td>
<td>3</td>
<td>2.3</td>
<td>0.5</td>
<td>80</td>
</tr>
</tbody>
</table>

**Table 1: Example tuples from sensors table**

<table>
<thead>
<tr>
<th>Result id</th>
<th>Time</th>
<th>AVG(temp)</th>
<th>Label</th>
<th>v</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha_1$</td>
<td>11AM</td>
<td>34.6</td>
<td>Hold-out</td>
<td>-</td>
</tr>
<tr>
<td>$\alpha_2$</td>
<td>12PM</td>
<td>56.6</td>
<td>Outlier</td>
<td>$&lt; -1 &gt;$</td>
</tr>
<tr>
<td>$\alpha_3$</td>
<td>1PM</td>
<td>50</td>
<td>Outlier</td>
<td>$&lt; -1 &gt;$</td>
</tr>
</tbody>
</table>

**Table 2: Query results (left) and user annotations (right)**
Exploratory Analysis

Mean and standard deviation of temperature readings from Intel sensor dataset.
Use Cases:

• **Intel Data set:**
  
  **Q:** Why the average temperature at 12PM and 1PM are unexpectedly high?
  
  **A:** The high temperature caused by sensors near windows that heat up under the sun around noon and another sensor running out of energy that starts producing erroneous readings.

• **Medical Cost Analysis:** Amongst a population of cancer patients, the top 15% of patients by cost represented more than 50% of the total dollars spent.

  **Q:** Were these patients significantly sicker? Did they have significantly better or worse outcomes than the median-cost patient.

  **A:** Small number of doctors were over-prescribing these procedures, which were presumably not necessary because the outcomes didn’t improve.

• **Election Campaign Expenses**
Goal: Predicate generation:

Describing the common properties of the input data points or tuples that caused the outlier outputs.

“explain” why they are outliers

Constructing a predicate over the input attributes that filter out the points in the responsible subset without removing a large number of other, incidental data points.

A predicate, $p$, is a conjunction of range clauses over the continuous attributes and set containment clauses over the discrete attributes, where each attribute is present in at most one clause.
Setup:

Input:

➔ D: a single relation with attributes $A = \text{attr}_1, \ldots, \text{attr}_k$
➔ Q: a group-by SQL query grouped by attributes $A_{gb} \subset A$,
➔ agg(): a single aggregate function that computes a result using aggregate attributes $A_{agg} \subset A$ from each tuple and $A_{agg} \cap A_{gb} = \emptyset$
➔ H: hold-out set
➔ O: outlier set
Predicate Influence:

\[
\begin{align*}
\Delta_{agg}(o, p) &= \text{agg}(g_o) - \text{agg}(g_o - p(g_o)) \\
inf_{agg}(o, p) &= \frac{\Delta o}{\Delta g_o} = \frac{\Delta_{agg}(o, p)}{|p(g_o)|} \\
in_{agg}(o, p, v_o) &= inf_{agg}(o, p) \cdot v_o \\
in_{agg}(o, h, p, v_o) &= \lambda inf_{agg}(o, p, v_o) - (1 - \lambda)|inf_{agg}(h, p)|
\end{align*}
\]
Influential Predicate Problem:

Given a select-project-group-by query $Q$, and user inputs $O$, $H$, $\lambda$ and $V$, find the predicate, $p^*$, from the set of all possible predicates, $P_{\text{Rest}}$, that has the maximum influence:

$$p^* = \arg \max_{p \in P_{A_{\text{rest}}}} \inf(f(p))$$
Scorpion Architecture:
Each point represents a tuple. Darker color means higher influence. (b) Output of *Partitioner*. (c) Output of *Merger*
Naive Partitioner, Basic Merger:

**Partitioner:**

- Defines all distinct single-attribute clauses, then enumerates all conjunctions of up to one clause from each attribute.
- The clauses over a discrete attribute, $A_i$, are of the form, “$A_i \text{ in } (\cdot \cdot \cdot)$” where the $\cdot \cdot \cdot$ is replaced with all possible combinations of the attribute’s distinct values.
- Clauses over continuous attributes are constructed by splitting the attribute’s domain into a fixed number of equi-sized ranges, and enumerating all combinations of consecutive ranges.

**Merger:**

- Two predicates are merged by computing the minimum bounding box of the continuous attributes and the union of the values for each discrete attribute.
- Each predicate is expanded by greedily merging it with adjacent predicates until the resulting influence does not increase.
Decision Tree (DT) Partitioner:

A top-down partitioning algorithm for independent aggregates.

*DT* recursively splits the attribute space of an input group to create a set of predicates

**Intuition:** the $\Delta$ function of independent operators cannot decrease when tuples with similar influence are combined together.
Bottom-Up (MC) Partitioner

*MC*: a bottom-up approach for independent, anti-monotonic aggregates.

