Nearest-Neighbor and Clustering based Anomaly Detection Algorithms for RapidMiner

Mennatallah Amer¹, Markus Goldstein²

¹ German University in Cairo, Egypt
² German Research Center for Artificial Intelligence (DFKI)

http://www.dfki.de

August 29th, 2012
Outline

- Introduction to Anomaly Detection
  - Scenarios
  - Global vs local
- Nearest-neighbor based algorithms
  - Global k-NN
  - Local Outlier Factor (LOF) and derivatives
- Clustering based algorithms
  - CBLOF and LDCOF
- RapidMiner Extension
  - Duplicate handling
  - Parallelization
- Experiments
- Conclusion/ Outlook
Introduction

An outlying observation, or **outlier**, is one that appears to deviate markedly from other members of the sample in which it occurs.  

(Grubbs, 1969)

- **Basic anomaly detection assumptions**
  - Outliers are very rare compared to normal data
  - Outliers are “different” w.r.t. their feature values

- **Synonyms**
  - Anomaly detection, outlier detection, fraud detection, misuse detection, intrusion detection, exceptions, surprises, ...
Applications

- Intrusion detection (network and host based)
  - Intrusion detection systems (IDS)
  - Behavioral analysis in anti virus appliances
- Fraud-/ misuse detection
  - Credit cards/ Internet payments/ transactional data
  - Telecommunication data
- Medical sector
- Image processing/ surveillance
- Complex systems
Introduction

Data cleansing application focus:
- Remove outliers for getting better models
- RapidMiner operators
  - Detect Outlier (Distances/ Densities) with binary outlier label as output
  - Class Outlier Factor (COF) uses class labels for finding class exceptions

Anomaly Detection application focus:
- Interested in the outliers, not in the normal data
- Scoring the examples is essential (ranking)
- RapidMiner operators
  - Local Outlier Factor (LOF), but limited implementation
  - DB-Scan clustering with a “noise” cluster (binary label)
Anomaly detection scenarios

- Algorithm output (binary labels vs. scoring)
- Trainings-/ test set availability
  - **Supervised anomaly detection**
    
    ![Supervised anomaly detection diagram]
    
    Traditional classification problem
  
  - **Semi-supervised anomaly detection**
    
    ![Semi-supervised anomaly detection diagram]
    
    Model of normal data only
  
  - **Unsupervised anomaly detection**
    
    ![Unsupervised anomaly detection diagram]
Anomaly detection scenarios (cont'd)

- Local vs. global anomalies

- $p_1, p_2$: global anomalies
- $p_3$: normal instance
- $p_4$: local anomaly
- $c_3$: microcluster
Outline

▶ Introduction to Anomaly Detection
  ▪ Scenarios
  ▪ Global vs local

▶ Nearest-neighbor based algorithms
  ▪ Global k-NN
  ▪ Local Outlier Factor (LOF) and derivatives

▶ Clustering based algorithms
  ▪ CBLOF and LDCOF

▶ RapidMiner Extension
  ▪ Duplicate handling
  ▪ Parallelization

▶ Experiments

▶ Conclusion/ Outlook
Nearest-neighbor based AD

- **k-NN Global Anomaly Score**
  - Score is the distance to the k-th neighbor
  - Score is the average distance of k neighbors

![k-NN Global Anomaly Score](image)

- k
- 10
- use k-th neighbor distance only (no average)
- measure types: MixedMeasures
- mixed measure: MixedEuclideanDistance
- parallelize evaluation process
LOF: Local Outlier Factor

- Most prominent AD algorithm by Breunig et al. 2000
- Is able to find local anomalies

1. Find the k-nearest-neighbors

2. For each instance, compute the local density

   \[ LRD_{\text{min}}(p) = \frac{1}{\left( \sum_{o \in N_{\text{min}}(p)} \frac{\text{reach} \_ \text{dist}_{\text{min}}(p, o)}{|N_{\text{min}}(p)|} \right)} \]

3. For each instance compute the ratio of local densities

   \[ LOF_{\text{min}}(p) = \frac{\sum_{o \in N_{\text{min}}(p)} \frac{LRD_{\text{min}}(o)}{LRD_{\text{min}}(p)}}{|N_{\text{min}}(p)|} \]
LOF: Local Outlier Factor (cont'd)

- Normal examples have scores close to 1.0
- Anomalies have scores > (1.2 ... 2.0)
- Parameter k needs to be chosen (microclusters)
- Only works if you want to detect local anomalies
- Effort is $O(n^2)$
Based on LOF, other algorithms exist

- Connectivity-based outlier factor (COF)
  Estimates densities by shortest-path of neighbors

