Anomaly Detection Applications in Earth Science

Mark Schwabacher¹
Stephen D. Bay²
Ranga Myneni³
Wenze Yang³

¹NASA Ames Research Center - http://ic.arc.nasa.gov/~schwabac/
²Institute for the Study of Learning and Expertise – http://www.isle.org/~sbay/
³Boston University

Symposium on Machine Learning for Anomaly Detection
Stanford University, May 22-23, 2004
Earth Science Application

• NASA has spent large sums of money collecting huge data sets from Earth-observing satellites such as MODIS

• NASA Earth scientists are interested in studying anomalies in the data, but finding all of the interesting anomalies in a very large data set is difficult. Scientifically interesting anomalies in vegetation can be caused by forest fires, logging, disease, or introduction of alien species, for example.

• These data sets sometimes contain errors, which need to be detected and corrected
• Initially we are focusing on the MODIS (Moderate Resolution Imaging Spectroradiometer) and AVHRR (Advanced Very High Resolution Radiometer) Leaf Area Index (LAI) and Fraction absorbed of Photosynthetically Active Radiation (FPAR) products, which are two measures of vegetation obtained from Earth-orbiting satellites.
We have the following LAI and FPAR data:

- from AVHRR: 20 years of monthly global data at 8 km resolution (1.6 million pixels per month)
- from MODIS: 4 years of weekly global data at 1 km resolution (111 million pixels per week)
Challenges

- Data volume
  - We developed a near-linear time algorithm
- Desire to find multi-pixel, multi-timestep anomalies
  - We have some early results using “bump hunting”
  - A direction for future research
Previous Work

• Potter et al. (2003) used deseasonalized AVHRR FPAR at 0.5 degree resolution (95,000 pixels/month)

• They defined an anomaly to be any pixel that was at least 1.7 standard deviations below the long-term average for at least 12 consecutive months
Near-linear time anomaly detection algorithm

I will

– Show that very simple modifications of a basic algorithm leads to very good performance
– Explain why this approach works well
– Discuss limitations of this approach
The main idea is to find points in low density regions of the feature space.

\[ P(x) \approx \frac{k}{NV} \]

- \( V \) is the total volume within radius \( d \)
- \( N \) is the total number of examples
- \( k \) is the number of examples in sphere

Distance measure determines proximity and scaling.
Outlier Definitions

• Outliers are the examples for which there are fewer than \( p \) other examples within distance \( d \)
  – Knorr & Ng
• Outliers are the top \( n \) examples whose distance to the \( k \)th nearest neighbor is greatest
  – Ramaswamy, Rastogi, & Shim
• Outliers are the top \( n \) examples whose average distance to the \( k \) nearest neighbors is greatest
  – Angiulli & Pizzuti, Eskin et al.

These definitions all relate to \( P(x) \equiv \frac{k}{NV} \).
Existing Methods

- **Nested Loops**
  - For each example, find its nearest neighbors with a sequential scan
  - $O(N^2)$

- **Index Trees**
  - For each example, find its nearest neighbors with an index tree
  - Potentially $N \log N$, in practice can be worse than nested loops

- **Partitioning Methods**
  - For each example, find its nearest neighbors given that the examples are stored in bins (e.g., cells, clusters)
    - Cell-based methods potentially $N$, in practice worse than nested loops for more than 5 dimensions (Knorr & Ng)
    - Cluster based methods appear sub-quadratic
Our Algorithm: Orca

- Assume we want to find the top m outliers
- Based on Nested loops
  - For each example, find it’s nearest neighbors with a sequential scan
- Two modifications
  - Randomize order of examples
    - Can be done with a disk-based algorithm in linear time
  - While performing the sequential scan,
    - Keep track of closest neighbors found so far
    - Prune examples once the neighbors found so far indicate that the example cannot be a top outlier
- Process examples in blocks
- Worst case $O(N^2)$ distance computations, $O(N^2/B)$ disk accesses
Pruning

- Outliers based on distance to the 3rd nearest neighbor (k=3)

$d$ is distance to 3rd nearest neighbor for the weakest top outlier
Experimental Setup

- 6 data sets varying from 68K to 5M examples
- Mixture of discrete and continuous features (23-55)
- Wall time reported (CPU + IO)
  - Time does not include randomization
- No special caching of records
- Pentium 4, 1.5 Ghz, 1GB Ram
- Memory footprint ~3MB
- Mined top 30 outliers, k=5, block size = 1000, average distance
Scaling with N

