Data Min Knowl Disc (2016) 30:891–927
DOI 10.1007/s10618-015-0444-8

On the evaluation of unsupervised outlier detection: measures, datasets, and an empirical study

Guilherme O. Campos[1] · Arthur Zimek[2] · Jörg Sander[3] · Ricardo J. G. B. Campello[1] · Barbora Micenková[4] · Erich Schubert[2] · Ira Assent[4] · Michael E. Houle[5]

Received: 17 July 2015 / Accepted: 18 November 2015 / Published online: 16 January 2016

Abstract The evaluation of unsupervised outlier detection algorithms is a constant challenge in data mining research. Little is known regarding the strengths and weaknesses of different standard outlier detection models, and the impact of parameter

Responsible editor: Johannes Fuernkranz.

✉ Arthur Zimek
zimek@dbs.ifi.lmu.de
http://www.dbs.ifi.lmu.de

Guilherme O. Campos
gocampos@icmc.usp.br

Jörg Sander
jsander@ualberta.ca

Ricardo J. G. B. Campello
campello@icmc.usp.br

Barbora Micenková
barbora@cs.au.dk
http://cs.au.dk

Erich Schubert
schube@dbs.ifi.lmu.de

Ira Assent
ira@cs.au.dk

Michael E. Houle
meh@nii.ac.jp

[1] University of São Paulo, SCC/ICMC/USP, C.P. 668, CEP 13566-590 São Carlos, SP, Brazil

[2] Ludwig-Maximilians-Universität München, Oettingenstr. 67, 80538 Munich, Germany

[3] Department of Computing Science, University of Alberta, Edmonton, AB T6G 2E8, Canada

[4] Department of Computer Science, Aarhus University, Aabogade 34, 8200 Aarhus, Denmark

[5] National Institute of Informatics, 2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo 101-8430, Japan

choices for these algorithms. The scarcity of appropriate benchmark datasets with ground truth annotation is a significant impediment to the evaluation of outlier methods. Even when labeled datasets are available, their suitability for the outlier detection task is typically unknown. Furthermore, the biases of commonly-used evaluation measures are not fully understood. It is thus difficult to ascertain the extent to which newly-proposed outlier detection methods improve over established methods. In this paper, we perform an extensive experimental study on the performance of a representative set of standard *k* nearest neighborhood-based methods for unsupervised outlier detection, across a wide variety of datasets prepared for this purpose. Based on the overall performance of the outlier detection methods, we provide a characterization of the datasets themselves, and discuss their suitability as outlier detection benchmark sets. We also examine the most commonly-used measures for comparing the performance of different methods, and suggest adaptations that are more suitable for the evaluation of outlier detection results.

Keywords Unsupervised outlier detection · Evaluation · Measures · Datasets

1 Introduction

An outlier, according to the well-known definition of Hawkins (1980), is “an observation which deviates so much from other observations as to arouse suspicions that it was generated by a different mechanism”, or according to Barnett and Lewis (1994), “an observation (or subset of observations) which appears to be inconsistent with the remainder of that set of data”. In the database community, the research on outlier detection algorithms was triggered by the seminal work of Knorr and Ng (1997). Since then, new models have continually been developed to characterize and identify outliers (Chandola et al. 2009; Zimek et al. 2012; Aggarwal 2013; Schubert et al. 2014b; Akoglu et al. 2015).

The efficiency and effectiveness of outlier detection is of much interest to researchers and practitioners. With regard to efficiency, a study by Orair et al. (2010) examines the basic techniques for accelerating a fundamental class of algorithms, the so-called ‘distance-based’ outlier detection models. The evaluation of efficiency can be problematic due to the dependency of performance on many different factors, such as the dataset size and dimensionality, the choice of parameters, the data structures employed, and other implementation details (Kriegel et al. 2015). Nevertheless, the efficiency of outlier detection is a well-studied topic in the literature. This is not our focus here.

On the other hand, the effectiveness of unsupervised algorithms for outlier detection has received far less attention than efficiency issues. One explanation for this lack of attention is the inherent difficulty of characterizing the nature of the candidates produced by a given outlier method. In terms of such aspects as accuracy, utility, or conciseness, any comparison between outlier methods is inappropriate when the natures of the outliers produced differ, or when their validity has not been established. As has already been pointed out by Schubert et al. (2014b) and Zimek et al. (2013a), the nature of ‘outlierness’ as modelled by different algorithms is not yet well understood.

