Fast and Scalable Outlier Detection with Metric Access Methods*

Altamir Gomes Bispo Junior and Robson Leonardo Ferreira Cordeiro

University of São Paulo, São Carlos, São Paulo, Brazil

Abstract. It is well-known that the existing theoretical models for outlier detection make assumptions that may not reflect the true nature of outliers in every real application. With that in mind, this paper describes an empirical study performed on **unsupervised outlier detection** using 8 algorithms from the state-of-the-art and 8 datasets that refer to a variety of real-world tasks of high impact, like spotting cyberattacks, clinical pathologies and abnormalities in nature. We present the lowdown on the results obtained, pointing out to the strengths and weaknesses of each technique from the **application specialist's point of view**, which is a shift from the designer-based point of view that is commonly considered. Interestingly, many of the techniques had unfeasibly high runtime requirements or failed to spot what the specialists consider as outliers in their own data. To tackle this issue, we propose *MetricABOD*: a novel angle-based outlier detection algorithm that makes the analysis up to **thousands of times faster**, still being in average **26% more accurate** than the most accurate related work. This improvement is essential to enable outlier detection in many real-world applications for which the existing methods lead to unexpected results or unfeasible runtime requirements. Finally, we studied two real collections of text data to show that our *MetricABOD* works also for adimensional, purely metric data.

Keywords: Applied Computational Sciences · Complex Data · Data Mining · Unsupervised Outlier Detection · Metric Access Methods

1 Introduction

An important task with high-impact real-world consequences is the forecasting and detection of extreme events and exception cases, like frauds in the financial sector, cyberattacks, clinical pathologies and abnormalities in nature. Such phenomena are known as outliers. The volume of data being collected by enterprises from a variety of scientific areas is increasing exponentially over time, thus forcing analysts to use automatic procedures to detect outliers [6, 8].

Unfortunately, the present literature lacks ample and thoughtful comparison between the many existing outlier detection methods. One recent work [5] succeeded to reduce this gap, but, according to its authors, it is a "meta-analysis".

* This work was partially supported by the São Paulo Research Foundation (FAPESP) – Grant N° 2018/05714-5, by the Coordination for Improvement of Higher Education Personnel (CAPES) – Finance Code 001, and by the National Council for Scientific and Technological Development (CNPq).

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22741-8_14

That is, their work aims to be as generalist as possible and, as such, it does not focus on any application nor domain and does not discuss accuracy in real applications from the point of view of the specialist users, i.e., "what the application specialists expect from the data". The major merit of their work lies in that it presents useful and very needed pointers to guide further investigation over a diversity of application domains. However, it is still unclear how accurate most of the existing methods are when spotting what the specialists consider as outliers in datasets of distinct natures, such as those collected in the many real-world applications that can benefit from the detection of outliers.

It is well-known in the literature that the existing theoretical models for unsupervised outlier detection make certain assumptions that may not reflect the true nature of specific outliers in every single application [1, 5]. With that in mind, we performed an empirical study to evaluate 8 state-of-the-art outlier detection algorithms using 8 datasets from a variety of real-world applications with ground truth data manually created by specialist users; we were focused on verifying whether or not the algorithms are able to spot in an automatic and timely manner what the specialists selected as outliers. Interestingly, many of the algorithms had unfeasibly high runtime requirements or failed to return what the specialists expect from their data. This paper's main contributions are:

1. **Empirical evaluation:** we evaluated 8 state-of-the-art outlier detection methods from the **application specialist's point of view**, and report results focused on quantifying how useful they can be for a variety of real world tasks of high impact, such as spotting breast cancer, heart and thyroid anomalies, detecting cyberattacks, and detecting musk species;
2. ***MetricABOD*:** we carefully designed a new angle-based algorithm that makes the analysis up to **thousands of times faster**, still being in average **26% more accurate** than the most accurate related work. The main innovation is a new usage for tree-based data structures known as Metric Access Methods (MAM) [17], which were originally designed to index complex data, such as images, audio, large graphs and fingerprints. This improvement was essential to enable outlier detection in many of the datasets that we studied, for which the existing methods lead to unexpected results or unfeasible runtime requirements. Finally, we studied real collections of text data to show that our *MetricABOD* works also for adimensional, purely metric data.

The rest of this paper is organized as follows: background concepts (Section 2), related works (Section 3), empirical evaluation (Section 4), proposed algorithm (Section 5) and conclusions (Section 6).

2 Background

There are many definitions for outliers in the literature. The distance-based one [10] is very commonly used: *"An object P in a dataset T is a $DB(f, \xi)$-outlier if at least one fraction f of the objects in T have a distance to P that is greater than ξ."* Here, term $DB(f, \xi)$-outlier is a shorthand notation for a Distance-Based outlier with supporting parameters f and ξ. This definition is independent of the data distribution and it is also intuitive, simple and practical. Due to such qualities, it serves as a basis for several outlier detection algorithms [4, 3].

