Scorpion

Explaining Away Outliers in Aggregate Queries

Eugene Wu
Samuel Madden
VLDB 2013

Presented by: Sanaz

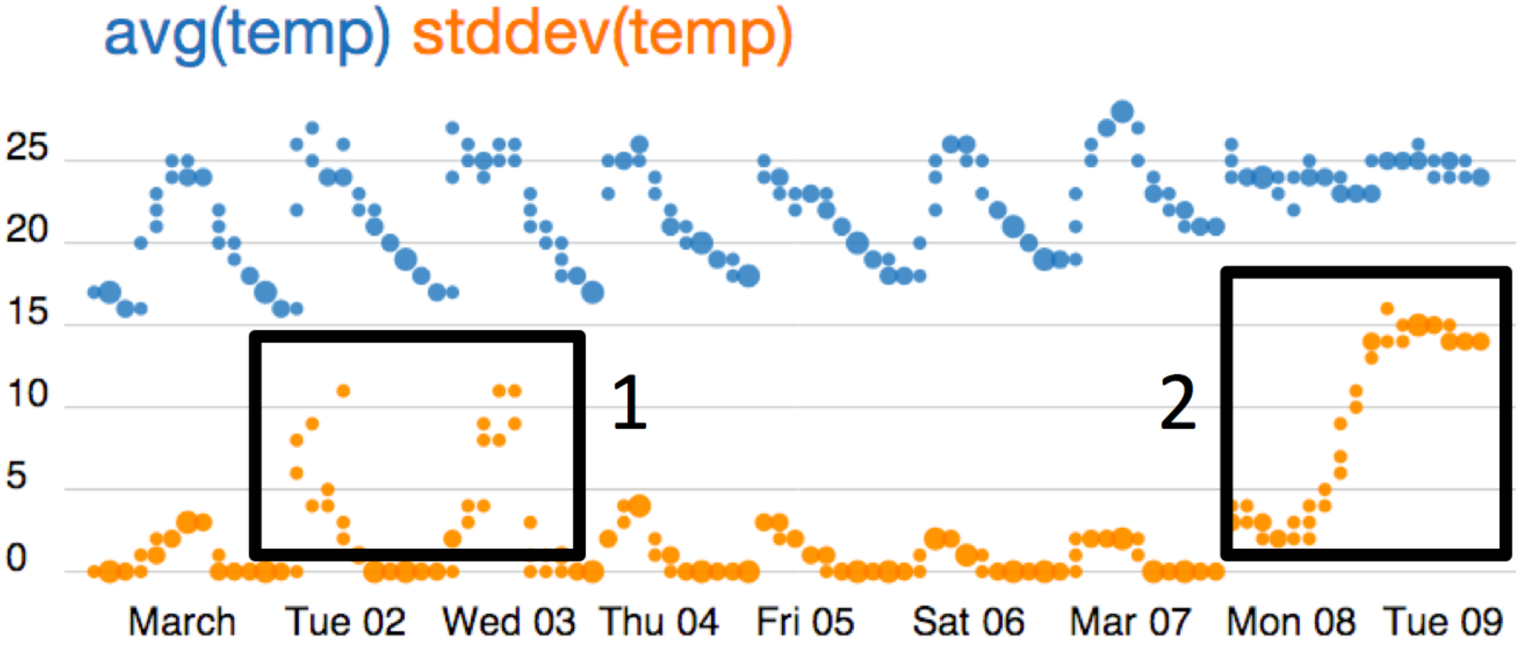
Tuple id	Time	SensorID	Voltage	Humidity	Temp.
T1	11AM	1	2.64	0.4	34
T2	11AM	2	2.65	0.5	35
T3	11AM	3	2.63	0.4	35
T4	12PM	1	2.7	0.3	35
T5	12PM	2	2.7	0.5	35
T6	12PM	3	2.3	0.4	100
T7	1PM	1	2.7	0.3	35
T8	1PM	2	2.7	0.5	35
T9	1PM	3	2.3	0.5	80

Table 1: Example tuples from sensors table

Result id	Time	AVG(temp)	Label	v
α_1	11AM	34.6	Hold-out	-
α_2	12PM	56.6	Outlier	$< -1 >$
α_3	1PM	50	Outlier	$< -1 >$