- Local Outlier Probability (LoOP)
  Uses normal distribution for density estimation

- Influenced Outlierness (INFLO)
  For “connected” clusters with varying densities

- Local correlation Integral (LOCI)
  Grows the r-neighborhood from k to a maximum.
  Computational effort $O(n^3)$, space requirement $O(n^2)$
Outline

▶ Introduction to Anomaly Detection
  ▪ Scenarios
  ▪ Global vs local
▶ Nearest-neighbor based algorithms
  ▪ Global k-NN
  ▪ Local Outlier Factor (LOF) and derivatives
▶ Clustering based algorithms
  ▪ CBLOF and LDCOF
▶ RapidMiner Extension
  ▪ Duplicate handling
  ▪ Parallelization
▶ Experiments
▶ Conclusion/ Outlook
Clustering based AD

Idea

- Cluster the data set, e.g. using *k-means*
- Use the distance from the data instance to the centroid as anomaly score

Cluster-based local outlier factor (CBLOF)

- Cluster data using k-means
- Separate into large (LC) and small clusters (SC) using 2 parameters
- Compute score:

\[
CBLOF(p) = \begin{cases} 
|C_i| \cdot \min(d(p, C_j)) & \text{if } C_i \in SC \text{ where } p \in C_i \text{ and } C_j \in LC \\
|C_i| \cdot d(p, C_i) & \text{if } C_i \in LC \text{ where } p \in C_i 
\end{cases}
\]
CBLOF (cont'd)

- In fact, method is not local (different densities not taken into account)
- Weighting with the cluster size might be a problem
CBLOF (cont'd)

- An “unweighted” CBLOF works better on real data
- Implemented weighting as option of the operator

Local density cluster-based outlier factor (LDCOF)

- Our approach is a real *local* approach
- Density of a cluster is estimated by an average distance to centroid
- Only one parameter for small/large cluster separation
- Score is easily interpretable (score of 1.0 means normal)
LDCOF (cont'd)

\[
distance_{avg}(C) = \frac{\sum_{i \in C} d(i, C)}{|C|}
\]

\[
LDCOF(p) = \begin{cases} 
\frac{\min(d(p,C_j))}{distance_{avg}(C_j)} & \text{if } p \in C_i \in SC \text{ where } C_j \in LC \\
\frac{d(p,C_i)}{distance_{avg}(C_i)} & \text{if } p \in C_i \in LC
\end{cases}
\]

- Flexible operator for CBLOF and LDCOF to work with any clustering algorithm with centroid cluster model output

- Important question: What is the number of clusters \(k\)?
Introduction to Anomaly Detection
- Scenarios
- Global vs local

Nearest-neighbor based algorithms
- Global k-NN
- Local Outlier Factor (LOF) and derivatives

Clustering based algorithms
- CBLOF and LDCOF

RapidMiner Extension
- Duplicate handling
- Parallelization

Experiments

Conclusion/ Outlook
RapidMiner Anomaly Detection Extension

- Available at RapidMiner Marketplace Beta
- Currently most downloaded extension
- Open source

- More information: http://madm.dfki.de/rapidminer/anomalydetection

- 10 different **unsupervised** anomaly detection operators

- **Anomaly Detection (10)**
  - Nearest Neighbor Based (7)
    - k-NN Global Anomaly Score
    - Local Outlier Factor (LOF)
    - Connectivity-Based Outlier Factor (COF)
    - Local Correlation Integral (LOCI)
    - approximate Local Correlation Integral (aLOCI)
    - Local Outlier Probability (LOOP)
    - Influenced Outlierness (INFLO)
  - Clustering Based (2)
    - Local Density Cluster-Based Outlier Factor (LDCOF)
    - Cluster-Based Local Outlier Factor (CBLOF)
  - Statistical Based (1)
    - Histogram-based Outlier Score (HBOS)
Duplicate Handling

- Local nearest-neighbor approaches need attention on duplicates
- If #duplicates > k, density estimation is infinite
- Solution: use k different examples to estimate the density
- For faster computation, filter out duplicates first and assign same outlier score after the algorithm
- Keep amount of duplicate examples (weight) for other algorithms (e.g. LDCOF)
Parallelization for nearest-neighbor based algorithms

- Searching the nearest neighbors is $O(n^2)$
- Taking symmetry into account we need at least $n \cdot (n-1)/2$ distance computations
- Each distance computation depends on the number of dimensions $d$
- Only the $k$ nearest-neighbors are kept in memory for each individual example
- Parallelization needs synchronization for computing $n \cdot (n-1)/2$ distances or all $n^2$ distances are computed without synchronization
- Synchronized blocks are used in Java (Reentrant Lock was slower)
Parallelization for nearest-neighbor based algorithms (cont'd)