In contrast, for other tasks of explorative data mining, the validity of results has been the topic of some discussion in the research literature—in the context of frequent pattern analysis, see the work of Vreeken and Tatti (2014), Zimmermann (2014), and Zimek and Vreeken (2015); and for clustering, that of Vendramin et al. (2010), Färber et al. (2010), and Kriegel et al. (2011b).

Another obstacle to the evaluation of outlier methods is the lack of commonly agreed-upon benchmark data. In a preliminary attempt, Emmott et al. (2013) gave a procedure for generating benchmark datasets according to a notion of 'classification hardness'. Their approach, although interesting and systematic, is not described in detail sufficient to allow the experiments to be reproduced, nor have the resulting datasets been made available.

Throughout the research literature, a common approach to the evaluation of outlier detection algorithms is to test their performance on classification datasets. With some datasets, semantic information may be associated with certain classes that suggests that they are sufficiently unusual and distinct from the remainder of the classes, so as to constitute a natural ground truth outlier object set. More commonly, when no such semantic information exists, a ground truth is generated by selecting as outliers a small proportion of data objects from a small number of classes, and as inliers the objects from the remaining classes (Zimek et al. 2013a). This downsampling-based strategy for generating datasets with annotated ground truth, previously pursued by Lazarevic and Kumar (2005), Abe et al. (2006), Yang et al. (2008), Zhang et al. (2009), Keller et al. (2012), Dang et al. (2013), Zimek et al. (2013b) and Dang et al. (2014) (among others), conforms with the notions of outlierness as characterized by Hawkins, and Barnett and Lewis, in that the outlier objects are selected from distributions (classes) that are different from those of the inlier objects. Moreover, the rarity of the outlier objects prevents them from forming a group that could be identified through standard techniques such as clustering. However, the generation cannot in principle be standardized, since varying the choice of classes from which outliers are drawn, or the choice of proportions of outliers actually selected, can lead to substantially different experimental outcomes.

In this paper, we present what is (to our knowledge) the first attempt of a broad evaluation of a variety of general-purpose models of outlierness, with respect to a collection of classification datasets previously used for the evaluation of outlier methods. We examine the major issues that arise in the conversion of classification datasets to outlier evaluation datasets. We review a number of outlier evaluation measures, and assess their characteristics and potential weaknesses. We also suggest modifications of these measures that are more suitable for the evaluation of outlier detection results when comparing performance over several datasets. Based on the aforementioned dataset collection, and using several evaluation measures, we perform a comparative analysis of a broad class of unsupervised outlier detection algorithms, namely the class of outlier models based on k-nearest neighborhoods. We focus on this class of outlier models for several reasons. The family of unsupervised models includes both very simple and more refined examples; the oldest and most well-known outlier methods such as kNN (Ramaswamy et al. 2000) and LOF (Breunig et al. 2000) belong here, as well as recent developments such as KDEOS (Schubert et al. 2014a). Although all methods studied here make use of k-nearest neighborhoods in one way or the other, the exact use and meaning of the k-nearest neighbors varies across the different models. Finally and most importantly, since all the methods considered share a parameterization in terms of a local neighborhood size k, the relative behavior of the algorithms can be compared across the various choices of this single parameter.

Although the evaluation of outlier methods is a motivation and an outcome of this study, *this* evaluation *per se* is not the main goal. Rather, we propose to study and discuss the issues surrounding *any* evaluation. Also, in this study we do not claim to provide a collection of datasets that can serve as a general-purpose benchmark for evaluation. Instead, we investigate the datasets and evaluation measures currently used in the existing practice of outlier evaluation, and demonstrate the need for the adoption of benchmarks by the outlier detection research community. We see as the main contributions of this work the study, characterization, and description of datasets in light of the outlier detection task, as well as a demonstration of the importance of generating ground truth data according to well-considered strategies, instead of by arbitrary selection. In the same spirit, our study does not merely evaluate and analyze existing methods for outlier detection—it also aims to provide guidance as to how to evaluate and analyze such methods. We also show that testing broad ranges of parameter values is crucial when evaluating outlier methods, in order to avoid misleading experimental outcomes.

We begin our study by describing and discussing our selection of algorithms (Sect. 2). We then introduce the evaluation measures that will be used, as well as our proposed modifications of these measures (Sect. 3). Next, we describe the selection and preprocessing of datasets (Sect. 4). We then summarize and discuss the results of experiments regarding the performance of the tested methods, and the suitability of the datasets (and their preprocessing procedures) for the outlier detection tasks (Sect. 5). We conclude the paper with a summary and discussion of our findings and of open issues for future work (Sect. 6).