**Idea:**
- First search for influential single-attribute predicates
- Then intersect them to construct multi-attribute predicates (similar to subspace clustering)
function \text{MC}(O, H, V) \\
\text{predicates} \leftarrow \text{Null} \\
\text{best} \leftarrow \text{Null} \\
\text{while} |\text{predicates}| > 0 \text{ do} \\
\quad \text{if} \ \text{predicates} = \text{Null} \text{ then} \\
\quad \quad \text{predicates} \leftarrow \text{initialize\_predicates}(O, H) \\
\quad \text{else} \\
\quad \quad \text{predicates} \leftarrow \text{intersect}(\text{predicates}) \\
\quad \text{best} \leftarrow \text{arg max}_{p \in \text{merged}} \text{inf}(p) \\
\quad \text{predicates} \leftarrow \text{prune}(\text{predicates}, O, V, \text{best}) \\
\quad \text{merged} \leftarrow \text{Merger}(\text{predicates}) \\
\quad \text{merged} \leftarrow \{p|p \in \text{merged} \wedge \text{inf}(p) > \text{inf}(\text{best})\} \\
\quad \text{if} \ \text{merged}\_\text{length} = 0 \text{ then} \\
\quad \quad \text{break} \\
\quad \text{predicates} \leftarrow \{p|\exists_{p_m \in \text{merged}} p < D p_m\} \\
\quad \text{best} \leftarrow \text{arg max}_{p \in \text{merged}} \text{inf}(p) \\
\text{return} \ \text{best} \\
\text{function} \ \text{PRUNE}(\text{predicates}, O, V, \text{best}) \\
\text{ret} = \{p \in \text{predicates}|\text{inf}(O, \emptyset, p, V) < \text{inf}(\text{best})\} \\
\text{ret} = \{p \in \text{ret}|\text{arg max}_{t^* \in P(O)} \text{inf}(t^*) < \text{inf}(\text{best})\} \\
\text{return} \ \text{ret}
Experiments:

The SQL query contains an independent anti-monotonic aggregate and is of the form:

\[
\text{SELECT SUM}(A_v) \text{ FROM synthetic GROUP BY } A_d
\]

10 distinct \(A_d\) values (to create 10 groups)

The \(A_v\) values are drawn from one of three gaussian distributions:
- Normal tuples: \(N(10, 10)\)
- high-valued outliers: \(N(\mu,10)\)
- medium valued outliers: \(N(\mu+10/2, 10)\)
SYNTH Dataset:
Optimal NAIVE Predicates:

\[ \inf_{agg}(o, p, c) = \frac{\Delta o}{(\Delta g_o)^c} \]

The exponent, \( c \geq 0 \), trades off the importance of keeping the size of affected tuples small and maximizing the change in the aggregate result.
Accuracy Statistics of NAIVE

![Graphs showing accuracy statistics for SYNTH-2D-Easy and SYNTH-2D-Hard datasets. The graphs plot F-Score, Precision, and Recall against the C parameter for Ground Truth, Inner, and Outer models.](Image)
Accuracy Measures:
Cost as dimensionality of dataset:

The diagrams illustrate the cost (in seconds, log scale) for different algorithms (DT, MC, Naive) as the C parameter varies. The graphs show that the cost generally increases with the dimensionality of the dataset, starting from 2D, 3D, and 4D, respectively. Each algorithm shows a distinct pattern, with DT and MC generally performing better than the Naive approach in terms of cost efficiency.
Accuracy Statistics of NAIVE

Graph showing the change of various statistical measures (F-Score, Precision, Recall) with cost (in seconds) for inner and outer regions, with different costs (C) represented by different line styles (0, 0.1, 0.5).
Finding Outliers:

(k, l)-means: Given a set of points \( X = \{x_1, \ldots, x_n\} \), a distance function \( d: X \times X \to \mathbb{R} \) and numbers \( k \) and \( l \), find a set of \( k \) points \( C = \{c_1, \ldots, c_k\} \) and a set of \( l \) points \( L \subseteq X \) so as to minimize the error

\[
E(X, C, L) = E(X \setminus L, C)
\]

Where

\[
E(X, C) = d(x \mid C) \quad x \in X
\]

and

\[
d(x \mid C) = \min\{d(x, c)\} \quad c \in C
\]
Finding Predicates:

Given a set of points \( X = \{x_1, ..., x_n\} \), a distance function \( d: X \times X \rightarrow \mathbb{R} \) and a set of outliers \( L \),

\[
\text{influence} \ (p) = |p(L)| - |p(X \setminus L)|
\]

is maximized

or (\( L \): number of outliers)

\[
\mathcal{E}(X, C, L) = \mathcal{E}(X \setminus p(X), C)
\]

s.t. \( |p(X)| \leq L \)

is minimized
Finding Anomalies:

Given a set of points $X = \{x_1, \ldots, x_n\}$, an aggregate function $agg$, number of outliers $l$, and a target value $t$,

$$| agg(x \setminus y) - t |$$

and $|y| \leq l$

is minimized.
Thank You!