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

Detecting anomalies and outliers from data is a well-studied problem in machine learning. When data occupy easily-described distributions, such as the Gaussian, the task is relatively easy: one need only identify when a datum is sufficiently far from the mean. However, in "big data" scenarios, where data can occupy high-dimensional spaces, anomalous behavior becomes harder to quantify. If the data happen to be uniformly distributed, one can conceive of simple mechanisms, such as a one-class SVM, that would be effective in any number of dimensions. However, real-world data are rarely distributed uniformly. Instead, data often obey the "manifold hypothesis" (Fefferman, Mitter, and Narayanan 2016), occupying a low-dimensional manifold in a high-dimensional embedding space. This low-dimensional manifold may weave itself through the high-dimensional space much like a crumpled sheet of paper does in 3-dimensional space. Detecting anomalies in such a landscape is not easy; in particular, correctly identifying an anomalous datum that sits within the gaps of a lower-dimensional manifold presents a challenge.

Anomalies (data that do not belong to a distribution) and outliers (data which represent extrema of a distribution) can arise from many sources: errors in measurement or collection of data; novel, previously-unseen instances of data; normal behavior evolving into abnormal behavior; and adversarial attacks as inputs to machine-learning algorithms (Elsayed et al. 2018). Modern algorithms designed to detect anomalous behavior fail for a variety of reasons, in particular when anomalies live close to, but not on, a complex manifold in high-dimensional space. Our approach is designed to learn these complex manifolds. Here we briefly survey contemporary approaches to anomaly detection in order to provide the context needed to understand how our approach differs.

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

Related Works

Clustering-based Approaches

Clustering refers to techniques for grouping points in a way that provides value. This is generally done by assigning *similar* points to the same cluster. Given a clustering and a new point, one can estimate the anomalousness of the new point by measuring its distance to its nearest cluster.

There have been few advancements in clustering techniques over the past decade (Wang, Bah, and Hammad 2019). This may be explained by the poor performance of clustering in high-dimensional space (Zhang 2013) thus far. The major approaches are as follows.

Distance-based clustering relies on some distance measure to partition data into some number of clusters. Within this approach, the numbers and/or sizes of clusters are often predefined: either user-specified, or chosen at random (Wang, Bah, and Hammad 2019). Some examples of distance-based clustering are: K-Means (MacQueen et al. 1967), PAM (Kaufman and Rousseeuw 2009), CLARANS (Ng and Han 1994) and CLARA (Kaufman and Rousseeuw 2009).

Hierarchical clustering methods utilize a tree-like structure, where points are allocated into leaf nodes (Wang, Bah, and Hammad 2019). These tree-like structures can be created bottom-up (agglomerative clustering) or top-down (divisive clustering) (Agrawal et al. 1998). A major drawback of these methods is the high cost of pairwise difference computations required to build each level of the tree. Examples of hierarchical clustering include MST (Charles Zahn), CURE (Guha, Rastogi, and Shim 1998) and CHAMELEON (Karypis, Han, and Chameleon 1999).

Density-based clustering methods rely on finding regions of high point-density separated by regions of low point-density. These algorithms generally do not work well when data are sparse or uniformly distributed. Some examples of density-based clustering algorithms are DBSCAN (Ester et al. 1996) and DENCLUE (Hinneburg, Keim et al. 1998).

Grid-based clustering works via segmenting the entire space into a discrete number of cells, and then scanning those cells to find regions of high density. Utilizing a grid structure for clustering means that these algorithms typically scale well to larger datasets. Some examples of grid-based clustering include STING (Wang et al. 1997), Wavecluster (Sheikholeslami, Chatterjee, and Zhang 2000),

and CLIQUE (Agrawal et al. 1998).

Distanced-based Approaches

Distance-based methods find anomalous points via distance comparisons. These methods largely employ k-Nearest Neighbors as their substrate (Wang, Bah, and Hammad 2019). Distance-based approaches tend to use the following intuitions: points with fewer than p other points within some distance d are outliers; the n points with the greatest distances to their $k^{th}-$nearest neighbor are outliers; or the n points with the greatest average distance to their k nearest neighbors are outliers.