Corel Histogram

KDDCup 1999

Person

Normal 30D
## Scaling Summary

<table>
<thead>
<tr>
<th>Data Set</th>
<th>Slope</th>
</tr>
</thead>
<tbody>
<tr>
<td>Corel Histogram</td>
<td>1.13</td>
</tr>
<tr>
<td>Covertype</td>
<td>1.25</td>
</tr>
<tr>
<td>KDDCup 1999</td>
<td>1.13</td>
</tr>
<tr>
<td>Household 1990</td>
<td>1.32</td>
</tr>
<tr>
<td>Person 1990</td>
<td>1.16</td>
</tr>
<tr>
<td>Normal 30D</td>
<td>1.15</td>
</tr>
</tbody>
</table>

Slope of regression fit relating log time to log N

\[
\log t = \log a + b \log N \quad \text{or} \quad t = aN^b
\]
Scaling with k

Person

Normal 30D

Total Time

K

K

1 million records used for both Person and Normal 30D
Scaling with the number of outliers requested

- Finding the top 1000 outliers out of 7.7 million points took 10 times as long as finding the top 10 outliers
- Appears to be sublinear
Why is it nearly linear?

- Intuitively, because of pruning, most points only have to be compared with a small number of other points to determine that they are not outliers
- For an average-case analysis, see our KDD-2003 paper
Limitations

- Pruning rule fails to achieve near-linear performance when
  - examples not in random order
  - examples not independent
  - no outliers in data
  - very large number of outliers in data (given parameters to algorithm)
LAI/FPAR anomaly detection results

- Used MODIS data from one time point at 4 km resolution (7.7 million pixels)
- Used 4 variables: LAI, FPAR, QA, and latitude
- The #1 outlier was in northern Russia and the #2 outlier was in southern New Zealand
- Both points had unusually high LAI and FPAR values for their latitudes
- Investigation revealed a bug in the software that produced the LAI and FPAR products
- The bug was fixed, we ran the algorithm again, and the scientists said that the new top outliers appeared to be normal
LAI/FPAR anomaly detection results

- Used MODIS data from one time point at 1 km resolution (111 million pixels)
- Ran for 24 days (on 500 MHz Sun)
- Top outliers found were considered to be normal by the scientists
Comparison with a density-based outlier detection method

• We used EM to fit a GMM to a sample of the 4km LAI/FPAR data
• We then found all the points in the full data set that had low density according to the model
• Much faster than Orca
• Outliers are
  – different from Orca’s outliers
  – weakly correlated with Orca’s outliers
  – potentially interesting
Comparison with a density-based outlier detection method

- Sample outliers in LAI/FPAR
  - According to Orca: top outliers had high LAI/FPAR values for their latitudes
  - According to density-based method: top outliers were very far south, where there is little land area
Comparison with GritBot

• Commercial software from Ross Quinlan
• Builds a decision tree for each variable in terms of the other variables
• Memory-based
• Generally slower than Orca
• Outliers are
  – different from Orca’s outliers
  – weakly correlated with Orca’s outliers
  – interesting
• We can’t run it on the 4km LAI/FPAR data
Comparison with GritBot

- Sample outliers in Census household data
  - According to Orca: Top outlier is single-family house in San Diego, 5 married couples, 5 mothers, 6 fathers, value of house is $85K, total income is $86K
  - According to GritBot: 228 records for which the household was listed as rural although another field indicated that it was urban (e.g. city population > 100,000)
Multi-point anomaly detection: Why not cluster the anomalies?

- Large region of weakly anomalous points might be significant
- Explicitly representing time and space can increase statistical power
- But we plan to try it anyway
Multi-point anomaly detection: Bump Hunting

- Bump hunting (Friedman & Fisher, 1999) looks for regions of the space in which the target variable is considerably larger (or smaller) than its average value over the entire space.
- We used deseasonalized 16 km AVHRR LAI monthly data for 19 years (228 timesteps)
- 3-D space (lat, lon, time) with 90 million points
- Used a 40,000-point random sample, with 2/3 for training and 1/3 for test
Bump hunting results

- The algorithm found the following “box”
  latitude > 57.36
  latitude < 59.67
  longitude > 69.94
  longitude < 95.15
  month >= Jan 1982
  month <= Dec 1984

- In this box, LAI is on average 1.09 less than its long-term average

- Box is in Russia in the first 3 years of the data

- Possible explanation: vegetation is increasing over time in this region due to global warming
Future Work

- Algorithms for detecting multi-pixel, multi-timestep anomalies using more than one variable
- Detecting anomalies in the relationship among the variables
- Parallel processing
- Explicitly handling spatiotemporal autocorrelation
Summary & Conclusions

• Our algorithm can find single-point anomalies in nearly linear time, and can scale to large data sets.
• Using single-point anomaly detection, we found an error in an important Earth science data set, which was then fixed.
• Multi-pixel, multi-timestep anomaly detection has the potential to find scientifically interesting anomalies in Earth science data.
• Different outlier detection methods find qualitatively different, but potentially interesting, outliers, so it may be worth running multiple methods.
Resources

- Datasets are public and are available by request