The Importance of *Features* for Statistical Anomaly Detection

David Goldberg
Yinan Shan

Outline

- I had an anomaly detection problem
- I went to the literature, it didn't seem helpful
- So I invented my own algorithm
- I draw some conclusions from this experience

Terminology

- Monitored Signals
  - These are continuously scanned, looking for trouble
- Disruption
  - The site is not working as designed
- Anomaly
  - A monitored signal has unusual behavior, suggesting a disruption
- Alert
  - What we do after detecting an anomaly

Summarizing terminology

- An ***anomaly*** is what we measure. We hope that it correlates with ***disruptions***
- A disruption is a symptom
  – A human still needs to investigate and determine the root cause

Outline

- I had an anomaly detection problem
- I went to the literature, it didn't seem helpful
- So I invented my own algorithm
- I draw some conclusions from this experience

My problem

- Finding ***disruptions*** in eBay search results
- Two types
  - Severe disruptions (e.g. no search results) are detected by other systems
  - We're mainly interested in more subtle disruptions
- Examples
  - Change in average price shown
  - Change in items from sellers with low feedback
  - Change in the category distribution of results

Tool for Finding Disruptions

- We detect via monitoring
  - We regularly monitor search results
- But monitoring returns a lot of data
  - Not clear when to signal an alert

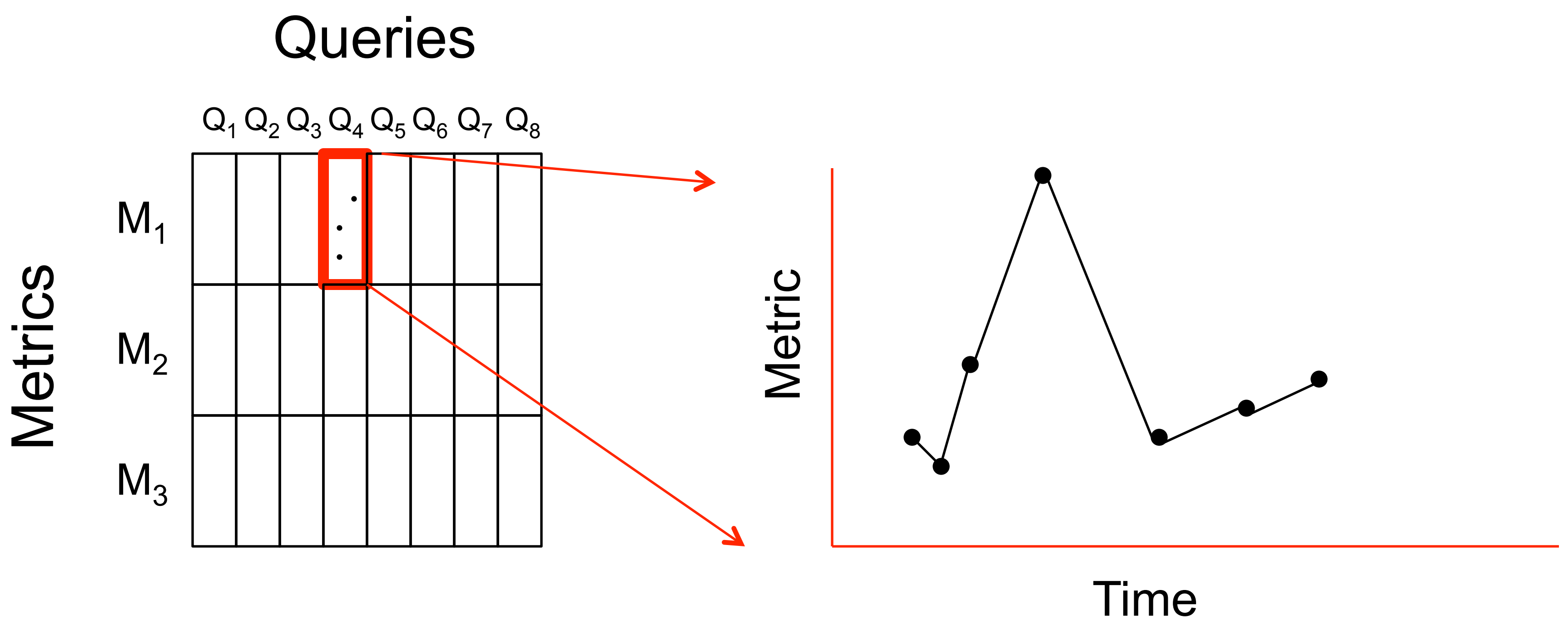
Monitored Signals for Search

- There are 3000 reference queries
- They are repeatedly issued every 4 hours
- Consecutive results for the same query are usually different
  - eBay has auctions, and they expire
  - A lot of 1-off (single quantity) items, and they get sold
  - Ranking for multiple quantity items changes based on popularity
- How much change in monitored data is a disruption?

