Clustered Hierarchical Anomaly and Outlier Detection Algorithms

Thomas J. Howard III,[1*] **Najib Ishaq,** [1*] **Noah M. Daniels** [1]

[1] Dept. of Computer Science and Statistics
University of Rhode Island
Kingston, RI
thoward27@uri.edu, najib_ishaq@uri.edu, noah_daniels@uri.edu

Abstract

Abstract is written last.

Introduction

TODO

Related Works

TODO

CHAODA

CLAM

We present a Manifold-Mapping algorithm called CLAM (Clustered Learning of Approximate Manifolds). This is an extension of earlier work presented in (Ishaq, Student, and Daniels 2019).

To start, we need a dataset and a distance function on the points in that dataset. A Dataset is a collection of n-points in a D-dimensional embedding space.

$$\mathbf{X} = \{x_1 \dots x_n\}, x_i \in \mathbb{R}^D$$

A Distance Function takes two points in the dataset and deterministically produces a non-negative real number.

$$f : (\mathbb{R}^D, \mathbb{R}^D) \mapsto \mathbb{R}^+$$

We require any distance function to have the following properties:

$$\forall x \in X,\, f(x, x) = 0$$
$$\forall x, y \in X,\, f(x, y) = f(y, x)$$

The distance function may or may not obey the triangle-inequality.

Clustering: We start by building a divisive-hierarchical clustering of the data. This gives us a tree of Clusters, with the root containing every point in the dataset, and each leaf containing a single point from the dataset. The procedure is detailed in (Ishaq, Student, and Daniels 2019).

Some important Cluster properties to consider are:

- *Cardinality*, i.e. the number of points in a cluster.
- *Center*, i.e. the geometric median of points contained in a cluster.
- *Radius*, i.e. the distance to the farthest point from a center.
- *Local fractal dimension*, as described in (Ishaq, Student, and Daniels 2019).
- *Parent-Child ratios* of cardinality, radius, and local fractal dimension.
- *Exponential moving averages* of the parent-child ratios along a branch of the tree.

In particular, we use the parent-child ratios and the exponential moving averages of those ratios to help generalize our anomaly detection method from a small set of datasets to a large, distinct set of datasets.

Graphs: Clusters that are near each other in the embedding space sometimes have overlapping volumes, i.e. the distance between their centers is less than or equal to the sum of their radii. We define a Graph with the clusters as the nodes and with edges between overlapping clusters. For our purposes, a Graph also has the following additional properties:

- The clusters in the graph collectively contain every point in the dataset.
- Each point in the dataset is in exactly one cluster in the graph.

A Graph can be built from clusters at a fixed depth in the cluster-tree, or can contain clusters from multiple different depths in the tree. We say that the cardinality of a graph is the number of clusters in that graph.

The Manifold: According to the Manifold Hypothesis (Fefferman, Mitter, and Narayanan 2016), datasets that come from constrained generating processes and are embedded in a high-dimensional space actually only occupy a low-dimensional manifold in that embedding space.

The graphs discussed so far map this low dimensional manifold in the original embedding space. Different graph do this at different levels of local and/or global resolution. Our aim is to properly build such a graph, i.e. we want to be "properly zoomed-in" to the various regions of the manifold formed by the data. We can then apply various anomaly

[*] These authors contributed equally to this work.
[*] These authors contributed equally to this work.

detection algorithms to these graphs. These algorithms will often also incorporate information from the tree.

We describe some algorithms in . While these algorithms are themselves fairly simple, the real challenge is in selecting the right clusters for the graphs that the algorithms operate on. We will demonstrate CLAM to be such a powerful technique in Manifold-Mapping that even such simple algorithms as described in more often than not outperform state-ot-the-art anomaly detection algorithms.

Individual Algorithms

Here we describe several simple methods for anomaly detection. Each of these methods uses a graph of clusters from CLAM to calculate an anomalousness score for each point in the dataset.

