DCSO: Dynamic Combination of Detector Scores for Outlier Ensembles

Yue Zhao  
Department of Computer Science  
University of Toronto

Maciej K. Hryniewicki  
Data Assurance & Analytics  
PricewaterhouseCoopers
Outlier Ensembles

Outlier ensembles combine the results (scores) of either independent or dependent outlier detectors [1].

Bagging (parallel learning) [2, 3]  
Boosting (sequential learning) [4, 5]  
Stacking [6, 7]
Advantages of Outlier Ensembles

- **Improved stability:** robust to uncertainties in complex data, e.g. high-dimensional data

- **Enhanced detection quality:** capable of leveraging the advantages of underlying models

Besides, practitioners usually feel more **confident** to use an ensemble framework with a group base detectors, than a single model.
Challenges in Outlier Ensembles

The ground truth (label), whether a data object is abnormal, is always absent.

Most unsupervised outlier ensembles are therefore parallel combination.

Examples of Parallel Detector Combination
Limitations in Parallel Outlier Score Combination

- **Static process**: the process to measure detector competency is missing
- **Global assumption**: the importance of the data locality is underestimated
- **Limited interpretability**: the explicability of is undermined during combination

Static & Global Combination (SG): conducted *statically* on the *global* scale with all data objects considered, resulting in limited *performance* and *interpretability*.
Research Objectives

Design an *unsupervised* combination framework to *select performing detectors* with a focus on *the local region*, for improved performance and interpretability.

**DCSO**: *Dynamic Combination of Detector Scores for Outlier Ensembles*
Dynamic Classifier Selection (DCS)

DCS is a well-established ensemble framework, which selects the best classifier for each test instance on the fly by evaluating base classifiers’ competency on the local region of the test instance.
## From DCS → DCSO

<table>
<thead>
<tr>
<th>DCS (Supervised Classification)</th>
<th>DCSO (Unsupervised Outlier Mining) (Imbalanced Data: outliers &lt;&lt; inliers)</th>
</tr>
</thead>
<tbody>
<tr>
<td>The ground truth <strong>exists</strong></td>
<td>The ground truth is <strong>missing</strong> Generate <strong>pseudo ground truth</strong> instead</td>
</tr>
<tr>
<td>Evaluate by <strong>accuracy</strong></td>
<td>Evaluate the detector competency by its similarity to the pseudo ground truth</td>
</tr>
</tbody>
</table>
DCSO Demonstration

Different from DCS, DCSO has an additional process to generate pseudo labels, and different competency evaluation approach.
SG (left) vs. DCSO (mid) & Key Difference (right)

Static Global Combination (SG)

\[ C_1, C_2, \ldots, C_d \]

- Global assumption
- Averaging or Maximization of all without selection
- Test Outlier Score

Dynamic Outlier Detector Combination (DCSO)

Dynamic Detector Selection

\[ C_1, C_2, \ldots, C_d \]

- Define the local region and pseudo target (target_k)
- Finding the most promising one by local competency
- \( C^* \)
- Test Outlier Score

Dynamic Ensemble Selection

\[ C_1, C_2, \ldots, C_d \]

- Define the local region and pseudo target (target_k)
- Finding s most promising detectors by local competency
- \( C_1^* \ldots C_s^* \)
- Second-phase Combination: Averaging or Maximization
- Test Outlier Score

Define the local region and pseudo target (target_k)

Finding the most promising one by local competency

Intro ➔ Proposal ➔ R&D ➔ Conclusions
Results & Discussions – Overall Performance

- DCSO frameworks outperform on 8 out of 10 datasets for both *ROC* and *Precision @ Rank N*

- In generally, DCSO brings consistent improvement over baselines, and significant enhancement on *Cardio* (25.33%) and *Pendigits* (31.44%).