How to measure change

- We compare two difference instances of the same query
- We measure change using metrics
- Sample metrics:
  - Recall size (number of returned items)
  - Percentage of used items in the result set
  - Median price of items in the result set
- There are about 50 metrics total

3000 Reference Queries, 50 Metrics



3000×50 Time series

When to signal alert

- Original system used rules
- Example
  - If the fraction of auctions metric for at least 10% of queries is 0, signal an alert
- Good rules are very hard to construct
- Rule based system was
  - Brittle
  - Missed disruptions

Outline

- I had an anomaly detection problem
- I went to the literature, it didn't seem helpful
- So I invented my own algorithm
- I draw some conclusions from this experience

The literature

- Seems to focus on two situations
- Points in *n*-space
  - Represent my data a points in space, look for a point that doesn't belong
- Time Series
  - Represent my data as a time series, look for a point that doesn't belong

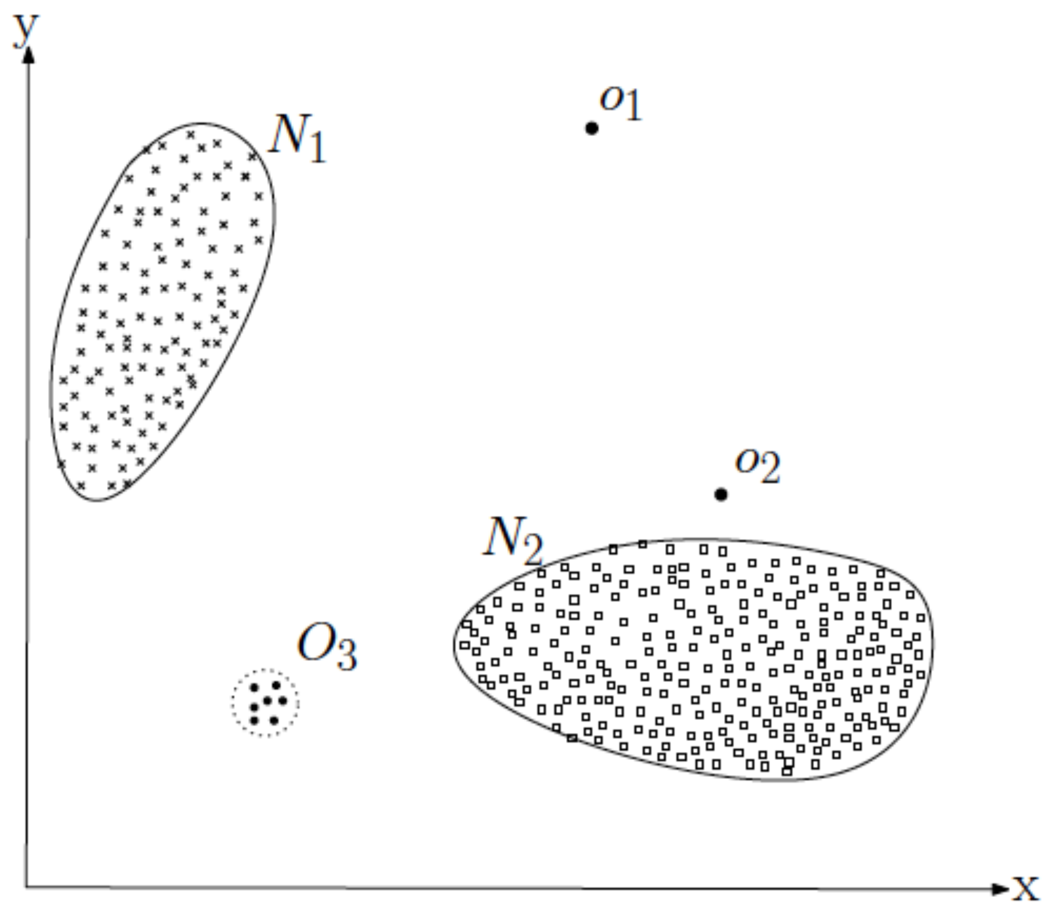


Fig. 1. A simple example of anomalies in a 2-dimensional data set.

