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*
From Chandola et al *Anomaly Detection: A Survey*

Fig. 4. Collective anomaly corresponding to an *Atrial Premature Contraction* in an human electrocardiogram output.
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 90th 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

Queries

<table>
<thead>
<tr>
<th>M1</th>
<th>M2</th>
<th>M3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q1</td>
<td>Q2</td>
<td>Q3</td>
</tr>
<tr>
<td>Q4</td>
<td>Q5</td>
<td>Q6</td>
</tr>
<tr>
<td>Q7</td>
<td>Q8</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>M1</th>
<th>M2</th>
<th>M3</th>
</tr>
</thead>
<tbody>
<tr>
<td>s1</td>
<td>s2</td>
<td>s3</td>
</tr>
<tr>
<td>s4</td>
<td>s5</td>
<td>s6</td>
</tr>
<tr>
<td>s7</td>
<td>s8</td>
<td></td>
</tr>
</tbody>
</table>

Metric

Time

90% 10%

S_{90}(T)
Algorithm in Pictures (2)

Metrics

\[ Q_1, Q_2, Q_3, Q_4, Q_5, Q_6, Q_7, Q_8 \]

\[ M_1, M_2, M_3 \]

\[ S_{90}(T) \]

\[ 1, 2, 3, 4, 5, 6, 7, 8, 9 \]

\[ T \]
Detecting Anomalies

- Each metric has a time series
- Declare anomaly if one or more of those series has a dramatic ‘blip’
  - $3\sigma$ from the median
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?