2 Outlier detection methods

Following the seminal work on distance-based outlier detection by Knorr and Ng (1997, 1998), a plethora of unsupervised algorithms has been developed. They define outliers to be those objects for which a large proportion of the data lies beyond a fixed distance threshold. Later approaches do not determine a set of outliers according to an explicit distance threshold, but instead assign to each object a score reflecting its ‘outlierness’. Such variants of the distance-based approach are based on the distance to the kth-nearest neighbor (the ‘kNN distance’) (Ramaswamy et al. 2000), or on an aggregate of the distances to each of the k nearest neighbors (kNNs) (Angiulli and Pizzuti 2005). Another family of unsupervised methods, so-called ‘local’ approaches, has its origins in the Local Outlier Factor (LOF) (Breunig et al. 2000). In general, methods for unsupervised outlier detection can be categorized according to the manner in which locality is used to determine outliers (Schubert et al. 2014b). Outlier scoring can be regarded as a comparison of a ‘local model’ or context of a test object (typically a neighborhood set according to some distance measure) with that of a reference set. If the reference set consists of all (or most) of the objects in the dataset, the outlier detection strategy is considered to be ‘global’—since the reference set is fixed, points outscore others simply based on how their local model is rated in the outlier criterion as compared to all other points. On the other hand, if the reference set is of a scale similar to that of the context set, the strategy is considered to be ‘local’. The number of different reference sets considered would be high, allowing for the discovery of outliers throughout different regions of the data space.

2.1 Methods

In this evaluation study, we focus on the following outlier detection strategies, all of which require a user-supplied parameter k to fix the size of the neighborhood set (which will henceforth be referred to as the ‘kNN set’ or ‘kNNs’, and the distance to the kth nearest neighbor will be referred to as ‘kNN distance’). Since finding the kNNs in a data set of size $N = |DB|$ by means of a linear scan requires $O(N^2)$ distance computations (which can often be reduced to $O(N \log N)$ using appropriate index structures), the cost of finding the neighbors usually dominates the algorithm runtime (Schubert et al. 2015b). In the following, we will focus the discussion on the execution cost over and above that due to the computation of neighborhood sets.

As will be discussed, the following methods are representative of the main fundamental models of outlierness, and their variants.

kNN (Ramaswamy et al. 2000) determines outliers as those data objects having the largest kNN distance. The intuition is close to the proposal of Knorr and Ng (1997, 1998) of using a density estimate to find low-density points. In addition to the cost of finding the kNN sets, this only requires $O(1)$ computation per point.

kNN-weight (kNNW) (Angiulli and Pizzuti 2002, 2005) uses the sum of distances to an object’s kNNs to reduce variation in scores and make the score less sensitive to a change of the parameter k. Again, a large sum of distances indicates a low density and outliers are those objects with the largest score. Computing the sum requires $O(k)$ additional operations per point.

ODIN (Outlier Detection using Indegree Number) (Hautamäki et al. 2004) defines outlierness as a low number of in-adjacent edges in the kNN graph—or equivalently, a low hubness value, defined as the cardinality of the reverse kNN set, (the ‘RkNNs’) (Radovanović et al. 2014). Accumulating the counts of occurrences requires $O(k)$ operations per data point.

ODIN, kNN, and kNNW can be seen as a family of ‘global’ methods, making direct use of the kNN set of each point. The other large family of methods is derived from LOF.

LOF (Local Outlier Factor) (Breunig et al. 2000) was the first local outlier method. It compares the local reachability density (lrd) of the kNN set of the test point, with those of the neighborhoods of each member of the kNN set: a high value indicates outlierness. Local reachability density is defined as the inverse average reachability distance from the neighbors

$$\operatorname{lrd}(p) := 1 \Bigg/ \frac{\sum_{o \in k\mathrm{NN}(p)} \text{reach-dist}_k(p \leftarrow o)}{|k\mathrm{NN}(p)|}, \tag{1}$$

where the (asymmetric) reachability distance in turn is defined as:

$$\text{reach-dist}_k(p \leftarrow o) = \max\{k\text{-dist}(o), d(p, o)\}. \tag{2}$$

Intuitively, the local reachability density (lrd) is a density estimate with additional smoothing to reduce variability.

The final LOF score then compares the locally relevant lrd values:

$$\text{LOF}_k(p) = \frac{1}{|k\text{NN}(p)|} \sum_{o \in k\text{NN}(p)} \frac{\text{lrd}_k(o)}{\text{lrd}_k(p)} \tag{3}$$

The LOF score achieves its highest values when the local density estimate (lrd) of the test point is small relative to the estimates of its nearest neighbors. If we can afford to store the kNN and lrd values of all data points simultaneously, computing the LOF scores requires $O(k)$ additional operations per point (in addition to those involved in the computation of all kNN sets).