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 $A = attr_1, .., attr_k$
- ➔ 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:

Influence of p	$\Delta_{agg}(o,p) = agg(g_o) - agg(g_o - p(g_o))$
Influence Ratio	$inf_{agg}(o,p) = \frac{\Delta o}{\Delta g_o} = \frac{\Delta_{agg}(o,p)}{\lvert p(g_o)\rvert}$
Error Vector	$inf_{agg}(o,p,v_o) = inf_{agg}(o,p) \bullet v_o$
Hold-out Result	$inf_{agg}(o,h,p,v_o) = \lambda inf_{agg}(o,p,v_o) - (1-\lambda)\lvert inf_{agg}(h,p)\rvert$

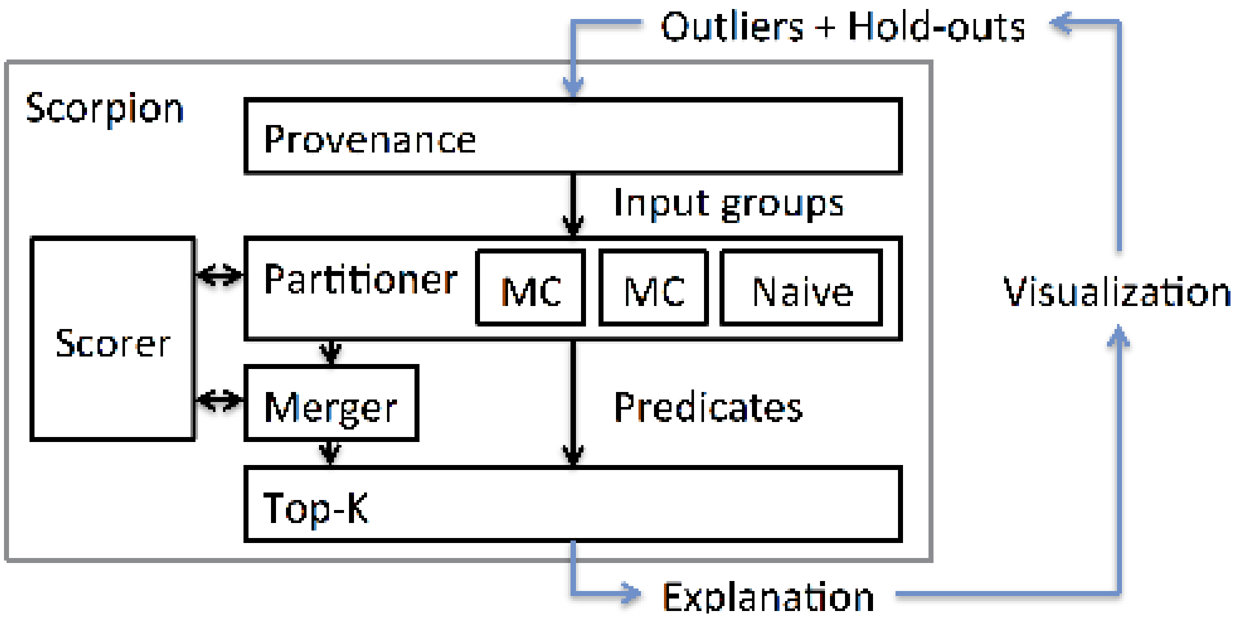
Influential Predicate Problem:

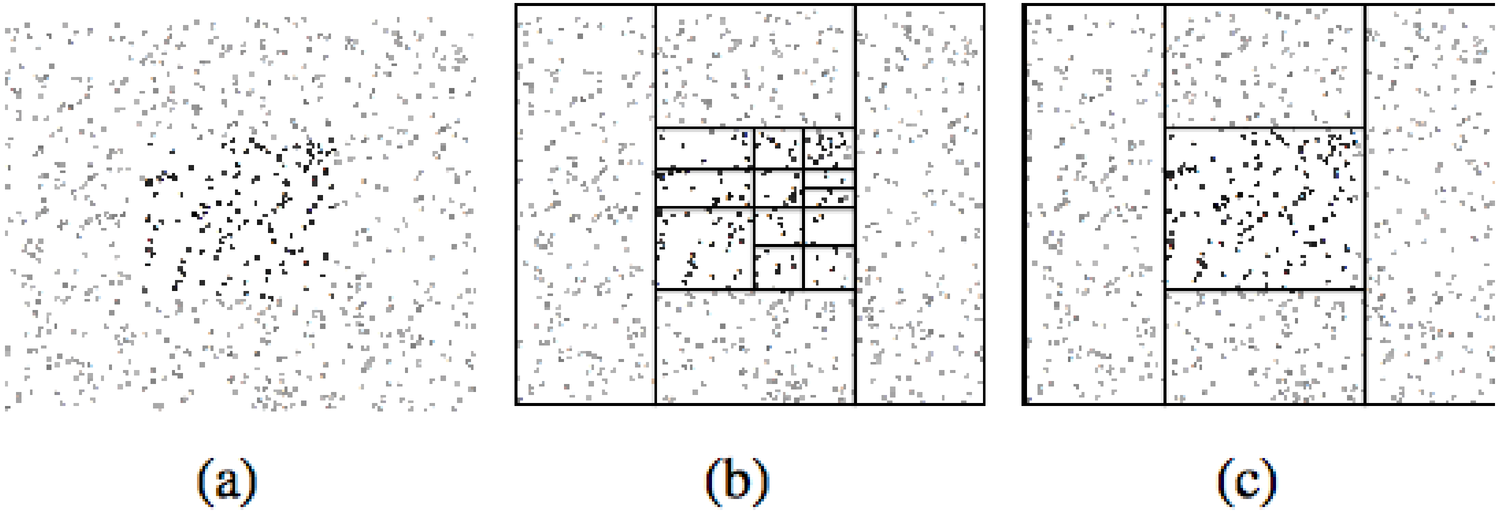
Given a select-project-group-by query Q, and user inputs O, H, λ and V ,

find the predicate, p^*, from the set of all possible predicates, P_{Rest} , that has the maximum influence:

$$p^* = \arg\max_{p \in P_{A_{rest}}} inf(p)$$

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 in (· · ·)” where the · · · 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 Δ 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 MC(O, H, V)
  $predicates \leftarrow Null$
  $best \leftarrow Null$
  while $|predicates| > 0$ **do**
    if $predicates = Null$ **then**
      $predicates \leftarrow \text{initialize\_predicates}(O, H)$
    else
      $predicates \leftarrow \text{intersect}(predicates)$
    $best \leftarrow \arg\max_{p \in merged} inf(p)$
    $predicates \leftarrow \text{prune}(predicates, O, V, best)$
    $merged \leftarrow \text{Merger}(predicates)$
    $merged \leftarrow \{p | p \in merged \wedge inf(p) > inf(best)\}$
    if $merged.length = 0$ **then**
      break
    $predicates \leftarrow \{p | \exists_{p_m \in merged} p \prec_D p_m\}$
    $best \leftarrow \arg\max_{p \in merged} inf(p)$
  return $best$

function PRUNE($predicates, O, V, best$)
  $ret = \{p \in predicates | \inf(O, \emptyset, p, V) < \inf(best)\}$
  $ret = \{p \in ret | \arg\max_{t^* \in p(O)} \inf(t^*) < \inf(best)\}$
  return ret

Experiments:

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

SELECT SUM(A_v) FROM synthetic GROUP BY A_d

10 distinct A_d values (to create 10 groups)