CHAODA

In this paper we introduce a novel technique, Clustered Learning of Approximate Manifolds (CLAM). This approach uses divisive hierarchical clustering to learn a manifold in a Banach space (Banach 1929) defined by a distance metric. In actuality, we do not require a metric. The space may be defined by a distance *function* that does not obey the triangle inequality, though this is not always optimal. Given a learned approximate manifold, we can almost trivially implement several anomaly-detection algorithms. In this manuscript, we present a collection of five such algorithms implemented on CLAM: CHAODA (Clustered Hierarchical Anomaly and Outlier Detection Algorithms).

The manifold learning component is derived from prior work, CHESS (Ishaq, Student, and Daniels 2019), to accelerate approximate search on large high-dimensional datasets. CLAM begins by divisively clustering the data until each cluster contains only one datum. CLAM then delineates *layers* of clusters at each depth in the tree. Each layer comprises all clusters that would have been leaf nodes if the tree building had been halted at the given depth.

CLAM then builds a graph for each layer in the tree by creating edges between clusters that have overlapping volumes. This process effectively learns the manifold on which the data lie at various resolutions, given by the depth of the layer. This is analogous to a "filtration" in computational topology (Carlsson 2009). Once we have learned a manifold, we can ask about the cardinality of various clusters at different depths, how connected a given cluster is, or even how often a cluster is visited by random walks on the manifold.

We test our methods on 24 real-world datasets. The datasets span a wide variety of domains, each having a different quantity of anomalous data. We consider several different definitions of outliers and anomalies: **distance-based**, examining several classical distance-based definitions of outliers, relying on CLAM's use of distance to cluster data; **density-based**, examining the cardinality of clusters, under the hypothesis that clusters with lower cardinality are more likely to contain outliers; **graph-based**, examining several graph-theoretic methods for anomaly detection, given graphs constructed from layers of clusters.

Historically, clustering approaches have suffered from several problems. The most common deficiencies are: the effective treatment of high dimensionality, the ability to interpret results, and the ability to scale to exponentially-growing datasets (Agrawal et al. 1998). CLAM largely resolves these problems.

Methodology

The Manifold

The Manifold is built from a dataset and a distance function. We start by calculating a divisive-hierarchical clustering on the data. This gives us a tree of Clusters, with the root containing every point in the dataset, and the leaves containing single points from the dataset. The procedure is detailed in (Ishaq, Student, and Daniels 2019).

Each Cluster has a center, i.e. the geometric median of points contained in that cluster, and a radius, i.e. the distance to the farthest point from the center. Nearby clusters can 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 as containing clusters as the nodes. Overlapping clusters have an edge connecting them in the Graph. Further, a Graph has the following two 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 use the Manifold to keep track of the cluster-tree and any associated Graphs. The algorithms described in work using one of these Graphs.

Individual Algorithms

Results

Discussion

We have presented CHAODA, a collection of five algorithms that exploit properties of a hierarchical cluster tree which represents an approximate manifold in n-dimensional space. The five algorithms are simple to implement on top of the manifold-learning framework we call CLAM. CHAODA builds on this framework for anomaly detection in much the same way as CHESS (Ishaq, Student, and Daniels 2019) did for accelerating approximate search. With CHESS, the geometric and topological properties of low fractal dimension and low metric entropy are advantages; indeed, CHESS does not offer asymptotic improvements over linear search if these properties are absent. CHAODA, on the other hand, while competitive with other state-of-the-art anomaly-detection approaches on "easy" datasets (we define as *easy* any dataset where a one-class SVM performs well), outperforms other current methods when the data exhibit precisely those properties that CHESS exploits for acceleration. In particular, CHAODA outperforms other approaches on high-dimensional datasets, with the exception of the "annthyroid" dataset where iForest and AOD perform better.