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22741-8_14

In spite of that, there is not a single, universally accepted definition for what an outlier is, neither an unanimous way to compute outlierness scores. Outliers matter to statisticians since long ago. Research works in the domain of Statistics routinely assume a parametric distribution for the data, following a fixed set of parameters. However, it is not safe to assume, a priori, any parametric distribution for a dataset in most real-world scenarios. And even if the distribution is assumed as non-parametric, the existing unsupervised outlier detection algorithms must also assume a specific definition of outliers within the distribution to create a model, and then use the model to search for data objects that fit this definition. Obviously, different algorithms employ distinct models, thus they may obtain different results. Since there is no consensus, it is not unusual that two given algorithms' results clash [12].

Let us highlight that the application specialist user expects results of his/her true interest, despite the model and algorithm at hand. At this point, we make a shift from the designer-based point of view that is routinely seen in the literature, to the user-based one. Even elaborate and theoretically sound techniques may not translate to acceptable results for the end user. Also, many real-world datasets from which spotting outliers is desirable have millions of objects and hundreds of attributes. Outlier detection in these data is a demanding task [6]. In fact, it is truly an unfeasible task in most cases. The main challenges are: (a) very large runtime requirements, even for approximate methods; (b) low availability of ground truth to assess a method's accuracy; (c) data of high dimensionality and its unwanted effects, and; (d) sensitivity to input parameter values.

The next subsections discuss the main undesirable effects of dealing with data of high dimensionality; we also present the basics of Metric Access Methods.

2.1 Curse of high dimensionality

As the number of dimensions of a dataset increases, the variance of distances between any two data objects tends to zero [1]. Figure 1 illustrates this fact by presenting dissimilarity matrices for two toy datasets with 10 objects each. The objects were randomly picked from a multivariate normal distribution. Figures 1a and 1b respectively refer to data of low and high dimensionality, i.e., 5 and 100 dimensions. Darker colors represent smaller distances; lighter colors refer to larger distances. As it can be seen, it is much harder to discriminate the distinct dissimilarities in the high-dimensional scenario than it is in the low-dimensional one. This simple example illustrates the evanescence of the variance of pairwise dissimilarities between objects, as the dimensionality increases.

In this fashion, according to the distance-based outlier definition, and also to most of the other existing definitions, all objects in a dataset of high dimensionality are potential outliers. To tackle this problem with subspace selection or dimensionality reduction is not obvious. According to [1], the main difficulties are: (a) circular references between neighborhoods and subspaces; (b) noise with its masking and dilution effects, and; (c) rarity-based subspace outliers. State-of-the-art solutions to the problem perform angle-based analysis, rather than the traditional distance-based one, as we discuss later in the paper.

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22741-8_14

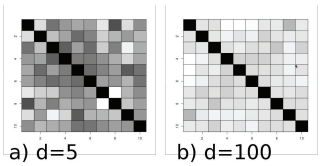


Fig. 1: Dissimilarity matrix for two toy datasets with 10 random objects each. a) 5 attributes; b) 100 attributes.

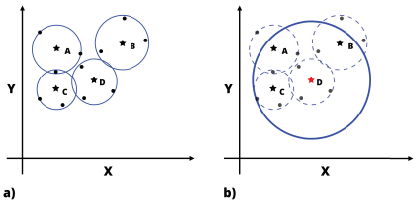


Fig. 2: 2-dimensional objects at: a) level h of a tree-based MAM structure; b) level $h-1$ of the same structure.

2.2 Metric Access Methods

Metric Access Methods (MAM) [17] allow efficient queries upon datasets embedded in a *metric space*. A metric space is formally defined by a data domain X and a distance function $d : X \times X \to \mathbb{R}$, such that the rules as follows apply:

i) Positivity: for all $x, y \in X$, $0 \leq d(x, y) < \infty$
ii) Non-degenerated: for all $x, y \in X$, $d(x, y) = 0 \leftrightarrow x = y$
iii) Symmetry: for all $x, y \in X$, $d(x, y) = d(y, x)$
iv) Triangle inequality: for all $x, y, z \in X$, $d(x, y) \leq d(x, z) + d(z, y)$

A metric space may not allow sorting objects. Thus, MAM structures must use distances only; nothing else. State-of-the-art MAMs are tree-based. Every tree node stores a subset of objects. One of them is selected to be the node's representative, which is also known as the pivot. The representative is usually the object with the shortest average distance to all others. The node's region of representation is a hyper-sphere centered at the representative object with a radius that is greater than or equal to the distance from it to any other object in the same node. Figure 2 illustrates a tree-based MAM upon objects in a two-dimensional space. We assume that each node stores up to 4 objects. In Figure 2a, four nodes with representatives A, B, C and D are defined out of all objects at the lowest level h. At level $h-1$, only these representatives are considered; as it can be seen in Figure 2b, object D is selected to represent the previous level's representatives into a single root node. When using a MAM, the triangle inequality property and the representatives serve as a basis to prune unnecessary distance calculations, thus leading to efficient data manipulation.

3 Related Works

To evaluate the distances from objects to their k-th nearest neighbors is one of the simplest approach to spot outliers. The *KNN-Outlier* method ranks the objects' respective distances to their k-th nearest neighbors, or KNN distances, from longest to shortest, and gives higher outlierness scores to objects with higher KNN distances. It is the basis for many other works, like *Orca* [3] and *RBRP* (Recursive Bin Partitioning and Re-projection) [7].