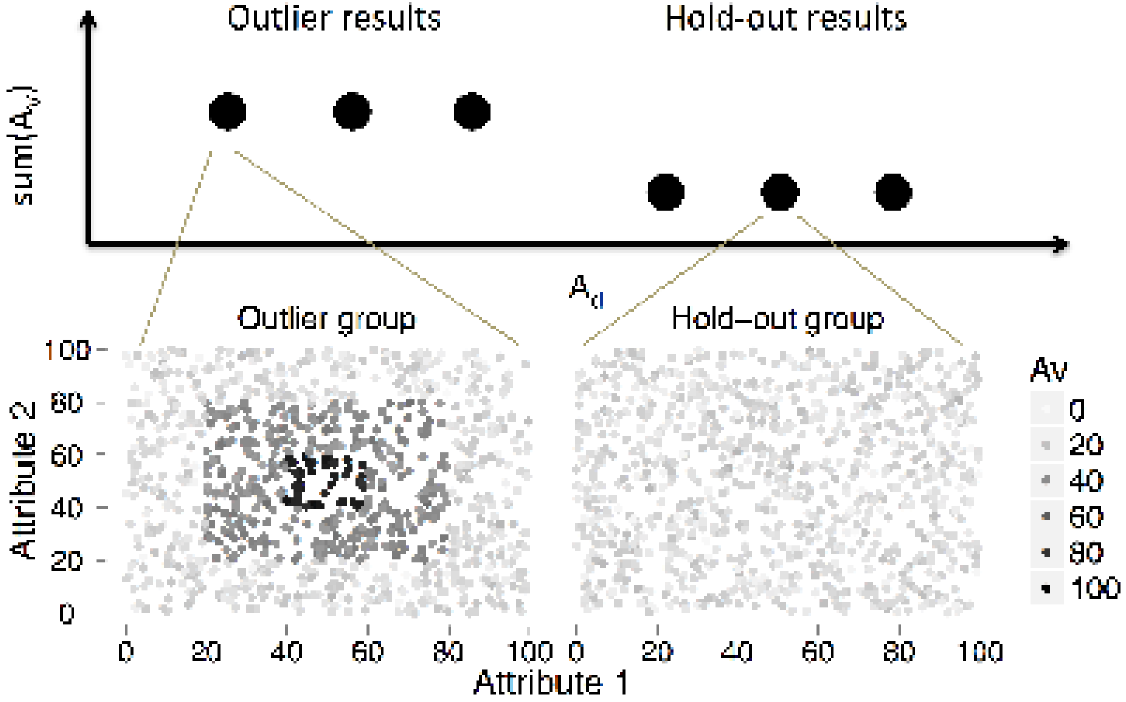
The A_v values are drawn from one of three gaussian distributions:

Normal tuples: $N(10, 10)$

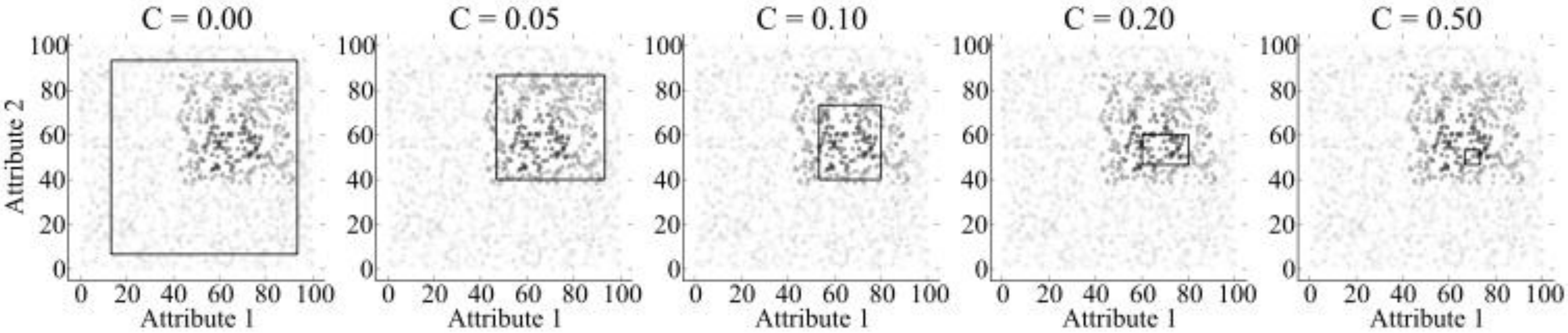
high-valued outliers: $N(\mu,10)$

medium valued outliers: $N(\mu+10/2, 10)$

SYNTH Dataset:



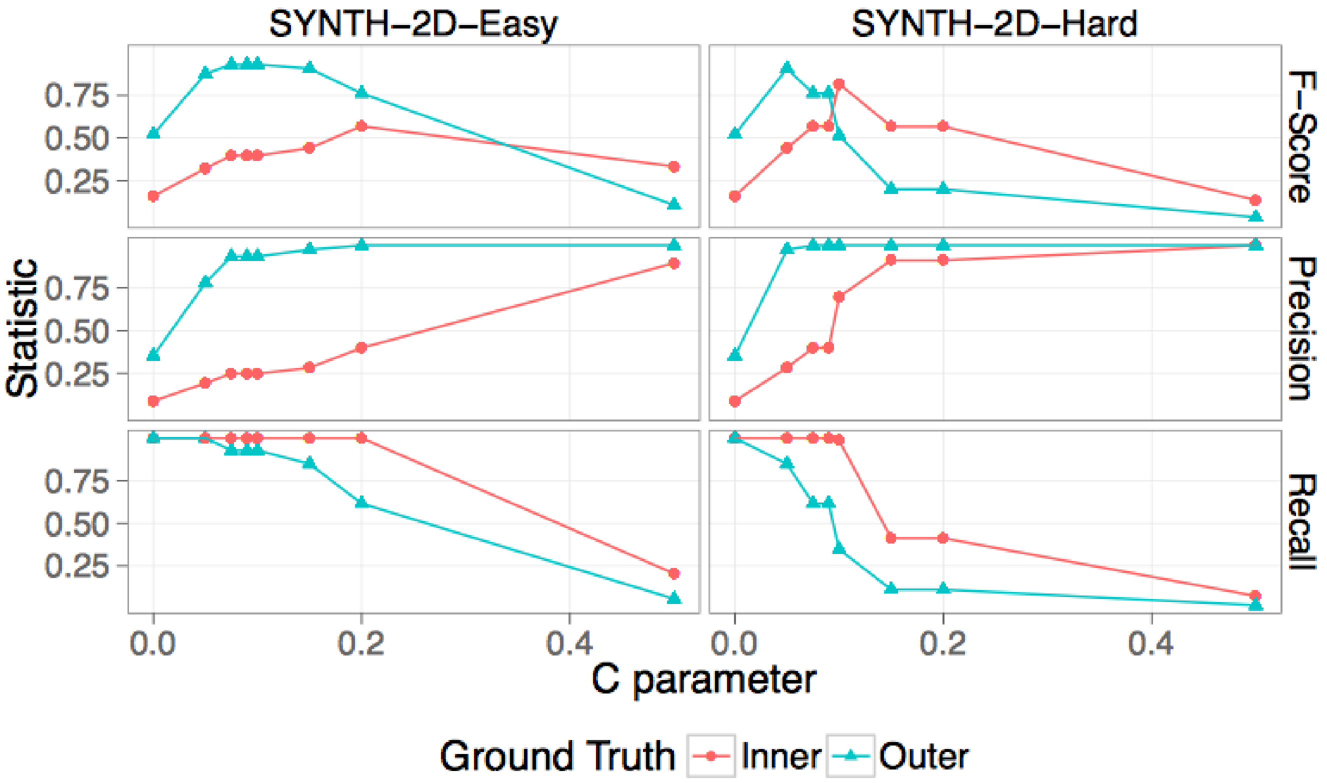
Optimal NAIVE Predicates:



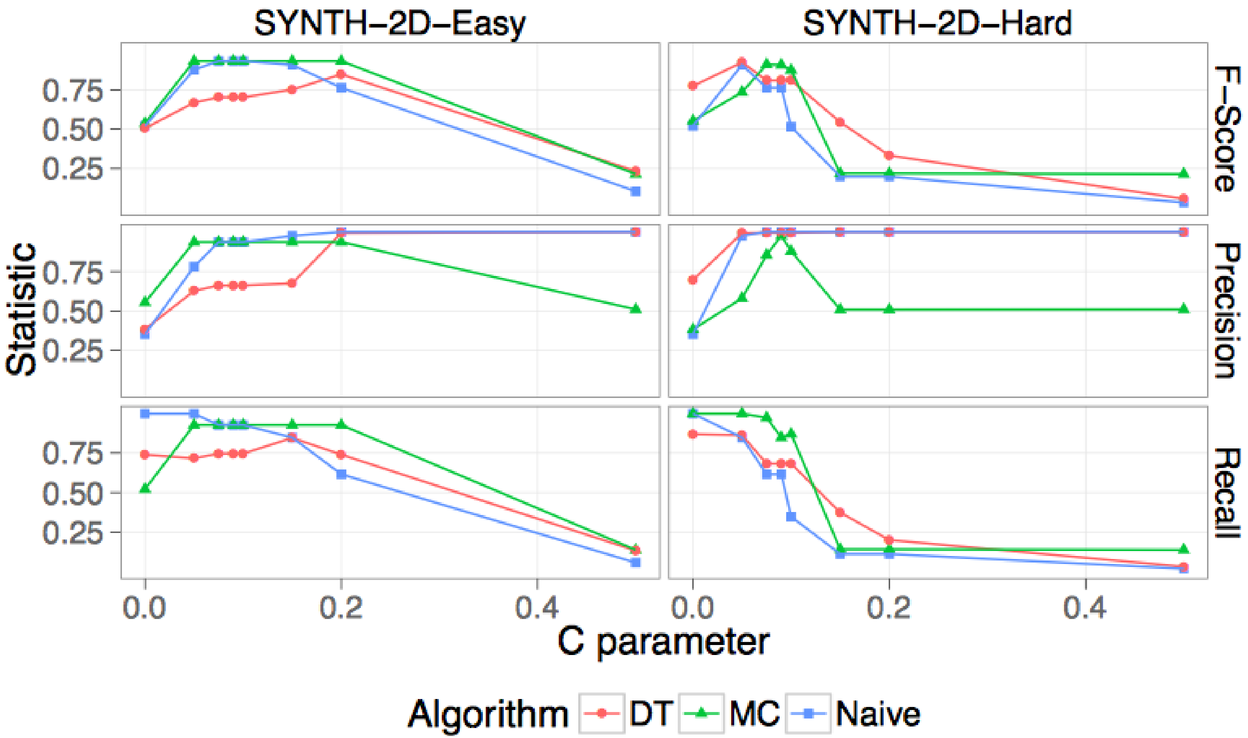
$$inf_{agg}(o, p, c) = \frac{\Delta o}{(\Delta g_o)^c}$$

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.

