Making sense of unusual suspects - Finding and Characterizing Outliers

Ira Assent
Department of Computer Science
ira@cs.au.dk
Aarhus University, Denmark

joint work with
Barbora Micenková, Xuan-Hong Dang, Raymond T. Ng, Brian McWilliams, Arthur Zimek, Erich Schubert
An outlier

.. deviates from other observations and raises suspicion that it could have been generated by a different mechanism (Hawkins)

search for such deviating objects = outlier detection
Outlier detection / scoring

- Provides set of outliers or ranking of outliers
  - No reasoning
Outlier explanation

Given an outlier, determine *how* it differs from the remainder of the data.

- = find outlier explanatory component / outlying property / outlier context / outlier characteristic..

- Helps domain expert in verifying outliers and understanding how the outlier method works

What is a good explanation?

an example: scanning traffic of a web service, each record contains stats about an IP address per day

<table>
<thead>
<tr>
<th>Requests count</th>
<th>Cookie present rate</th>
<th>Query length</th>
<th>Query entropy</th>
<th>Traffic distr.</th>
<th>ICV</th>
<th>Requests failed rate</th>
<th>Invalid cookies rate</th>
<th>Avg. request size</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>0.7</td>
<td>10.2</td>
<td>0.14</td>
<td>0.5</td>
<td>108</td>
<td>0.1</td>
<td>0.02</td>
<td>4087</td>
</tr>
<tr>
<td>67</td>
<td>0.6</td>
<td>15.1</td>
<td>0.22</td>
<td>0.33</td>
<td>213</td>
<td>0.08</td>
<td>0.01</td>
<td>5001</td>
</tr>
<tr>
<td>13</td>
<td>0.5</td>
<td>21</td>
<td>0.15</td>
<td>0.6</td>
<td>103</td>
<td>0.13</td>
<td>0</td>
<td>6789</td>
</tr>
<tr>
<td>1087</td>
<td>0.4</td>
<td>16.3</td>
<td>0.79</td>
<td>0.4</td>
<td>199</td>
<td>0.09</td>
<td>0.03</td>
<td>6185</td>
</tr>
<tr>
<td>862</td>
<td>0.6</td>
<td>14</td>
<td>0.17</td>
<td>0.71</td>
<td>176</td>
<td>0.11</td>
<td>0.02</td>
<td>5764</td>
</tr>
<tr>
<td>2003</td>
<td>0.5</td>
<td>12.9</td>
<td>0.56</td>
<td>0.5</td>
<td>201</td>
<td>0.07</td>
<td>0.04</td>
<td>6652</td>
</tr>
</tbody>
</table>
Attribute subspace explanation

Example outlier explanation: \{\textit{requests count}; \textit{query entropy}\}

- Properties:
  - semantics of the outlier in terms of original attributes
  - a globally interpretable explanation (usability)
  - each outlier has its own explanatory subspace
Subspace scoring function

Cannot derive explanatory subspace just by analyzing vicinity of the point in full space => need to consider different subspace projections

Goal: for a given outlier \( p \), assess its deviation in each subspace \( S \) and assign a score \( \omega_p(S) \)

The more deviating the subspace, the higher the scores

\[
\omega_p(\{x, z\}) = 1.0 \\
\omega_p(\{x, y\}) = 1.3 \\
\omega_p(\{x, z\}) = 5.1
\]
Issues

- Handle dimensionality bias
  - E.g. scores based on $L_p$ norms are not comparable
- naïve solution has time complexity $\Theta(2^d) \ast \Omega(n)$!

- no monotonicity property for outliers wrt. subspaces
- infeasible to find an optimal solution for practical data set sizes
- we need a fast heuristic
Separability vs. outlierierness

- Approach:
  - use what we call separability as an indication of outlierierness
  - intuitively, outlierierness of a point is related to its separability from the rest of the data
- measure of separability → measure of deviation → subspace scoring function
Measure of separability

• assume that the data follows a distribution $f$
• place a kernel (Gaussian) $g$ centered at outlier $p$
• quantify separability as an inverse of the overlap of functions $f$ and $g$

$$sep(p) = \frac{1}{\int_{-\infty}^{\infty} \min(f(x), g_p(x)) \, dx}$$
From separability to classification