Relative Cluster Cardinality: We measure the anomalousness of a point by the cardinality of the cluster that point belongs to relative to the cardinalities of the other clusters in the graph. Points in the same cluster are considered equally anomalous and points in clusters with relatively low cardinalities are considered more anomalous than points in clusters with relatively high cardinalities.

Child-Parent Cardinality Ratio: As described in CHESS (Ishaq, Student, and Daniels 2019), a cluster is partitioned by using its two maximally distant points as poles. The points are split among children by whichever pole they are closer to. Consider the fraction of points in a cluster that are assigned to each child. If a child cluster only contains a small fraction of the points that its parent did, then we consider the points in that child cluster to be anomalous. These child-parent cardinality ratios are accumulated for each point down its branch in the tree, terminating when the child cluster is a node in the graph. Points with a low value of these accumulated ratios are considered more anomalous than points with a high value of these accumulated ratios.

Graph Neighborhood Size: Given the graph with clusters and edges, consider the number of clusters reachable from a starting cluster within a given graph distance k. We call this number the *graph-neighborhood* of the starting cluster. When k is relatively small compared to the diameter of the graph, we can consider the relative *graph-neighborhoods* of every cluster in the graph. Points in clusters with small graph-neighborhoods are considered more anomalous than points in clusters with large graph-neighborhoods.

Relative Subgraph Cardinality We define disconnected components of a graph by the property that no two clusters from different disconnected components have an edge between them. Consider the relative cardinalities of each component in much the same we we considered the relative cardinalities of clusters in the Cluster Cardinality method. Points in clusters in the same component are considered equally anomalous and points in clusters in relatively small components are considered more anomalous than points in clusters in relatively large components.

Meta Machine Learning

The heart of the problem with our methods of anomaly detection is building the right graph to represent the underlying manifold. One could try using every possible combination of clusters to form a graph but this quickly leads to combinatorial explosion. Instead, we must intelligently select those clusters that, when used to build the graph, perform best for anomaly detection.

AUC ROC is often used to measure the performance of anomaly detectors; we will choose clusters so as to maximize this measure. Our aim is to learn a function that takes a cluster and predicts contribution to AUC from that cluster. Any function of the following form will suffice.

$$f : Cluster \mapsto \mathbb{R}^+$$

We chose a simple Linear Regression to fill this role. Such a model needs some data to train with. To generate this data, we took a random sample of some of the datasets described in . We generate CLAM Manifolds for these training datasets, use the Linear Regression model to learn from these datasets, and apply the results to an entirely different set of datasets.

We generate the initial training data by considering graphs built from clusters at a uniform depth in the tree. For each such graph, we calculate the means of the cardinality ratio, radius ratio, and local fractal dimension ratio, and the exponential moving averages of these ratios. These ratios are described in . These ratios form the feature vector for one training sample. We apply a method described in to the graph and obtain an AUC ROC. We then train the Linear Regression model to predict this AUC from the features extracted from the graph. We train a separate Linear Regression model for each method described in .

Ensemble

Given the learned meta-ml model from the training datasets, we can use it to build graphs for any other dataset. In our case with linear regression as that meta-ml model, we use the cluster ratios and the associated regression constants to rank every cluster in a CLAM tree. These rankings are normalized by the cardinality of each cluster. The highest ranked clusters are then used to build a graph. This graph is then used with the corresponding individual algorithm to calculate anomaly scores for all points in the dataset. The scores from each individual algorithm are then combined into an ensemble. We present the AUC scores from this ensemble in the Results section.

