Macrobase: Prioritizing Attention in Fast Data

CS265: Harshita Gupta
machine generated data: 12M events recorded per second
AWS datacenter anomalies, unusual behavior in electricals, bugs in new application releases
Too much data to manually screen!
Existing Analytical Insight Systems
Feature Generation (timeseries etc)

Analysis and Aggregation Operators:
- Highlighting,
- Grouping,
- Contextualizing Behaviors

Input data

Insights
Feature Generation (timeseries etc)

Analysis and Aggregation Operators:
Highlighting, Grouping, Contextualizing Behaviors

Not all insights are useful
Input data

Data changes!

Feature Generation (timeseries etc)

Analysis and Aggregation Operators:
Highlighting, Grouping, Contextualizing Behaviors

Insights

Not all insights are useful
Input data

Feature Generation (timeseries etc)

Analysis and Aggregation Operators:
Highlighting, Grouping, Contextualizing Behaviors

Useful insights are different
current solution is STATIC
Input data

Feature Generation (timeseries etc)

Analysis and Aggregation Operators:
Highlighting, Grouping, Contextualizing Behaviors

Insights

Not all insights are useful
Not all insights are useful
Input data

RULES AND THRESHOLDS

Data changes!

RULES AND THRESHOLDS

Insights

Not all insights are useful.
System no longer works, hours of tuning required due to high data volume!

Not all insights are useful
Domain Expertise + Statistics + Machine Learning + Dataflow Processing
Input data

Feature Generation (timeseries etc)

Analysis and Aggregation Operators:
Highlighting, Grouping, Contextualizing Behaviors

Statistics and Machine Learning

User-Specified Metrics (Domain Expertise)

Dataflow Processing

Domain Expertise

Insights
MacroBase’s Modular Default Analysis Pipeline (MDP)
Input data

Feature Generation (timeseries etc)

Analysis and Aggregation Operators:
Highlighting, Grouping, Contextualizing Behaviors

Insights
Feature Generation (timeseries etc)

Analysis and Aggregation Operators:
Highlighting, Grouping, Contextualizing Behaviors

Input data
- Ingestion via MacroBase JDBC from SQL query
- indicative metrics (battery drain, session length)

Insights
Feature Generation (timeseries etc)

Analysis and Aggregation Operators:
Highlighting, Grouping, Contextualizing Behaviors

Input data

Ingestion via MacroBase JDBC from SQL query
indicative metrics (battery drain, session length)

Domain Expertise - timeseries, Fourier, normalization, etc supported

Insights
Feature Generation (timeseries etc)

Analysis and Aggregation Operators:
Highlighting, Grouping, Contextualizing Behaviors

Ingestion via MacroBase JDBC from SQL query
indicative metrics (battery drain, session length)

Domain Expertise - timeseries, Fourier, normalization, etc supported

Classification and Explanation

Insights
Input data

Feature Generation (timeseries etc)

Analysis and Aggregation Operators:
Highlighting, Grouping, Contextualizing Behaviors

Classification and Explanation

Presentation - ranked list of explanations

Ingestion via MacroBase JDBC from SQL query

Domain Expertise - timeseries, Fourier, normalization, etc supported

Insights

indicative metrics (battery drain, session length)
Additional functionality

• addition of labelled data for supervised learning

• new domain-specific feature transformations

• custom streaming transformation, classification, and explanation operators in alignment with MacroBase’s type system
Streaming Classification Operators (ADR)
aka Outlier Detection!
statistical language

• z-score: number of std. deviations point lies away from sample mean (outlying-ness of point), but sensitive to outliers

• contamination percentage: % of data that is “misbehaving” or bad data

• estimator: best description of the data’s distribution (eg. normal estimator)
robust statistical estimation

finding statistical distributions for data with ill-behaving points
mean absolute deviation (MAD)

median of absolute distance from each point in sample to sample median. median resistant to outliers!
MCD and MAD resilient up to 50%

Figure 3: Discriminative power of estimators under contamination by outliers (high scores better). Robust methods (MCD, MAD) outperform the Z-score-based approach.
But how can we calculate MAD and MCD in a streaming context?
No existing algorithm!
No existing algorithm!
Our answer: Adaptable Damped Reservoir (ADR)
Adaptable Damped Reservoir

- window-based, takes a sample of data
- exponentially weights data based on recency
- **arbitrary** window sizes
- decay over arbitrary decay intervals (time-based or batch-based)
Algorithm 1 ADR: Adaptable Damped Reservoir

given: $k$: reservoir size $\in \mathbb{N}$; $r$: decay rate $\in (0, 1)$

initialization: reservoir $R \leftarrow \{\}$; current weight $c_w \leftarrow 0$

function OBSERVE($x$: point, $w$: weight)

\[ c_w \leftarrow c_w + w \]

if $|R| < k$ then

\[ R \leftarrow R \cup \{x\} \]

else with probability $\frac{k}{c_w}$

remove random element from $R$ and add $x$ to $R$

function DECAY()

\[ c_w \leftarrow r \cdot c_w \]
Streaming Explanation Operator (AMC Sketch)
aka: what attributes differentiate inliers and outliers?
eg. does a specific bluetooth version correlate highly with power failures?
statistical measures used for explanation

- relative risk ratio: relative occurrence of key attributes among different populations
- support: relative occurrence in outliers

MDP finds attribute-metric combinations with high RR and support
Attributes A through H

Outliers

A, C, D
C, D, F
A, F, G
C, D, G
Attributes A through H

Outliers

- A, C, D
- C, D, F
- A, F, G
- C, D, G

Attributes with low *support* in outliers:

B, E, H
Attributes A through H

Outliers

- A, C, D
- C, D, F
- A, F, G
- C, D, G

Attributes with low support in outliers:

- B, E, H

Searches through inliers with attributes A, C, D, F, G, first singly, and then with combinations of attributes, to maximize RR
calculating supports and RR while streaming: Amortized Maintenance Counter (AMC)
Algorithm 3 AMC: Amortized Maintenance Counter

given: $\epsilon \in (0, 1)$; $r$: decay rate $\in (0, 1)$
initialization: $C$ (item $\rightarrow$ count) $\leftarrow \{\} ;$ weight $w_i \leftarrow 0$

function OBSERVE($i$: item, $c$: count)
    $C[i] \leftarrow w_i + c$ if $i \notin C$ else $C[i] + c$

function MAINTAIN()
    remove all but the $\frac{1}{\epsilon}$ largest entries from $C$
    $w_i \leftarrow$ the largest value just removed, or, if none removed, 0

function DECAY()
    decay the value of all entries of $C$ by $r$
    call MAINTAIN()
• Observe: $O(1)$

• Maintain: $O(C \log(1/\epsilon))$ amortized

• Decay: Maintain

• Maintenance performed once sketch reaches upper bound
Experimental Evaluation
MDR is accurate up to 20-30% noise

Figure 4: Precision-recall of explanations. Without noise, MDP exactly identifies misbehaving devices. MDP’s use of risk ratio improves resiliency to both label and measurement noise.
AMC outperforms other heavy hitters sketches

Figure 6: Streaming heavy hitters sketch comparison. AMC: Amortized Maintenance Counter with maintenance every 10K items; SSL: Space Saving List; SSH: Space Saving Hash. All share the same accuracy bound. Varying the AMC maintenance period produced similar results.
ADR produces a 3.2x speedup

Figure 5: ADR provides greater adaptivity compared to tuple-at-a-time reservoir sampling and is more resilient to spikes in data volume (see text for details).
Gaps in the logic, next steps

- contextual outlier detection
- not-normally-distributed data (non-parametric estimation)
- non-categorical attributes