LOF (Local Outlier Factor) [4] is another well-known outlier detection method. It introduced the concepts of core distance and reachability distance, inspiring many posterior works, such as *LOCI* (Local Correlation Integral) [15],

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22741-8_14

aLOCI (Approximated Local Correlation Integral) [15] and *ABOD* (Angle-Based Outlier Detection) [11]. According to *LOF*, a *core object* is an object that has at least a certain number of neighbours in its ξ-neighborhood. An object x is *density-reachable* from object y if there is a chain of objects $p_1, p_2, ...p_q$, where $p_1 = y$ and $p_q = x$, such that p_{i+1} is in the ξ-neighborhood of p_i for $i \in \{1, 2, ... \ q-1\}$. The *core-distance* of an object x is the smallest distance ξ that makes it a core object. The *reachability-distance* of x is the smallest distance ξ that puts it in the ξ-neighbourhood of a core object y. LOF employs these concepts to calculate outlierness scores for all data objects according to their ξ-neighbourhoods' density of objects in the feature space.

From here on, we briefly describe methods that were developed having in mind the problems inherent to high dimensional data. Two basic approaches have been explored in the literature: subspace analysis and angle-based analysis. A subspace is a representation of the original space that excludes some of the dimensions/attributes present in the latter. Since a proper subspace has lower dimensionality than that of the original space, the data projection can alleviate the dilution and noise effects present in higher dimensional spaces and may also improve the contrast for outlier detection methods that use distances as a basis.

Unfortunately, the number of distinct subspaces to analyze grows exponentially with regard to the number of attributes of a dataset. Therefore, extensive subspace search is not feasible. Aggarwal and Yu [2] faced this issue using an evolutionary algorithm. Their method works by iteratively improving candidate sets of subspaces until a final set of feasible subspace candidates is generated. OUTRES [14] works by performing statistical tests upon subspace projections of dataset objects, starting from 1-dimensional projections and adding new dimensions one-by-one. Only subspace projections that pass their statistical test are considered. OUTRANK (Outlier Ranking) [13] enumerates as outliers the objects that overlap less, considering their presence or absence inside clusters found in distinct subspaces. HiCS (High Contrast Subspaces) [9] selects feasible subspaces by performing a statistical test upon each candidate subspace. HiCS itself does not measure the outlierness of objects; it merely retrieves and aggregates the results obtained by a coupled outlier detection method, such as LOF. Thus, HiCS is described by its authors as a "meta" outlier detection method. Finally, note that all of the aforementioned methods allow to set thresholds in an attempt to prune uninteresting subspaces.

ABOD (Angle-Based Outlier Detection) [11] uses a different approach. It was inspired by previous research works in the domain of text comparison that had verified that angles are more resilient than distances to the "curse of high dimensionality". The authors of ABOD noted that this fact remains true also when detecting outliers from high-dimensional data in a general context, and not only for text. For each object P, ABOD calculates the angles between every possible pair of objects (x, y) using P as the pivot, such that $x \neq y \neq P$. The variance of angles is then attributed to P as its outlierness score. Small variances of angles indicate outliers; larger variances refer to inliers. The intuition here is that outliers tend to be in the borders of the feature space, so they have neighbors concentrated in one specific direction, while the inliers' neighbours are spread all over the space. Figure 3 illustrates this idea considering an inlier (Figure 3a) and an outlier (Figure 3b). In both cases, colored semi-circles represent some of

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22741-8_14

the angles associated to pairs of objects when taking the object of interest P as the pivot. Figure 4 complements this example by plotting the corresponding angle values and variances.

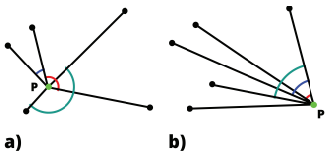


Fig. 3: ABOD's angle-based analysis for an inlier (left) and an outlier (right).

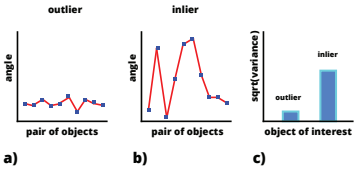


Fig. 4: Angle values and their variances (outlierness scores) for data in Figure 3.

ABOD may not be limited in practice by data dimensionality. However, due to scalability issues, its use is clearly impractical for data of high cardinality. It computes angles for every possible triple of objects, so ABOD is $O(n^3)$ where n is the number of objects. LB-ABOD(Lower Bound-based ABOD) [11], FastABOD [11] and FastVOA (Fast Variance Of Angles) [16] improve upon ABOD focused on tackling its scalability issues. Unfortunately, they either obtain only minor speedup or degrade accuracy considerably.

4 Unsupervised outlier detection at work

Many of the state-of-the-art methods in outlier detection validate their results only on synthetic data of low cardinality. Some methods are also tested for one or two real applications using data with up to a few thousand objects, but not for a broad range of applications in which diversity takes place, nor using datasets that are as large as the ones required for commercial use. Therefore, it is still unclear how efficient and effective these methods are to be used in the real world.