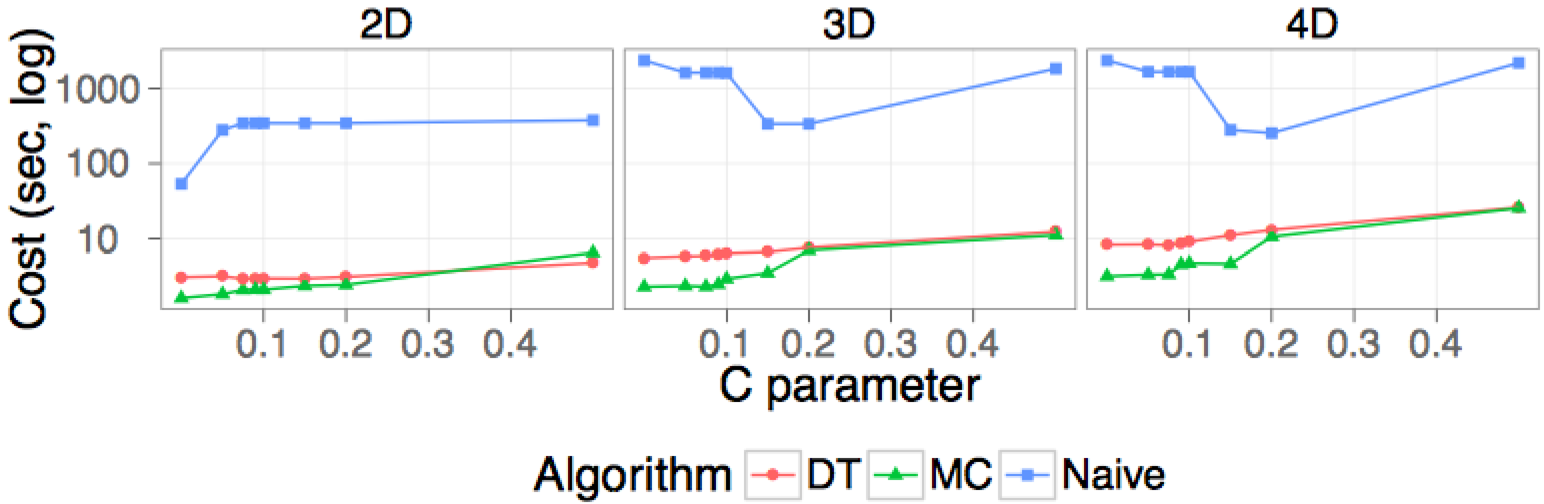
Accuracy Statistics of NAIVE



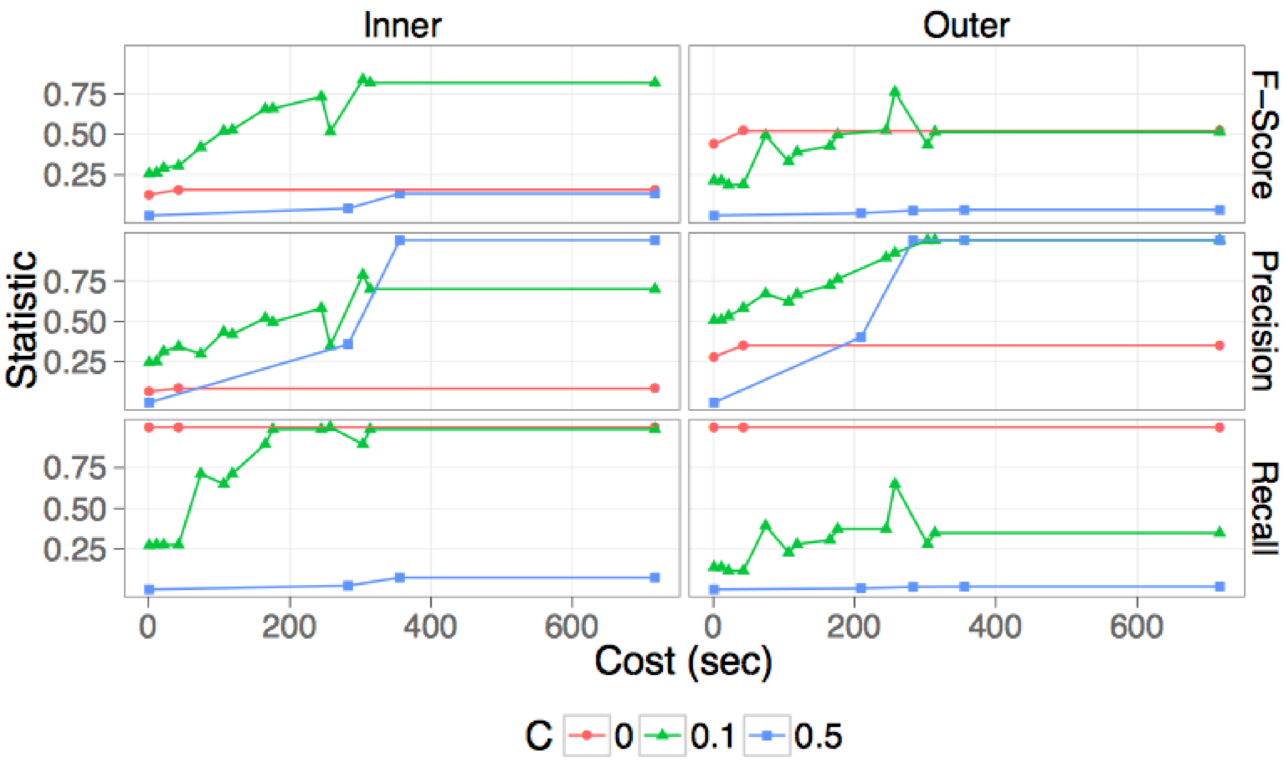
Accuracy Measures:



Cost as dimensionality of dataset:



Accuracy Statistics of NAIVE



Finding Outliers:

(k, l)-means: Given a set of points $X = \{x_1,...,x_n\}$,a distance function $d:X\times X\rightarrow R$ and numbers k and l,

find a set of k points $C=\{c_1 ,...,c_k \}$ and a set of l points $L \subseteq X$ so as to minimize the error

$$E(X,C,L)=E(X\backslash L,C)$$

Where

$$E(X, C) = \ d(x \mid C) \ \ x\in X$$

and

$$d(x \mid C) = \min\{d(x, c)\} \ \ c\in C$$

Finding Predicates:

Given a set of points $X = \{x_1,...,x_n\}$,a distance function $d:X\times X\to R$ and as et of outliers L ,

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

is maximized

or (L: number of outliers)

$$E(X,C,L)=E(X \setminus p(X), C)$$
$$\text{s.t. } |p(X)| <= L$$

is minimized

Finding Anomalies:

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

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

$$\text{and } |y| <= l$$

is minimized.

Thank You!