- generate a distribution $g$ of artificial points around $p$
- original data = \textit{inlier} class; outlier + artificial points = \textit{outlier} class
- measure their separability as an error at classification or classification accuracy
Feature selection

• With classification setup:
  • May use standard feature selection methods to find explanatory subspaces

a, b, c, d, e, f

FEATURE SELECTION
subset selection by SVM / shrinkage with lasso / ...

a small subset of the most relevant attributes

b, c
Histogram of explanatory subspaces

- KDD Cup’99 data set
  - Intrusion detection
- Same type of attack is characterized by similar subspaces (few dark cells)
  - E.g. `guess_passwd` dominated by `num_failed_logins`

ODD@KDD 2018, ira@cs.au.dk
Domain expertise

• In some cases, additional information on some outliers is available or can be generated
  • E.g. examples from the past of issues with the data
    • Known network intrusion attacks
  • Domain expert is willing to screen a (small) subset of the data
    • Label some of the network traffic
• How can this information be incorporated?
  • Ideally make use of all information available
Learning outlier ensembles

Scenario:
• majority of data normal, some observations deviate (= outliers)
• we have access to historical data and some labels
• detect and score outliers in new data

Ensembles
• Well established technique in classification
• Also used in clustering and outlier detection
• Combination of diverse learners stabilizes and improves accuracy
• Here: gives access to different outlier models

Supervised

• Generally better detection performance (more information)
• Requires labeled training data
• Typically worse at uncovering new types of outliers

Unsupervised

• Not as good overall detection performance (less information)
• Also in the absence of labeled data
• Typically better at uncovering new types of outliers
Approach

- Combine supervised and unsupervised outlier detection
  - make use of the labels
  - capture also new types of outliers

- Integrate different outlier detection approaches

- Handle class imbalance (supervised detection)
Basic concept

original data

new features

Supervised detection

labels

\[ x_1 \rightarrow x_2 \rightarrow \ldots + \Phi_1(X) \rightarrow \Phi_2(X) \rightarrow \Phi_3(X) \rightarrow \ldots \]
New features

- Output of unsupervised outlier detectors
  - Each detector provides a new feature: scores objects in terms of outlierness
- Combine different types of outlier detectors to capture different deviations and to increase stability of performance
  - Different outlier detection algorithms: kNN-outlier, LOF, ABOD, COP, SOD, ...
  - Different parameter settings,
  - Different distance functions,
  - Different subspaces,
  - ...

- Diversity improves accuracy and gives access to different outlier models
New feature example

kNN-outlier scores as a new feature

\[ \Phi(x_1, x_2) \]
Handling class imbalance

- Re-sampling by bagging
- Sample $m$ bags from the training data such that the proportion of outliers and normal data is the same in each bag
  - Outliers are “used” more often
  - Impure inliers only affect the result in one or few bags
  - Helps stability of the outlier detection
- A model is trained on each bag
- All models are combined in a single ensemble that is used for prediction

- Supervised detection (logistic regression in paper) in transformed space including labels (unlabeled data is labeled as inliers, knowing this is imperfect)
Experiments

- proposed: only transformed features
- proposed+: original and transformed
- base1:orig: no transformed
- base2:sup: no transformed, fully labeled data (more info/unrealistic)
- base3:ens: outlier ensemble Schubert et al. SDM 12]
Detection and description at once

Image from AR face database (A. Martinez and R. Benavente)
LOF scores as example

A subspace approach for high-d

- In high-dimensional data, both detection and description of outliers more challenging
- Projecting to low dimensional subspaces
  - If based on e.g. PCA, obtain mapping based mostly on inliers, not so much on discrimination between inliers and outliers
- In the example, the two outliers share neighbors, but the features that separate them from inliers are different
Graph-based approach

• Construct a global graph over all objects
• Extract neighboring subgraphs to capture local geometrical data structure
• Build dual-objective function for optimization
• Local projection to uncover discriminative features
• Outlier identification + interpretation via eigenspace
Global graph

Construct graph:

- Objects are vertices of the graph
- Each object is connected to $k$ nearest objects
- Edge weights encode similarity using radial symmetric Gaussian kernel
- Yields fully connected graph (all vertices can be reached from each other)
- Intuition: graph captures local neighborhoods

$X = \{x_1, x_2, \ldots, x_N\}$

$G = \{V, E\}$

Affinity matrix $K$
Subgraph and objective function

- Extract $G^{(i)} = \{V^{(i)}, E^{(i)}\}$ from $G$ to capture local structure of $x_i$’s vicinity