In this section, we shrink this limitation considering a variety of real-world tasks of high impact, like spotting cyberattacks, clinical pathologies and abnormalities in nature. We studied the behaviour of 8 state-of-the-art algorithms face to 8 datasets of distinct natures, that is, data from diverse real applications with varying cardinality and dimensionality. To make it possible, we assume that the ground truth created by specialist users is correct, and verify whether or not the algorithms are able to obtain similar results in an automatic and timely manner. The main motivation here is that outlier detection must be useful in practice, not only in academia. Specifically, we focus on answering the questions as follows:

1. How effective and efficient are the state-of-the-art algorithms to be used in diverse real world applications?
2. What are the best applications for each algorithm?
3. Is there any algorithm with remarkably high accuracy in a general context?

4.1 System configuration

The algorithms studied are: KNN-Outlier, LOF, ABOD, FastABOD, LB-ABOD, FastVOA, aLOCI and HiCS. We used the Elki (https://elki-project.github.io)

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22741-8_14

implementation for all algorithms, except for FastVOA that was evaluated with an implementation of our own. The algorithms were tested in a Xeon E5-2640v3 machine with 16 GB of RAM, running Debian 9 64-bit. The number of samples k was set to 100 in both FastABOD and LB-ABOD. For LB-ABOD, we also set parameter l as two times the number of outliers known to exist in the ground truth. The same selection was used for parameter k in KNN-Outlier, LOF and HiCS. Finally, FastVOA was tuned with t=100, $s1$=1,600 and $s2$=10, following its authors recommendation.

The evaluation was performed by asking each algorithm for the o objects with the highest outlierness scores, where o is the number of outliers present in the ground truth. Considering correctly detected outliers as true positives (tp), the accuracy of each algorithm was calculated as $\frac{tp}{o}$. Every experiment was ran 5 times. We report average accuracy and average runtime.

4.2 Datasets

Table 1 summarizes the datasets studied. They are available at the ODDS repository (http://odds.cs.stonybrook.edu/). All datasets come from real-world applications and include ground truth created by specialist users. For brevity, they are described in the following with focus on the practical benefits of spotting outliers in each case. Detailed descriptions are found in the data source website.

- **BreastW** and **Mammography:** two datasets with features from medical exams of breast cancer suspicious cases. They were collected in separate from two distinct locations. Spotting outliers here means detecting severe health problems, that is, distinguishing between the benign and the malignant cases.
- **Cardio:** features extracted from Fetal Heart Rate (FHR) and Uterine Contraction (UC) of cardiotocograms classified by expert obstetricians. To spot outliers in this scenario is important to prevent and treat heart malfunctions.
- **Annthyroid:** features from medical thyroid exams. Here, to detect outliers refers to the identification of the hypothyroid disease in human patients.
- **Satimage-2:** features from satellite imagery. Here, to spot outliers means uncovering abnormal and unexpected patterns of topography, land use, etc.
- **Shuttle:** features collected from the use of a space shuttle. Identifying outliers here means spotting potentially dangerous in-flight abnormalities.
- **Http (KDDCUP99):** log files of network communication. To spot outliers here means distinguishing between cyber attacks and regular transactions.
- **Musk:** features extracted from molecules classified as musk or non-musk by human experts in chemistry. Spotting outliers in such kind of data helps discovering new material that may be valuable to pharmaceutical corporations.

4.3 Results

Table 2 reports accuracy and runtime results for the algorithms and datasets that we studied[1]. Runtime results are in minutes. As it was expected, some of the

[1] For brevity, the last column of Table 2 reports the results obtained with our proposed *MetricABOD* algorithm. They will be discussed latter in the paper; see Section 5.1.

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22741-8_14

Table 1: Summary of datasets.

Dataset	# Axes	# Objects	# Outliers	Dataset	# Axes	# Objects	# Outliers
BreastW	9	683	239	Mammography	6	11,183	260
Cardio	21	1,831	176	Annthyroid	6	7,200	534
Satimage-2	36	5,803	71	Shuttle	9	49,097	3,511
Http (KDDCUP99)	3	567,479	2,211	Musk	166	3,062	97

algorithms had unfeasibly high runtime requirements for our largest datasets `Shuttle` and `Http (KDDCUP99)`, with $\sim 50k$ and $\sim 500k$ objects respectively. Therefore, we represent with "N/A" the cases that exceeded a timeout limit of **24 hours**. With regard to accuracy, note that many of the state-of-the-art techniques failed to return what the application specialists expect from their own data. LOF, HiCS, FastABOD and FastVOA had the lowest average accuracies, overall. On the other hand, aLOCI, ABOD, LB-ABOD and KNN-Outlier performed considerably better, being the most accurate methods in average. Let us highlight that the original ABOD method obtained the the second best accuracy, but its use is unfortunately prohibitive due to scalability issues. In fact, ABOD is the worst methods in terms of runtime. The variation LB-ABOD is slightly faster without losing much in accuracy, although its runtime requirements are still far from being acceptable for most real-world uses. Finally, FastABOD is tens of times faster than ABOD and LB-ABOD in average, but its accuracy degrades considerably, that is, it spots correct outliers in only 34% of the times.