From Chandola et al *Anomaly Detection: A Survey*

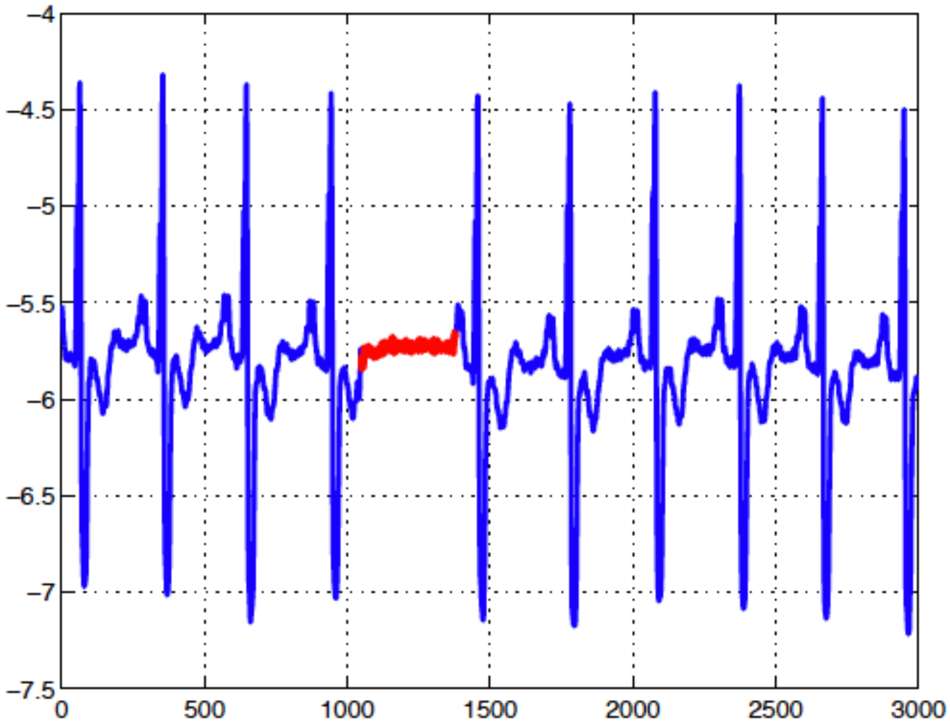


Fig. 4. Collective anomaly corresponding to an *Atrial Premature Contraction* in an human electrocardiogram output.

From Chandola et al *Anomaly Detection: A Survey*

The literature and our problem

- Neither of these approaches seems natural
- My data isn't a time series
  - It's 150,000 time series
- Not clear how to represent monitoring info as a point cloud

Outline

- I had an anomaly detection problem
- I went to the literature, it didn’t seem helpful
- So I invented my own algorithm
- I draw some conclusions from this experience

Algorithm philosophy

- Based on understanding the problem, rather than feeding into an “black box” anomaly detector
- Import advantage: disruptions must be investigated (to get root cause), so want help understanding why the alert was raised

The algorithm idea

- For each time series, define 'surprise' of the last point
  - This is deviation from expected value
  - Can do this using standard time series methods
- But….
  - With so many time series, you expect some very large surprise values