<table>
<thead>
<tr>
<th>Dataset</th>
<th>SG_A</th>
<th>SG_M</th>
<th>SG_WA</th>
<th>SG_THRESH</th>
<th>SG_AOM</th>
<th>SG_MOA</th>
<th>DCSO_A</th>
<th>DCSO_M</th>
<th>DCSO_MOA</th>
<th>DCSO_AOM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pima</td>
<td>0.5100</td>
<td>0.4683</td>
<td>0.5127</td>
<td>0.4933</td>
<td>0.4957</td>
<td>0.5039</td>
<td>0.5175</td>
<td>0.4576</td>
<td>0.5083</td>
<td>0.4576*</td>
</tr>
<tr>
<td>Vowels</td>
<td>0.3074</td>
<td>0.3250</td>
<td>0.3029*</td>
<td>0.3074</td>
<td>0.3302</td>
<td>0.3185</td>
<td>0.3682</td>
<td>0.3044</td>
<td>0.3395</td>
<td>0.3161</td>
</tr>
<tr>
<td>Letter</td>
<td>0.2508</td>
<td>0.3547</td>
<td>0.2469</td>
<td>0.2508</td>
<td>0.2950</td>
<td>0.2699</td>
<td>0.2426*</td>
<td>0.3795</td>
<td>0.2862</td>
<td>0.3785</td>
</tr>
<tr>
<td>Cardio</td>
<td>0.3601</td>
<td>0.3733</td>
<td>0.3624</td>
<td>0.3728</td>
<td>0.4233</td>
<td>0.4104</td>
<td>0.3553</td>
<td>0.3676</td>
<td>0.4453</td>
<td>0.3201*</td>
</tr>
<tr>
<td>Thyroid</td>
<td>0.3936</td>
<td>0.2589</td>
<td>0.4061</td>
<td>0.3968</td>
<td>0.3731</td>
<td>0.3896</td>
<td>0.4182</td>
<td>0.2080*</td>
<td>0.3730</td>
<td>0.2449</td>
</tr>
<tr>
<td>Satellite</td>
<td>0.4301*</td>
<td>0.4500</td>
<td>0.4306</td>
<td>0.4466</td>
<td>0.4480</td>
<td>0.4414</td>
<td>0.4400</td>
<td>0.4427</td>
<td>0.4509</td>
<td>0.4398</td>
</tr>
<tr>
<td>Pendigits</td>
<td>0.0733</td>
<td>0.0590</td>
<td>0.0709</td>
<td>0.0700</td>
<td>0.0637</td>
<td>0.0617</td>
<td>0.0749</td>
<td>0.0595</td>
<td>0.0811</td>
<td>0.0560*</td>
</tr>
<tr>
<td>Annthyroid</td>
<td>0.2943</td>
<td>0.2951</td>
<td>0.2975</td>
<td>0.2997</td>
<td>0.3215</td>
<td>0.3103</td>
<td>0.3065</td>
<td>0.2904*</td>
<td>0.3075</td>
<td>0.3046</td>
</tr>
<tr>
<td>Mnist</td>
<td>0.3936</td>
<td>0.3737</td>
<td>0.3944</td>
<td>0.3956</td>
<td>0.3966</td>
<td>0.3976</td>
<td>0.3973</td>
<td>0.3541</td>
<td>0.4123</td>
<td>0.3520*</td>
</tr>
<tr>
<td>Shuttle</td>
<td>0.1508</td>
<td>0.1484</td>
<td>0.1434</td>
<td>0.1582</td>
<td>0.1591</td>
<td>0.1600</td>
<td>0.1589</td>
<td>0.1389*</td>
<td>0.1604</td>
<td>0.1393</td>
</tr>
</tbody>
</table>
Results & Discussions – When DCSO Works?

DCSO works especially well when data forms local clusters.

Visualization by t-distributed stochastic neighbor embedding (TSNE)
Conclusion

DCSO is an ensemble framework to select outperforming base detectors for each test instance on its local region.

Advantages:

1. Outperform on most of the benchmark datasets with improved detection quality
2. Easy to use and robust to underlying assumptions
3. Better interpretability to show how the prediction is made individually
Code & Outlier Detection Toolbox

DCSO code is openly shared at: https://github.com/yzhao062/DCSO

PyOD: a comprehensive Python outlier detection toolbox: https://github.com/yzhao062/Pyod

Google: “Python” + “Outlier Detection”
DCSO: Dynamic Combination of Detector Scores for Outlier Ensembles

Yue Zhao
Department of Computer Science
University of Toronto

Maciej K. Hryniewicki
Data Assurance & Analytics
PricewaterhouseCoopers
Limitations & Future Directions

Limitation: high time complexity using kNN for defining the local region
Future direction: define the local region by clustering instead

Limitation: pseudo generation methods are not accurate
Future direction: involve more advanced generation methods

Limitation: focusing on homogeneous base detectors only
Future direction: include heterogeneous base detectors for more diversity
Reference