It is **important** to note that bad results over specifc datasets occurred for all methods, even for KNN-Outlier and ABOD that are overall the most accurate ones. Remarkable examples are: KNN-Outlier with the Http network logs and ABOD with the Musk chemical data. This scenario was already expected; it simply corroborates the fact that the theoretical outlier detection models make assumptions that may not reflect the true nature of outliers in every application.

Table 2: Accuracy/runtime (in minutes) results.

Dataset	ABOD	LABOD	FABOD	LOF	KNN	aLOCI	HiCS	FVOA	MABOD
BreastW	**0.95**/0.04	**0.95**/0.04	0.94/**0.00**	0.32/**0.00**	0.94/**0.00**	0.62/**0.00**	0.22/1.16	0.47/0.16	**0.95**/**0.00**
Cardio	0.52/1.23	0.52/1.25	0.36/0.01	0.23/**0.00**	0.51/**0.00**	0.04/**0.00**	0.32/10.0	0.19/0.46	**0.61**/**0.00**
Sat-2	0.88/58.9	0.88/61.9	0.29/0.06	0.07/0.02	**0.91**/0.02	0.00/**0.00**	0.11/36.5	0.11/1.59	0.90/0.01
Shuttle	*N/A*	*N/A*	0.31/9.24*	0.19/0.70	0.34/0.68	**0.88**/9.28	*N/A*	0.44/20.8	0.58/**0.09**
Annth	0.25/123	0.24/81.8	0.26/0.09	**0.32**/0.02	0.24/0.01	0.10/**0.00**	0.15/28.7	0.15/2.06	0.24/0.01
Mammo	**0.28**/253*	0.11/223*	0.26/0.28	0.23/**0.02**	0.27/0.04	0.08/0.03	0.21/60.9	0.17/3.42	0.26/0.03
Http	*N/A*	*N/A*	*N/A*	0.04/24.0	0.03/58.7	**0.98**/**0.24**	0.05/194	0.00/272*	**0.98**/2.98
Musk	0.07/7.50	0.07/7.33	0.02/0.03	0.00/0.03	**1.00**/0.03	0.98/**0.00**	0.95/8.89	0.05/0.80	0.77/0.01
Average	0.49/73.9	0.46/62.5	0.34/1.38	0.17/3.09	0.53/7.43	0.46/1.19	0.28/48.5	0.19/37.6	**0.66**/**0.39**

* result obtained on one AMD EPYC 7571 machine with 128GB RAM, due to main memory exceed.

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22741-8_14

5 *MetricABOD*

The results reported so far indicate that ABOD is one of the most accurate outlier detectors in a general context. Unfortunately, ABOD's cubic time complexity on the number of objects makes its use impractical for most real-world scenarios. Data dimensionality does not really play a factor in ABOD's runtime, even less with precomputation of inner products to make angle calculations faster. Thus, one must reduce the number of calculations **per object** to speed-up ABOD.

As it was discussed before, ABOD uses every data object P as the pivot for $O(n^2)$ angle calculations, where n is the total number of objects. Specifically, the outlierness score of an object P is the variance of angles between every possible pair of objects (x, y) where P is the pivot. Obviously, x, y and P must be distinct objects. Figures 3 and 4 from the previous Section 3 illustrate this process.

To calculate angles only for pairs (x, y) taken from a **sample dataset** is one clever way to achieve better scalability, still obtaining an outlierness score for **every** data object. It turns the original $O(n^3)$ complexity into $O(n.m^2)$ with $n \gg m$, where n and m are the full dataset cardinality and the sample size respectively. Obviously, the use of samples leads to approximate outlierness scores that may negatively impact the accuracy of results. So, the question is:

– *How to select appropriate objects to be in the sample?*

Random sampling is a quick and unburdensome way to answer this question. In fact, it has already been evaluated in one of the existing ABOD sequels, the FastABOD algorithm. However, in high-dimensional spaces, random sampling sports unbearably magnified sampling errors, thus leading to skewed variances of angles. Due to this fact, FastABOD's authors suggest the use of one distinct sample for each data object: its own k nearest neighbours. In other words, FastABOD computes approximate outlierness scores for each object P by using P as the pivot for angle calculations among its own k-nearest neighbours. Unfortunately, k-nn sampling considerably degrades accuracy, as it could be observed in the results of the previous section.

This section presents a novel sampling strategy to improve ABOD by taking advantage of tree-based Metric Access Methods (MAM). As it was described in Section 2.2, tree-based MAMs have nodes that follow an hierarchical organization. Each node stores a set of objects; one of them is selected to be the node's representative, which is usually the object with the shortest average distance to the others. The leaf nodes store all dataset objects; their representatives are stored redundantly in nodes of the immediate higher tree level, from which representatives are also selected. This organization continues recursively up to the root of the tree; see Figure 2 for an illustrative example.