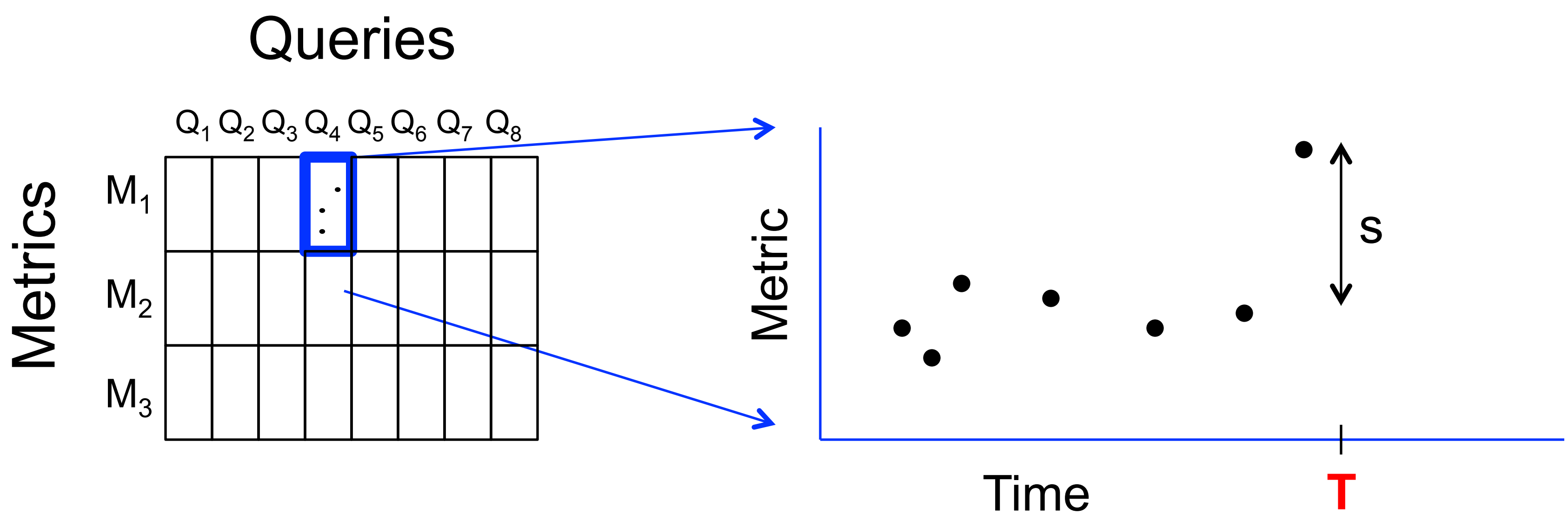
The algorithm idea (2)

- For each time series, define ‘surprise’ of the last point
  - This is deviation from expected value
  - Can do this using standard time series methods
- But….
  - With so many time series, you expect some very large surprise values
- So instead ask how many queries have large surprise
- And if alert if **that** number is surprising

The algorithm, more formally

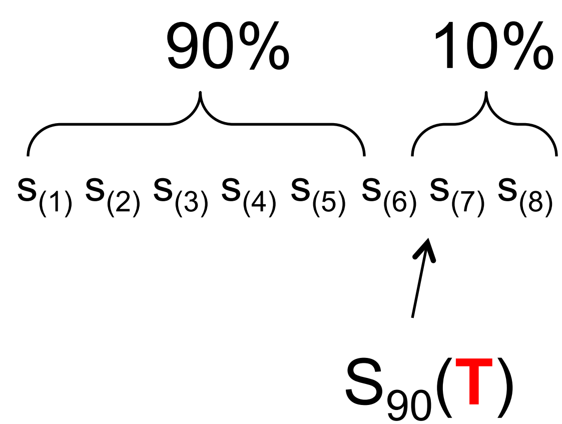
- Surprise is the deviation from a linear fit, normalized by median deviation
  - So there is a surprise for each (query, metric, end-time) triple
- Use this to define a new time series
  - For each fixed metric, compute the 90^{th} percentile of surprise over all queries
  - Now there are 50 time series, instead of 50 x 3000
- Look for outliers in this new time series