Datasets

TODO

Results

TODO

Discussion

TODO

Acknowledgements

TODO

Table 1: Performance on Train Datasets

dataset	annthyroid	mnist	pendigits	satellite	shuttle	thyroid
ensemble-L2	0.62	0.63	0.77	**0.63**	0.84	0.87
ensemble-L1	0.62	0.59	**0.96**	0.41	0.81	**0.96**
CBLOF	**0.76**	0.56	0.64	0.58	**0.98**	**0.96**
COF	0.51	0.58	0.55	0.54	*time*	0.52
HBOS	0.62	0.51	0.83	**0.65**	0.96	0.88
IFOREST	0.65	0.60	0.92	**0.64**	**0.98**	0.94
KNN	0.72	0.62	0.56	0.57	0.60	0.93
LMDD	0.70	*time*	0.69	0.44	*time*	0.93
LOCI	*time*	*time*	*time*	*time*	*time*	*time*
LODA	0.53	0.62	0.77	**0.64**	0.53	0.61
LOF	0.52	0.55	0.55	0.53	0.53	0.54
MCD	0.73	**0.67**	0.60	**0.65**	**0.96**	0.94
OCSVM	0.71	0.63	0.80	**0.64**	**0.98**	0.94
SOD	0.69	0.57	0.58	0.56	*time*	0.87
SOS	0.50	0.51	0.51	0.48	*time*	0.50

Table 2: Performance on the first half of the Test Datasets

dataset	arrhythmia	breastw	cardio	cover	glass	http	ionosphere	lympho	mammography
ensemble-L2	**0.72**	**0.93**	0.66	0.82	**0.82**	**1.00**	**0.81**	0.92	0.62
ensemble-L1	0.63	0.85	0.50	0.75	0.76	**0.99**	0.79	0.87	**0.81**
CBLOF	**0.71**	0.63	0.70	0.72	0.50	0.96	0.64	0.88	0.65
COF	0.64	0.43	0.54	*time*	0.56	*time*	0.63	0.71	0.60
HBOS	0.68	0.64	0.73	0.62	0.50	0.97	0.45	**0.97**	0.68
IFOREST	0.65	0.64	0.67	0.83	0.56	0.96	0.64	**0.97**	0.73
KNN	0.69	0.59	0.58	0.52	0.56	0.48	0.63	**0.98**	0.68
LMDD	0.66	0.64	0.70	*time*	0.45	*time*	0.62	0.71	0.77
LOCI	**0.70**	*time*	*time*	*time*	0.52	*time*	0.64	0.82	*time*
LODA	0.64	0.64	0.71	**0.88**	0.56	0.48	0.61	0.53	0.65
LOF	0.67	0.44	0.52	0.55	0.57	0.47	0.59	**0.97**	0.63
MCD	0.70	0.64	0.71	0.52	0.45	0.96	0.64	0.80	0.49
OCSVM	**0.71**	0.64	**0.75**	*time*	0.50	0.96	0.64	**0.97**	0.75
SOD	0.62	0.62	0.57	*time*	0.56	*time*	0.64	0.62	0.63
SOS	0.51	0.50	0.50	*time*	0.50	*time*	0.64	0.62	*time*

References

Agrawal, R.; Gehrke, J.; Gunopulos, D.; and Raghavan, P. 1998. Automatic subspace clustering of high dimensional data for data mining applications. In *Proceedings of the 1998 ACM SIGMOD international conference on Management of data*, 94–105.

Banach, S. 1929. Sur les fonctionnelles linéaires II. *Studia Mathematica* 1: 223–239.

Carlsson, G. 2009. Topology and data. *Bulletin of the American Mathematical Society* 46(2): 255–308.

Charles Zahn. ???? Graph Theoretic Methods for Detecting and Describing Gestalt Clusters C-20(1). URL https://www.cs.bgu.ac.il/~icbv161/wiki.files/Readings/1971-Zahn-Graph_Theoretic_Methods_for_Detecting_and_Describing_Gestalt_Clusters.pdf.

Elsayed, G.; Shankar, S.; Cheung, B.; Papernot, N.; Kurakin, A.; Goodfellow, I.; and Sohl-Dickstein, J. 2018. Adversarial examples that fool both computer vision and time-limited humans. In *Advances in Neural Information Processing Systems*, 3910–3920.