The nodes of a tree-based MAM are built in a bottom-up fashion, like in a B-tree, so to minimize the number of levels, the number of nodes per level and the overlap among the nodes' hyper-spheres of representation in the feature space. This fact is **essential** to our proposal: it means that the representatives of nodes in any tree level preserve key characteristics of the full dataset, with more or less details according to the level. The highest degree of detail is in the leaf nodes' representatives. Since the representative is the object with the shortest average distance to the others, it is akin to the first moment or center of mass of the

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22741-8_14

node's hyper-sphere. So, it is similar to the median for the node's objects. In fact, as it happens with random sampling, there is a strong correspondence between the representatives' statistical moments and those of the full dataset. Note, however, that low density regions in the feature space end up underrepresented with random sampling, while it never happens with MAM representatives. The representatives also compare favorably with FastABOD's k-nn sampling. Although the k nearest neighbours of an object account for most of its variance of angles, to use them as references does not necessarily discriminate outliers correctly, as it was shown in the previous section. For example, FastABOD fails for the case of an outlier object whose k nearest neighbours spread in many directions, while the rest of the dataset, i.e., the vast majority of it, concentrates in a single direction. Note that this case would not be a problem with MAM representatives; most of them would also concentrate in one direction, just like in the full data.

With that in mind, we **propose** to use the leaf nodes' representatives from any tree-based MAM as samples to speed-up ABOD. Algorithm 1 gives the full pseudo-code; let us call it the ***MetricABOD*** algorithm. We argue that the leaf node's representatives lead to a reasonable estimate for the exact variances of angles, and report latter in the paper experimental results that support our claim. In fact, the representatives even improved accuracy for the datasets that we studied; see details at Section 5.1. Figure 5 illustrates how our *MetricABOD* works. To easy comprehension, the same toy data of our running example from Figure 2 is reused here. There are $n = 15$ objects with two attributes each. At first, a tree-based MAM is created. Four circles in the 2-dimensional space illustrate its leaf nodes. The representative objects are A, B, C and D. In this setting, the approximate outlierness score for an object of interest P is computed per P as the pivot for angle calculations among pairs of objects (x, y), such that $x, y \in \{A, B, C, D\}$ and $x \neq y \neq P$. Colored semi-circles illustrate some of the angles to be computed. Specifically, only $C_2^4 = \frac{4!}{2!.(4-2)!} = 6$ angle calculations are performed to process P. Note that the original ABOD algorithm would require $C_2^{n-1} = \frac{14!}{2!.(14-2)!} = 91$ calculations for the same case.

In a general scenario, our *MetricABOD* avoids computing $C_2^{n-1} - C_2^{m-1}$ angles **per object**, with $n \gg m$, where n and m are respectively the full dataset cardinality and the number of leaf nodes, i.e., the sample cardinality. So, it turns ABOD's $O(n^3)$ overall complexity into $O(n.m^2)$. Note that the time complexities to build and traverse a tree-based MAM are respectively $O(n \log n)$ and $O(n)$ for the most typical scenarios, so the additional tree-related cost does not modify the aforementioned overall complexity of *MetricABOD*. Finally, it is worth noting that the sample size m is linearly correlated with the degree of the tree, i.e., the maximum capacity of a node, so m can be easily tuned for any case of use.

5.1 Results: comparison with related works

This section describes the experiments performed to evaluate our proposed *MetricABOD* algorithm. The state-of-the-art Slim-tree [17] MAM was used as the supporting tree structure in all experiments. *MetricABOD* was validated on the same 8 real world datasets of the previous Section 4 using the same methodology and system configuration. For a fair comparison with the related works, the tree degree was tuned to generate nearly $m = 100$ leaf nodes' representatives

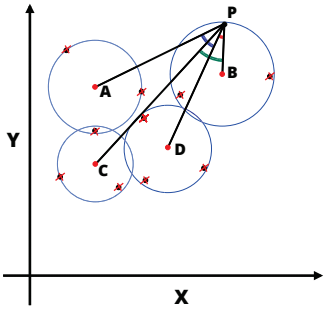
ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22741-8_14

Algorithm 1 The *MetricABOD* algorithm

Require: **Data** – input dataset;
Ensure: **Result** – pairs (**P**, **Score**) with the objects of **Data** sorted as per their corresponding outlierness scores;

```
Build a tree-based MAM for dataset Data. Let it be Tree;
Sample = ∅;
for each leaf node n of Tree do
    Let r be the representative object of n;
    Sample = Sample ∪ {r};
end for
Result = ∅;
for each object P in set Data do
    Angles = ∅;
    for each object x ≠ P in set Sample do
        for each object y ≠ x ≠ P in set Sample do
            Angle_(x,y) = the angle θ_(Px Py) between the difference vectors Px and Py;
            Angles = Angles ∪ {Angle_(x,y)};
        end for
    end for
    Score = the variance of angles in Angles;
    Result = Result ∪ {(P, Score)};
end for
Sort Result in descending order with regard to the Score values.
```

Fig. 5: Our proposed angle-based analysis for an object P. Circles in the 2-dimensional space illustrate the leaf nodes of any tree-based MAM. The new *MetricABOD* algorithm uses only the representatives A, B, C and D to compute the outlierness score of P.



in every experiment; note that it is the same value used for the related works' similar parameter k. As it was done for the other algorithms, we were particularly interested in evaluating *MetricABOD*'s ability to spot what the application specialists expect from their data in an automatic and timely manner.