- If synchronization should be used depends on the number of dimensions (computation time vs. waiting time and overhead)

- Threshold of 32 used in the extension as decision boundary, but depends on ordering and number of threads
Outline

▶ Introduction to Anomaly Detection
  ▪ Scenarios
  ▪ Global vs local
▶ Nearest-neighbor based algorithms
  ▪ Global k-NN
  ▪ Local Outlier Factor (LOF) and derivatives
▶ Clustering based algorithms
  ▪ CBLOF and LDCOF
▶ RapidMiner Extension
  ▪ Duplicate handling
  ▪ Parallelization
▶ Experiments
▶ Conclusion/ Outlook
Experiments

Evaluation on UCI standard data sets

- **Breast Cancer Wisconsin (Diagnostic)**
  - Features from medical image data
  - 367 examples, 30 dimensions, 10 anomalies (cancer)

- **Pen-based Recognition of Handwritten Text (local)**
  - Features from handwritten digits of 45 different writers
  - 6724 examples, 16 dimensions, 10 anomalies (digit “4”)

- **Pen-based Recognition of Handwritten Text (global)**
  - 809 examples, 16 dimensions, 10 anomalies
  - Only digit “8” is normal
Evaluation on UCI standard data sets

- Receiver operator characteristic (ROC) is computed by varying the outlier threshold.

- Area under curve (AUC) is computed using the ROC.
  - AUC = 1.0: perfect anomaly detection
  - AUC = 0.5: guessing if anomaly or normal

- Optimized parameters
  - k for nearest-neighbor based methods
  - \( \alpha \) for clustering based methods (small/ large cluster threshold)
Breast cancer results (nearest-neighbor based)

INFLO and LOF perform best
Except for COF, all nearest-neighbor algorithms perform well
In a global anomaly detection problem, local NN methods fail
Breast-cancer results (clustering based)

▶ The original CBLOF performs poor
Pen-global results (clustering based)

▶ unweighted-CBLOF/ LDCOF work well on a global task
Experiments

Best algorithms with optimized parameters

<table>
<thead>
<tr>
<th>Data set</th>
<th>k-NN</th>
<th>LOF</th>
<th>COF</th>
<th>INFLO</th>
<th>LoOP</th>
<th>LOCI</th>
<th>CBLOF</th>
<th>u-CBLOF</th>
<th>LDCOF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Breast-cancer</td>
<td>.9826</td>
<td>.9916</td>
<td>.9888</td>
<td>.9922</td>
<td>.9882</td>
<td>.9678</td>
<td>.8389</td>
<td>.9743</td>
<td>.9804</td>
</tr>
<tr>
<td>Pen-local</td>
<td>.9852</td>
<td><strong>.9878</strong></td>
<td>.9688</td>
<td>.9875</td>
<td>.9864</td>
<td>-</td>
<td>.7007</td>
<td>.9767</td>
<td>.9617</td>
</tr>
<tr>
<td>Pen-global</td>
<td>.9892</td>
<td>.8864</td>
<td>.9586</td>
<td>.8213</td>
<td>.8492</td>
<td>.8868</td>
<td>.6808</td>
<td><strong>.9923</strong></td>
<td>.9897</td>
</tr>
</tbody>
</table>

- CBLOF performs poor in general
- LOF performs well on local AD problems
- k-NN performs best on average, u-CBLOF 2\textsuperscript{nd} best
Outline

▶ Introduction to Anomaly Detection
  ▪ Scenarios
  ▪ Global vs local
▶ Nearest-neighbor based algorithms
  ▪ Global k-NN
  ▪ Local Outlier Factor (LOF) and derivatives
▶ Clustering based algorithms
  ▪ CBLOF and LDCOF
▶ RapidMiner Extension
  ▪ Duplicate handling
  ▪ Parallelization
▶ Experiments
▶ Conclusion/ Outlook
New findings

- Local methods fail on global anomaly detection tasks
- LOCI is too slow for real-world data
- u-CBLOF/LDCOF are fast alternatives for nearest-neighbor based methods
- In clustering-based methods, $k$ should be overestimated
Conclusion

Outlook

► Further development of the extension
  ▪ aLOCI implemented
  ▪ Histogram-based outlier score (HBOS) implemented

► Currently working on
  ▪ Operator generating ROCs/ AUCs
  ▪ Clustering-based operator with multivariate Gaussian density estimator

► Future plans
  ▪ SVM-based unsupervised anomaly detection
  ▪ Integrate semi-supervised algorithms
Thank you for your attention!

Questions?