MacroBase: Prioritizing Attention in Fast Data

Jacob Yim
Role: Paper Author
Original paper by Peter Bailis, Edward Gan, Samuel Madden, Deepak Narayanan, Kexin Rong, and Sahaana Suri (SIGMOD 2017)

Problem statement

- Data is being stored in increasingly large capacities
- *Fast data* is automatically generated by machines over time
    - Tends to be especially large in volume
    - e.g. sensor readings, logs from automated processes
- Humans have limited attention that does not grow
- The growth of our data is outpacing our ability to manually inspect it.
- How can we build a system that prioritizes attention by automatically showing users the most important insights about their data?

Motivating use cases

- Mobile applications
  - Cambridge Mobile Telematics uses a mobile app collecting data about drivers
  - MacroBase can help detect bugs on specific devices and firmware
  - CMT actually deployed MacroBase in production, used in evaluation
- Datacenter operations
  - Server outages can be prevented or diagnosed by examining logs
  - Workloads are highly heterogenous, making logs difficult to inspect manually
- Industrial monitoring
  - Sensor readings can be used to identify hazards ahead of time

Related work

- Builds upon other data streaming systems (e.g. Storm, StreamBase, IBM Oracle Streams) and inspired by other specialized streaming systems (Gigascope, MCDB)
    - No prior system for classifying and explaining fast data!
- Inspired by work in statistics and machine learning on outlier detection and data explanation, but must adapt to the domain of fast data streaming

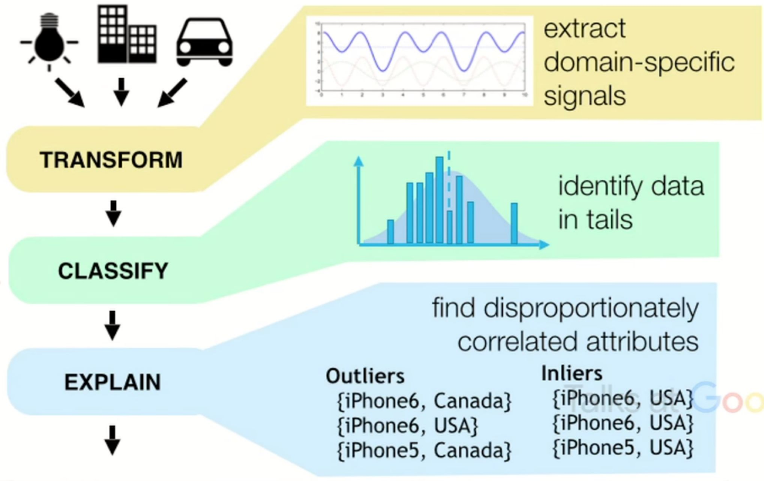
Demo

System architecture

- MacroBase executes queries with *pipelines* of streaming operators
- Fully extensible: users can write their own operators and pipelines
- Three operating modes:
    - GUI allows for interactive exploration
    - One-shot queries can be run programmatically as individual passes
    - Streaming queries can be run over a continuous stream of data