Algorithm in Pictures

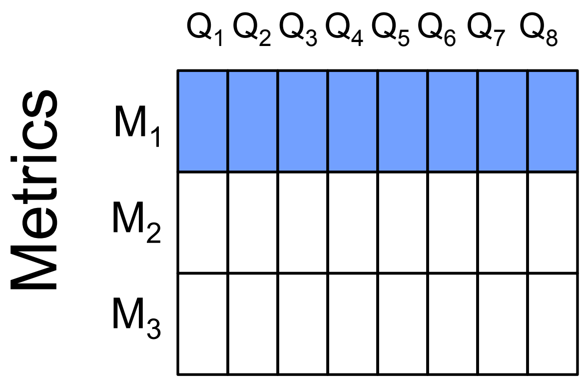


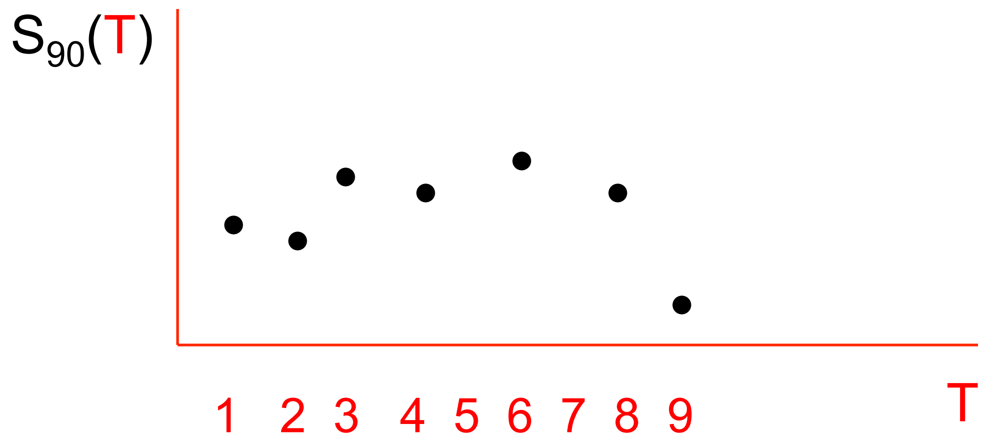
Metrics

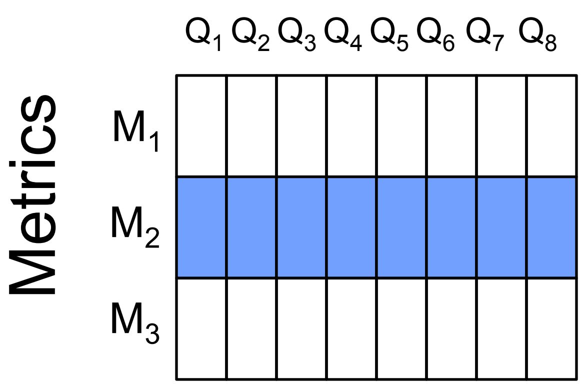
	Q_1	Q_2	Q_3	Q_4	Q_5	Q_6	Q_7	Q_8
M_1	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8
M_2								
M_3								

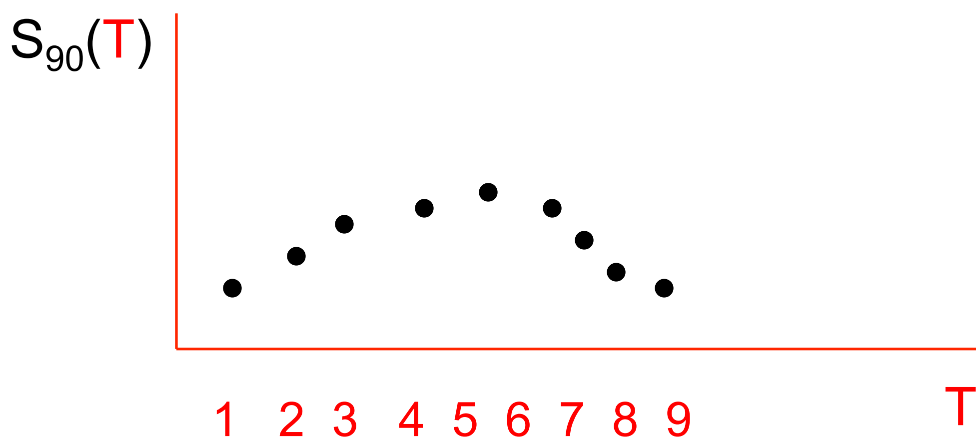


Algorithm in Pictures (2)