Except for the annthyroid dataset, the algorithms presented here outperform or at least nearly match all other approaches. CHAODA outperforms a 1-class SVM on every dataset, and

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*

of the datasets where AUC values were available for other results (20 datasets), CHAODA matches or exceeds the AUC of other approaches on 12 of them. On 5 of the remaining 8, CHAODA is close to the best-performing approach, typically within 1 and 3 percentage points.

Several reasons may contribute to CHAODA's difficulty with the annthyroid dataset in particular. First, this dataset was specifically created for use with ANNs. Upon further investigation, it appears that these data may be in too few dimensions for CLAM to partition it into a useful manifold. In UMAP (McInnes, Healy, and Melville 2018) projections we created on annthyroid Figure 1, it can be clearly seen that the anomalous data appear to live directly on the manifold, with only a small pocket appearing to be distinctly off. In contrast, the wbc dataset, where CHAODA significantly outperforms HiCS (Keller, Muller, and Bohm 2012), appears to have most of the outliers along the periphery of the manifold. This aligns well with our expectations of CHAODA. Indeed, the manifold being both learnable and distinctly separate from the anomalous data are mandatory properties for any of our approaches to be effective. Fortunately, we can see that these properties are apparent in all other datasets studied.

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

One current limitation in CHAODA is that the depth of the cluster tree at which anomaly detection performs best is not the same for every dataset; thus, our results could be seen as "cherry-picking" from a scattershot approach. The optimal depth varies because as depth increases, the induced graph "shatters", i.e. the number of components in the graph approaches the number of clusters in the graph. Future work should explore optimal stopping criteria so that we can automate stopping just before the graph shatters. Fortunately, as shown in Figures, for most datasets, performance is not overly sensitive to the choice of depth, especially for the Parent-Child algorithm. For now we can treat depth as a hyperparameter to all of the methods described, but a detailed analysis of possible stopping criteria for clustering depth will likely reveal automatic methods to find the optimal depth.

The choice of distance function also has a significant impact on anomaly-detection performance. In this case, domain knowledge is likely the best way to determine the distance function of choice. Future work will seek to explore a more diverse collection of domain-appropriate distance functions, such as Wasserstein distance on images, Levenshtein edit distance on strings, and Jaccard Index on the Maximal Common Sub-Graph of molecular structures.

In conclusion, we have demonstrated that by learning approximate manifolds, we can exploit the embedded knowledge to implement simple algorithms capable of outperforming other state-of-the-art approaches to anomaly detection.

Acknowledgements

The authors would like to thank the members of CSC 592 - Algorithms for Big Data for helpful feedback and discussions.

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.

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.

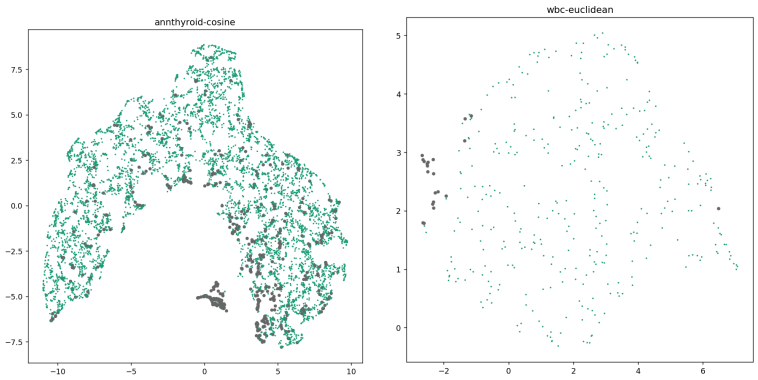

Figure 1: UMAP projection of Annthyroid (left) and WBC (right). Anomalies are in gray. Note that for Annthyroid, while there is a cluster of anomalies off the main manifold, many anomalies are distributed throughout the manifold. For WBC, the anomalies tend to be at the edge of the manifold.

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.