Table 3: Performance on the second half of the Test Datasets

dataset	musk	optdigits	pima	satimage-2	smtp	vertebral	vowels	wbc	wine
ensemble-L2	**1.00**	0.58	0.45	0.95	**0.86**	0.41	0.69	0.76	0.71
ensemble-L1	**1.00**	0.49	**0.60**	**0.99**	0.82	**0.46**	0.81	0.78	0.70
CBLOF	0.64	0.47	0.54	0.96	0.82	**0.46**	0.72	0.82	0.45
COF	0.53	0.52	0.54	0.61	*time*	**0.46**	0.89	0.78	0.45
HBOS	0.96	**0.75**	0.57	0.93	0.81	**0.46**	0.57	**0.85**	0.77
IFOREST	0.96	0.53	0.56	0.95	0.83	**0.46**	0.62	0.82	0.61
KNN	0.59	0.48	0.54	0.80	0.82	0.45	**0.94**	0.80	0.45
LMDD	0.96	0.45	0.53	0.45	*time*	0.44	0.51	0.75	0.66
LOCI	*time*	*time*	*time*	*time*	*time*	**0.47**	*time*	0.78	0.53
LODA	0.96	0.46	0.54	0.94	0.82	0.44	0.58	0.82	0.66
LOF	0.65	0.54	0.51	0.60	0.59	**0.46**	0.80	0.83	0.61
MCD	0.96	0.45	0.54	0.96	0.83	0.44	0.51	0.80	**0.88**
OCSVM	0.96	0.45	0.54	0.93	*time*	0.44	0.59	0.82	0.50
SOD	0.62	0.48	0.55	0.69	*time*	**0.46**	0.82	0.80	0.45
SOS	0.49	0.52	0.52	0.50	*time*	**0.48**	0.61	0.52	0.45

Ester, M.; Kriegel, H.-P.; Sander, J.; Xu, X.; et al. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, 226–231.

Fefferman, C.; Mitter, S.; and Narayanan, H. 2016. Testing the manifold hypothesis. *Journal of the American Mathematical Society* 29(4): 983–1049.

Guha, S.; Rastogi, R.; and Shim, K. 1998. CURE: an efficient clustering algorithm for large databases. *ACM Sigmod record* 27(2): 73–84.

Hinneburg, A.; Keim, D. A.; et al. 1998. An efficient approach to clustering in large multimedia databases with noise. In *KDD*, volume 98, 58–65.

Ishaq, N.; Student, G.; and Daniels, N. M. 2019. Clustered Hierarchical Entropy-Scaling Search of Astronomical and Biological Data. In *2019 IEEE International Conference on Big Data (Big Data)*, 780–789. IEEE.

Karypis, G.; Han, E.-H.; and Chameleon, V. K. 1999. A hierarchical clustering algorithm using dynamic modeling. *IEEE Computer* 32(8): 68–75.

Kaufman, L.; and Rousseeuw, P. J. 2009. *Finding groups in data: an introduction to cluster analysis*, volume 344. John Wiley & Sons.

Keller, F.; Muller, E.; and Bohm, K. 2012. HiCS: High contrast subspaces for density-based outlier ranking. In *2012 IEEE 28th international conference on data engineering*, 1037–1048. IEEE.

MacQueen, J.; et al. 1967. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, 281–297. Oakland, CA, USA.

McInnes, L.; Healy, J.; and Melville, J. 2018. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426* .

Ng, R.; and Han, J. 1994. :" Efficient and Effective Clustering Methods for Spatial Data Mining", Proc. 20th Int. Conf. On Very Large Data Bases, Santiago, Chile, Morgan Kaufmann Publishers .

Sheikholeslami, G.; Chatterjee, S.; and Zhang, A. 2000. WaveCluster: a wavelet-based clustering approach for spatial data in very large databases. *The VLDB Journal* 8(3-4): 289–304.

Wang, H.; Bah, M. J.; and Hammad, M. 2019. Progress in Outlier Detection Techniques: A Survey. *IEEE Access* 7: 107964–108000.

Wang, W.; Yang, J.; Muntz, R.; et al. 1997. STING: A statistical information grid approach to spatial data mining. In *VLDB*, volume 97, 186–195.

Zhang, J. 2013. Advancements of outlier detection: A survey. *ICST Transactions on Scalable Information Systems* 13(1): 1–26.