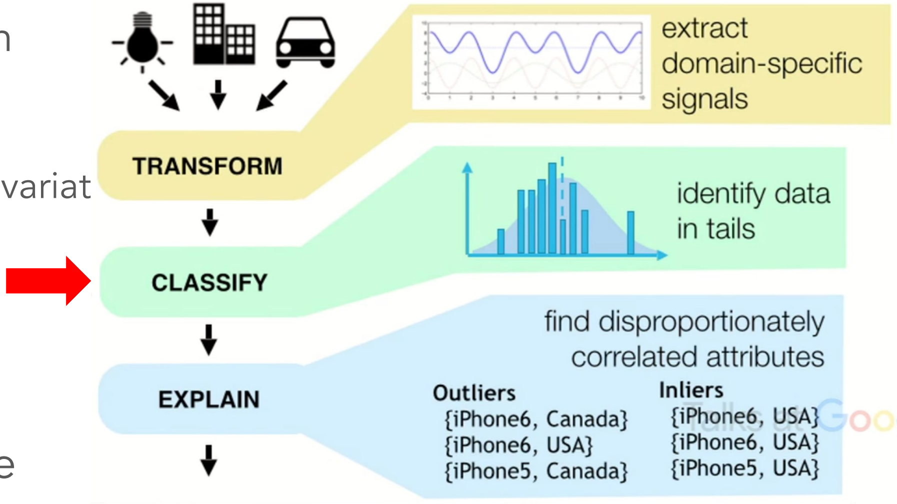
MacroBase pipeline

- Steps in a MacroBase pipeline:
  - Ingestion: data streams ingested from an external source. Data points contain *metrics* and *attributes*
  - Feature Transformation: optional domain-specific data transformations
  - Classification: data points are labeled based on metrics
  - Explanation: labeled data points are aggregated to produce explanations
  - Presentation: explanations are ranked (default by degree of outlier) and displayed to users

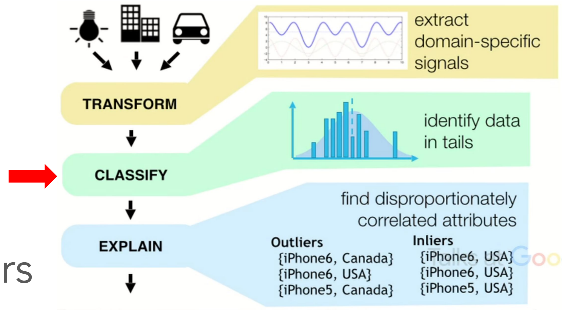


Classification system

- Default classifier labels outliers in a distribution of points (unsupervised density-based classification)
- Z-score (measure of standard deviations from mean) is not robust to outliers
  - Use median absolute deviation (MAD) for univariat data
  - Minimum covariance determinant (MCD) for multivariate
- Classify all points with a score above some percentile as outliers



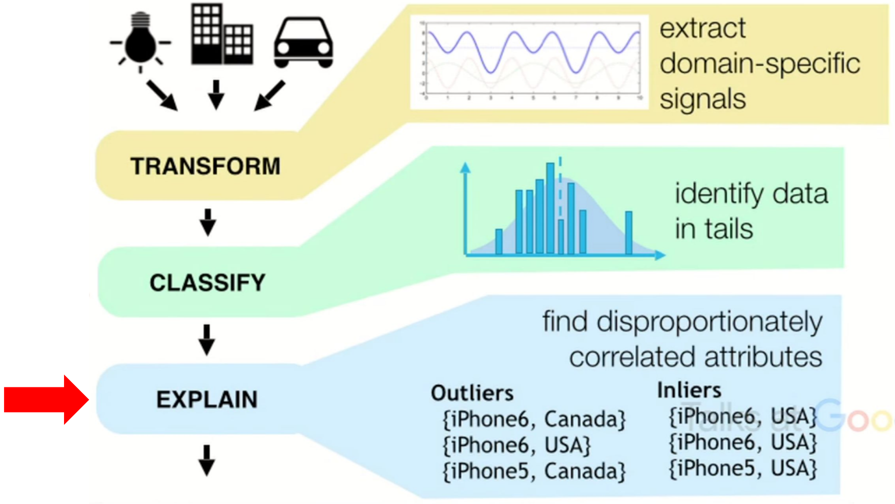
Classification system



- Problem: how do we efficiently update MAD / MCD estimators as the data changes?
- Adaptable Damped Reservoir (ADR)
  - Use a sample of input data exponentially weighted towards more recently added points
  - Reservoir “decays”
  - Unlike existing techniques, in ADR the decay interval is arbitrary
  - Decay can be time-based or based on number of data points
- ADRs used both to sample input data for retraining model and to sample outlier scores for calculating percentiles