- Let $y_p, y_q$ be projections of $x_p, x_q$, we aim to
  - retain the natural local data structure
    -> minimize to ensure quality of outlier explanation (avoid distortion)
  - discriminate $x_i$ from its neighboring objects
    -> maximize to ensure true outlier far away

\[
\text{minimize } \sum_p \sum_q \|y_p - y_q\|^2 K^{(i)}_{(p,q)} \quad (1)
\]
\[
\text{maximize } \sum_p \|y_i - y_p\|^2 K^{(i)}_{(i,p)} \quad (2)
\]
Subspace learning

- Choosing mapping function is important
  - Non-linearity reduces explanatory power

- Explore linear mapping as matrix $W$
  - Rewrite as two matrix optimization problems, combine by converting to sparse matrix form
    $$
    W^* = \arg \max_W \left\{ \text{tr} \left( W^T X^{(i)} L^{(i'')} X^{(i)T} W \right) - \alpha W^T W \right\}
    $$
    subject to
    $$
    W^T X^{(i)} D^{(i)} X^{(i)T} W = I
    $$
    and
    $$
    w_p^T w_q = 0 \quad \text{for} \ p \neq q
    $$
  - Constraints ensure non-redundant solutions and regularize (L2 norm)
  - $X^{(i)}$ has $k$ nn of $x_i$ as column vectors; $L^{(i'')}$, difference of Laplacians of equations, tr(.) trace

- Solution requires some algebra:
  - Decompose $X^{(i)}$ to singular values/vectors, transform variables to ensure stability, derive generalized eigenvalue problem

Please see paper for details
Outlier analysis

• What is an outlier?
  - Computer outlier score as statistical distance to neighbors in transformed space
    \[ OS(x_i) = \frac{1}{d} \sum_{p=1}^{d} \max \left\{ \left( \frac{1}{k} \sum w_p^T x(i) \right)^2, \sigma_p^2 \right\} \]

• What describes an outlier?
  - Leading eigenvector in transformed space
  - Top original features from coefficients in eigenvector
  - Due to L2 regularization, discriminative features are those corresponding to largest absolute coefficients
Experiments

- Outliers from pendigit dataset:
  - 1602 objects, 8 (x,y)-positions
  - All digits 1 and 5 inliers
  - 2 of each of the remaining 8 digits outliers

- Discrimination in writing styles.
  - E.g., 7,4 are closest to 1’s distribution
  - 6,9 are closest to 5’s distribution
Experiments

- CMU images: 32 per person, 20 people
  - combination of facial expression, head position, eyes
  - treat sunglass as outlier

![CAM images: 32 per person, 20 people](image)

- LOGP top outliers with leading eigenvectors and discriminative features; plus highly ranked inlier

- SOD top outliers with subspace features

ODD@KDD 2018, ira@cs.au.dk
Discussion points

- **Explanations crucial for outlier detection in practice**
  - In typical data set sizes and dimensionality, reporting or ranking of outliers alone not very useful
  - Explanatory subspaces useful in narrowing the verification and validation to fewer attributes
  - Requires a reduction in the number of objects to compare to
    - Clustering comes in handy
    - Approaches using reference sets
      - similar to our samples of the neighborhood / the inlier class, but agnostic to groups / patterns
  - Observation: semantic gap between these explanations and the verification and validation by the domain expert / data analyst
Research challenges

• How do domain experts / data analysts verify and validate outliers?
  • Background information
    • A priori models
  • Domain knowledge
    • Assumptions for data / inliers / outliers
  • Relative comparison
    • Reference sets → structures? Semantics?
  • What-if-analysis
    • What kind of change would turn an outlier into an inlier?
    • Often more than one explanation!
  • Lineage
    • How was the data generated / processed?
• Describe outliers in relation to inliers
• Semi-supervised models allow incorporation of human input
Conclusion

• Making outlier analysis available for domain experts
  • Provide subspace explanations
    • Make it possible to easily verify outliers
• Providing transparency and interpretation
  • Validation / verification
  • Right to explanation → EU regulation!
  • Avoiding feedback loops, building trust
  • Challenging models
    • Providing active feedback
    • Learning from feedback
      • Both for detection and description!

Thank you for your attention!