The last column of Table 2 reports the results obtained with *MetricABOD*. Note that the tree-related costs are **included** in the runtime results. As it was expected, *MetricABOD* coped with low-to-high dimensional data being considerably faster than the original ABOD method; surprisingly, it even improved ABOD's accuracy for the datasets that we studied. In fact, our proposed algorithm made the analysis up to **thousands of times faster** when compared with

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22741-8_14

the 8 related works evaluated, still being in average **26**% **more accurate** than the most accurate related work, i.e., KNN-outlier. These results indicate that our *MetricABOD* is the best option, among those, which were considered, for use with a broad range of applications in which diversity takes place, especially when considering datasets that are as large as those required for commercial use.

To beter understand why *MetricABOD* outperformed ABOD in accuracy, we investigated the case with the highest discrepancy; that is, the analysis of the `Musk` dataset. Ground truth shows that there are 97 outliers in this data. ABOD and *MetricABOD* respectively identified 7% and 77% of them. There are 29 common objects among the 97 most outlying objects indicated by each technique, and only 7 of them are true outliers. Interestingly, the common objects are among ABOD's most outlying objects, while they figure among the least outlying ones for *MetricABOD*, thus further corroborating the superiority of our proposal's results for this dataset. Let us highlight that `Musk` has the largest dimensionality among all datasets studied, with 166 attributes. Due to this fact, we conjecture that the dimensionality was high enough to affect ABOD's results, even with its angle-based analysis that is less susceptible than distances to the curse of the high dimensionality. On the other hand, there are evidences in the literature [18] indicating that to spot appropriate representatives for a dataset is somehow similar to reducing its dimensionality. Thus, we believe that the tree nodes' representatives obtained from `Musk` reduced the effects of the high dimensionality, thus enabling the angle-based analysis to obtain better results.

5.2 Results: spotting outliers in adimensional data

This section describes additional experiments performed with two real collections of text data. We aim at showing that the new *MetricABOD* algorithm works also for adimensional, purely metric data. As we discussed before, *MetricABOD* must compute angles among pairs of objects (x, y) using other object P as the pivot. But, *how to compute angles in an adimensional data space?* To tackle the problem, we used a simple angle estimation strategy based on distances only: given x, y and P, the corresponding angle was computed from a triangle in the 2-dimensional space, whose side sizes are distances $d(x, y)$, $d(x, P)$ and $d(y, P)$.

The datasets studied are:

- **NSF abstracts:** the NSF Research Award Abstracts 1990-2003 from the UCI repository (archive.ics.uci.edu/ml/datasets/). It has 129,000 abstracts describing awards granted by the National Science Foundation for basic research from 1990 to 2003. The abstracts were represented as sets of steamed words, and compared using the Jaccard distance function;
- **Brazilian names:** a set of 2,755 first names used in Brazil. It is available at: https://gabrielrb.net/2011/10/18/dados-prontos-em-formato-sql-e-csv/. The L-Edit distance function was applied to compare the names.

We ran *MetricABOD* in both datasets, thus obtaining the corresponding outlierness scores. Since there is no ground truth for these data, accuracy was empirically evaluated by verifying whether or not the highest scores are in objects far away from the others. Figure 6 ranks the Brazilian names according to their average distances, on metric space, to all other names. Figure 7 ranks the NSF

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22741-8_14

abstracts in a similar way. In each illustration, orange vertical lines indicate the top-20 outliers. Five of these lines report the actual names or abstracts' NSF identifiers; they are the top-5 outliers. Let us emphasize that in most cases the outliers are far away from any other object; note the log scale in the horizontal axes. Finally, as Brazilian citizens, we also argue that the most outlying names are indeed very distinct from names of people that are regularly used in Brazil.

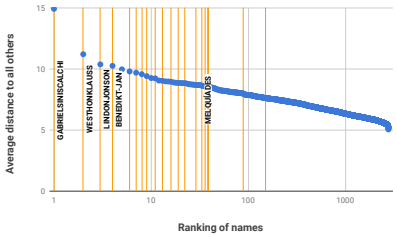


Fig. 6: Rank of Brazilian names as per their average L-Edit distances to all others. Orange lines are the top-20 outliers; lines with text are the top-5 ones.

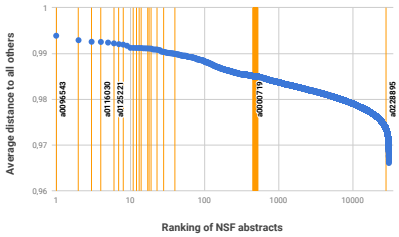


Fig. 7: Rank of NSF abstracts as per their average Jaccard distances to all others. Orange lines are the top-20 outliers; lines with text are the top-5 ones.