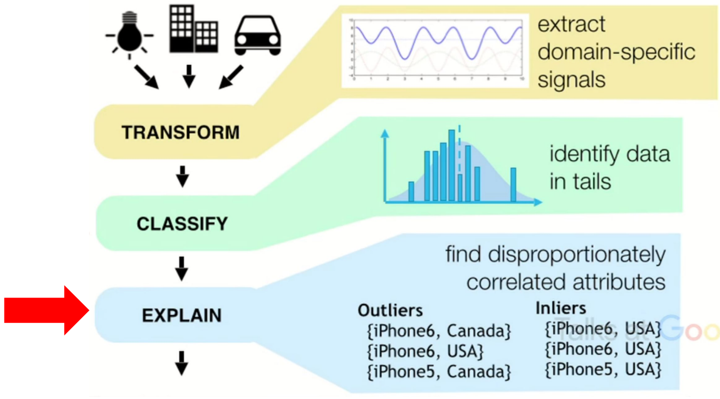
Explanation system

- Goal: find attributes common to outliers but uncommon to inliers
- Find combinations of attribute values with high *risk ratio* $\frac{a_o/(a_o+a_i)}{b_o/(b_o+b_i)}$
    - Where the combination appears a_0 times in outliers and a_i times in inliers, and there are b_0 other outliers and b_i other inliers
    - Quantifies how much more likely a data point of this combination is to be an outlier
- Also find combinations with high *support* (presence in outliers)



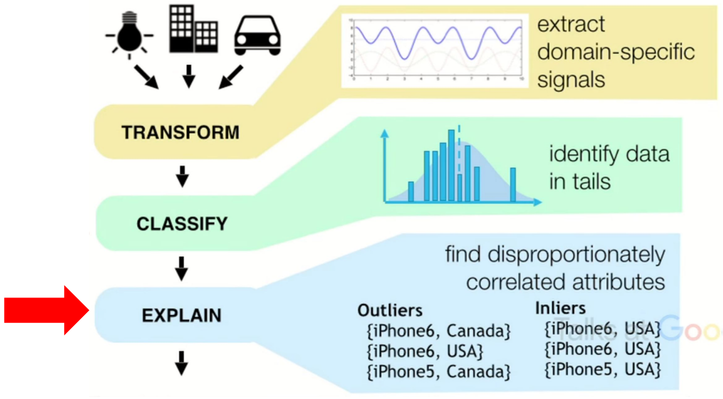
Explanation system

- Naive approach involves iterating over all outliers and inliers
- Optimizations:
  - Since outliers are fewer than inliers, first find combinations of attributes with support in the outliers, then search the inliers for those combinations
  - Compute risk ratios for individual attributes first, then compute support of combinations of attributes with high risk ratio



Explanation system

- To stream explanations, use a heavy-hitters sketch called the Amortized Maintenance Counter (AMC)
    - Maintains attributes with top *k* occurrence in the stream
    - Compared to other sketches, is faster to update at the cost of memory usage
- For tracking combinations of attributes, use a tree data structure



Evaluation

- Evaluated MacroBase on synthetic and real-world data:
  - On a synthetic dataset, MacroBase correctly identifies causes of outliers with <20% noise
  - Using real-world server data to find anomalous hosts, MacroBase has >85% accuracy on all forms of anomalies
- Additional end-to-end testing on real datasets for throughput and # explanations:

Queries					Thru w/o Explain (pts/s)		Thru w/ Explain (pts/s)		# Explanations		Jaccard
Dataset	Name	Metrics	Attrs	Points	One-shot	EWS	One-shot	EWS	One-shot	EWS	Similarity
Liquor	LS	1	1	3.05M	1549.7K	967.6K	1053.3K	966.5K	28	33	0.74
	LC	2	4		385.9K	504.5K	270.3K	500.9K	500	334	0.35
Telecom	TS	1	1	10M	2317.9K	698.5K	360.7K	698.0K	469	1	0.00
	TC	5	2		208.2K	380.9K	178.3K	380.8K	675	1	0.00
Campaign	ES	1	1	10M	2579.0K	778.8K	1784.6K	778.6K	2	2	0.67
	EC	1	5		2426.9K	252.5K	618.5K	252.1K	22	19	0.17
Accidents	AS	1	1	430K	998.1K	786.0K	729.8K	784.3K	2	2	1.00
	AC	3	3		349.9K	417.8K	259.0K	413.4K	25	20	0.55
Disburse	FS	1	1	3.48M	1879.6K	1209.9K	1325.8K	1207.8K	41	38	0.84
	FC	1	6		1843.4K	346.7K	565.3K	344.9K	1710	153	0.05
CMT	MS	1	1	10M	1958.6K	564.7K	354.7K	562.6K	46	53	0.63
	MC	7	6		182.6K	278.3K	147.9K	278.1K	255	98	0.29

- Case studies show applicability in combining supervised and unsupervised classification, time series data transformation, and video stream processing

Future work

- Expand to new domains!
- Designed to be highly flexible for any application involving classifying and explaining fast data, so possibilities are endless