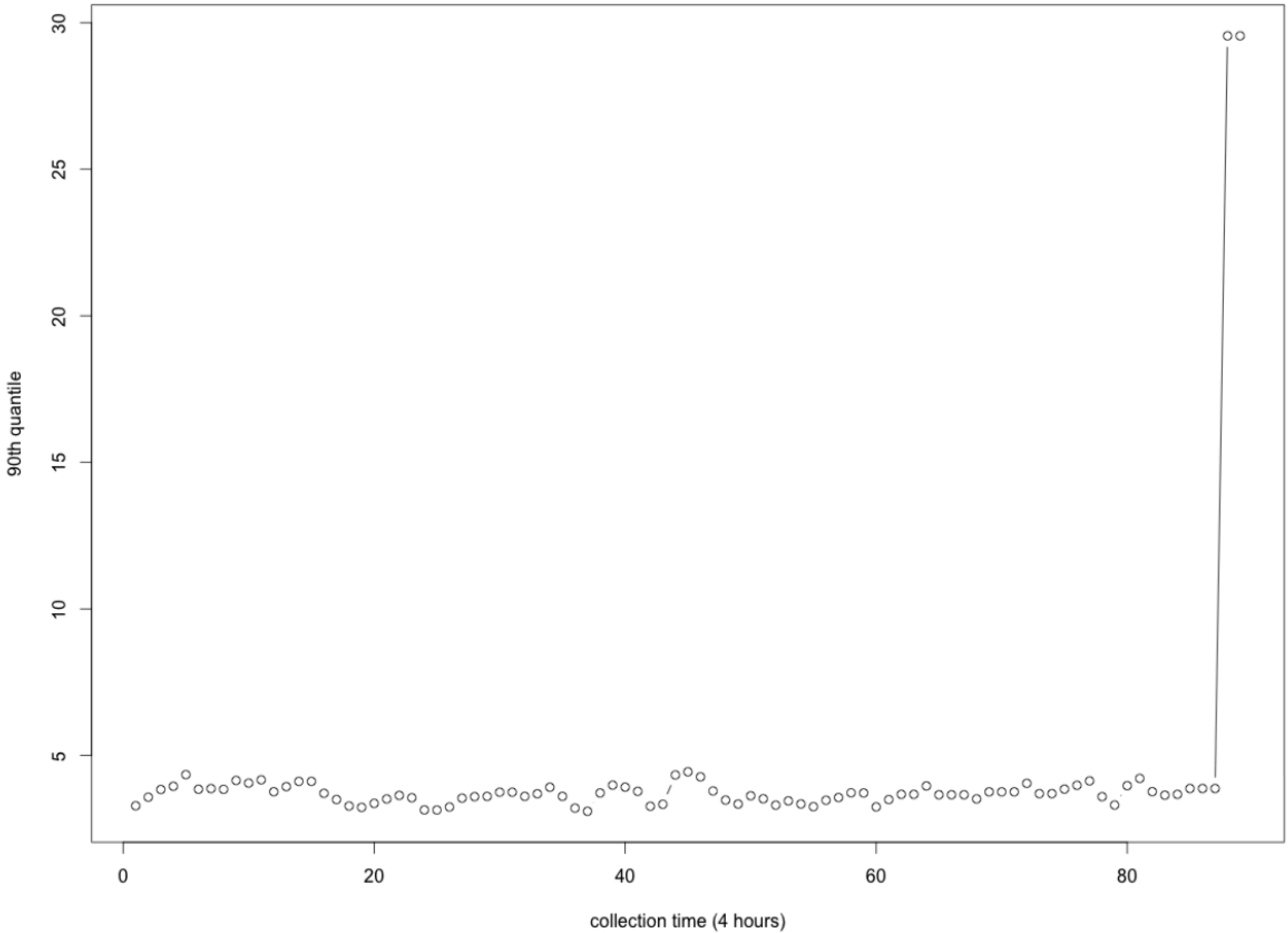






Detecting Anomalies

- Each metric has a time series
- Declare anomaly if one or more of those series has a dramatic ‘blip
  - 3σ from the median


30
25
20
15
10
5
90th quantile
0
20
40
60
80
collection time (4 hours)

Outline

- I had an anomaly detection problem
- I went to the literature, it didn't seem helpful
- So I invented my own algorithm
- I draw some conclusions from this experience

Lessons Learned

- There's a tradeoff of features and statistics
  - Good features may eliminate need for subtle statistical tests
- Doubt existence of general purpose anomaly software
  - Not obvious how to adapt them for our problem
- Conclusion: features are important!
  - Perhaps even more important than statistics

Summary

- Real-life anomaly detection might not easily map to time series or points in *n*-space.
  - Or in general, to black-box statistical tests.
- At least in one example, good features eliminate the need for sophisticated statistics
  - Side effect: features can help with root cause

For Discussion

- Your examples of anomaly detection?
- Other approaches for my data?
- Does trading features for statistics make sense to you?