6 Conclusion

This paper described an empirical study performed on unsupervised outlier detection using 8 datasets that refer to a variety of real-world tasks of high impact, like spotting cyberattacks, clinical pathologies and abnormalities in nature, and 8 algorithms from the state-of-the-art. We presented the lowdown on the results obtained, pointing out to the strengths and weaknesses of each technique from the **application specialist's point of view**, which is a shift from the designer-based point of view that is commonly considered. Interestingly, many of the techniques had unfeasibly high runtime requirements or failed to spot what the specialists consider as outliers in their own data. To tackle this issue, we carefully designed *MetricABOD*: a novel ABOD-based algorithm that made the analysis up to **thousands of times faster**, still being in average **26**% **more accurate** than the most accurate related work. The main innovation is a new usage for tree-based Metric Access Methods that were originally designed to index complex data. This improvement is essential to enable outlier detection in many real-world applications for which the existing methods lead to unexpected results or unfeasible runtime requirements. Additionally, we studied two real collections of text data to show that our *MetricABOD* works also for adimensional, purely metric data.

Finally, we must highlight that all theoretical models existing for unsupervised outlier detection make assumptions that may not reflect the true nature of outliers in every single application [5, 1]. The results reported in our work corroborate this fact; they demonstrate how a given method can be very accurate

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22741-8_14

in specific cases and still fail in others, in the sense of returning or not what the application specialist expects from his/her data. Here, one must remember that specific data mining tasks all have biases that are not well-understood; they may be hindering some of the techniques that we studied. Due to this fact, supervised outlier detection can be handy in tasks where unsupervised methods do not perform acceptably. Further discussion can also be conducted as to the validity of the ground truth used and their representativeness as outliers. Nevertheless, a **few** exceptions apart, it is always desirable that outlier detection techniques return what the application specialists understand as outliers in their own data.

References

1. Aggarwal, C.C.: Outlier Analysis. Springer (2013)
2. Aggarwal, C.C., Yu, P.S.: Outlier detection for high dimensional data. SIGMOD Rec. **30**(2), 37–46 (2001)
3. Bay, S.D., Schwabacher, M.: Mining distance-based outliers in near linear time with randomization and a simple pruning rule. In: ACM SIGKDD. pp. 29–38 (2003)
4. Breunig, M.M., Kriegel, H.P., Ng, R.T., Sander, J.: LOF: Identifying density-based local outliers. SIGMOD Rec. **29**(2), 93–104 (2000)
5. Campos, G.O., Zimek, A., Sander, J., Campello, R.J.G.B., Micenková, B., Schubert, E., Assent, I., Houle, M.E.: On the evaluation of unsupervised outlier detection: measures, datasets, and an empirical study. DMKD **30**(4), 891–927 (2016)
6. Fan, J., Li, R.: Statistical challenges with high dimensionality: feature selection in knowledge discovery. In: Congress of Mathematicians. pp. 595–622 (2006)
7. Ghoting, A., Parthasarathy, S., Otey, M.E.: Fast mining of distance-based outliers in high-dimensional datasets. DMKD **16**(3), 349–364 (2008)
8. Johnstone, I.M., Titterington, D.M.: Statistical challenges of high-dimensional data. Philosophical Transactions A **367**(1906), 4237–4253 (2009)
9. Keller, F., Muller, E., Bohm, K.: HiCS: High contrast subspaces for density-based outlier ranking. In: IEEE ICDE. pp. 1037–1048 (2012)
10. Knorr, E.M., Ng, R.T.: Algorithms for mining distance-based outliers in large datasets. In: VLDB. pp. 392–403 (1998)
11. Kriegel, H.P., Schubert, M., Zimek, A.: Angle-based outlier detection in high-dimensional data. In: ACM SIGKDD. pp. 444–452 (2008)
12. Marques, H.O., Campello, R.J.G.B., Zimek, A., Sander, J.: On the internal evaluation of unsupervised outlier detection. In: SSDBM. pp. 7:1–7:12 (2015)
13. Muller, E., Assent, I., Steinhausen, U., Seidl, T.: OutRank: Ranking outliers in high dimensional data. In: IEEE ICDE Workshop. pp. 600–603 (2008)
14. Muller, E., Schiffer, M., Seidl, T.: Statistical selection of relevant subspace projections for outlier ranking. In: IEEE ICDE. pp. 434–445 (2011)
15. Papadimitriou, S., Kitagawa, H., Gibbons, P.B., Faloutsos, C.: LOCI: Fast outlier detection using the local correlation integral. In: IEEE ICDE. pp. 315–326 (2003)
16. Pham, N., Pagh, R.: A near-linear time approximation algorithm for angle-based outlier detection in high-dimensional data. In: ACM SIGKDD. pp. 877–885 (2012)
17. Traina Jr., C., Traina, A., Faloutsos, C., Seeger, B.: Fast indexing and visualization of metric data sets using slim-trees. IEEE TKDE **14**(2), 244–260 (2002)
18. Zimek, A., Schubert, E., Kriegel, H.P.: A survey on unsupervised outlier detection in high-dimensional numerical data. ASA Data Science **5**(5), 363–387 (2012)

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22741-8_14