LOF was a seminal work inspiring many variants, several of which are included in this study. For a detailed theoretical analysis of LOF and its variants, see the work of Schubert et al. (2014a, b).

SimplifiedLOF (Schubert et al. 2014b) replaces LOF's reachability distance (in Eq. 1) by the kNN distance, resulting in a simpler density estimate:[1]

$$\text{dens}(p) = \frac{1}{k\text{-dist}(p)} \tag{4}$$

This density estimate arises from LOF when in Eq. 2 one substitutes k-dist(p) for k-dist(o): since $o \in k\text{NN}(p)$, $d(p, o) \leq k\text{-dist}(p)$ by definition of the k nearest neighbors, from which Eq. 1 can be seen to simplify to Eq. 4. The complexity of SimplifiedLOF is similar to LOF.

COF (Connectivity-based Outlier Factor) (Tang et al. 2002) modifies the density estimation of SimplifiedLOF to account for the 'connectedness' of a neighborhood via a minimum spanning tree (MST) rooted at the point under study. The MST can be 'shelled' or reduced back to the root p by progressively deleting the leaf node incident to the edge having greatest length. Computing the total edge lengths for each of the MSTs in the shelling, and then averaging over the MSTs, yields the 'average chaining distance' for the kNN set of p:

$$\text{ac-dist}(p) = \sum_{i=1}^{k} \frac{2(k+1-i)}{k(k+1)} \lambda(e_i), \tag{5}$$

where e_{k-i} is the edge deleted in the ith shelling step, and $\lambda(e_{k-i})$ is its length. The COF value of p is then given as a ratio of the ac-dist of p with the average of the ac-dist values of its k-nearest neighbors:

[1] While only recently defined formally, SimplifiedLOF has been implicitly used (and adapted), often presumably unintentionally [i.e., not being aware of the special definition of the reachability distance (Eq. 2)], in many earlier variants of LOF. Here, for the first time, it is evaluated explicitly.

$$\mathrm{COF}(p) = \frac{\text{ac-dist}(p)}{\frac{1}{k}\sum_{o\in k\mathrm{NN}(p)} \text{ac-dist}(o)} \tag{6}$$

Computing the spanning tree of the kNNs requires the computation of pairwise distances, and thus the cost is in $O(k^2)$.

INFLO (Influenced Outlierness) (Jin et al. 2006) compares the local model of SimplifiedLOF with the same density estimate applied to the reference set of the union of kNN and RkNN sets. INFLO is thus an example of a local outlier detection strategy for which different definitions of neighborhood are used for the context set and reference set. Computing the RkNNs can be performed in a list-inversion step, which requires $O(k)$ operations.

LoOP (Local Outlier Probabilities) (Kriegel et al. 2009a) uses a more robust local density estimate than SimplifiedLOF based on the quadratic mean distance:

$$\text{LoOP-dens}(p) = 1 \Bigg/ \sqrt{\frac{1}{|k\mathrm{NN}(p)|}\sum_{o\in k\mathrm{NN}(p)} d(o,p)^2} \tag{7}$$

Furthermore, LoOP also normalizes the outlier detection scores. The compexity is the same as SimplifiedLOF, i.e. $O(k)$ per point.

LDOF (Local Distance-based Outlier Factor) (Zhang et al. 2009) defines outlierness as the ratio of the average of the distances from the test point to its kNNs, and of the average pairwise distances within the kNN set. LDOF favors outliers that are far from a kNN set which is itself compact. As with COF, LDOF requires that all pairwise distances be computed ($O(k^2)$).

LDF (Local Density Factor) (Latecki et al. 2007) replaces LOF's density estimate by a variable-width Gaussian kernel density estimation (KDE) modified to use LOF's reachability distance; the resulting estimator is no longer a kernel density in the mathematical sense. In the KDE formulation, the original (Euclidean) distance is replaced with the reachability distance of LOF (Eq. 2) to yield a local density estimation function LDE:

$$\mathrm{LDE}_d(p) := \frac{1}{|k\mathrm{NN}(p)|}\sum_{o\in k\mathrm{NN}(p)} \frac{1}{(2\pi)^{d/2}(h\cdot k\text{-dist}(p))^d} e^{-\frac{1}{2}\frac{\text{reach-dist}(p,o)^2}{(h\cdot k\text{-dist}(p))^2}} \tag{8}$$

The LDE densities are then compared to neighbor estimates in a similar fashion

$$\mathrm{LDF}(p) = \frac{\frac{1}{k}\sum_{o\in k\mathrm{NN}(p)} \mathrm{LDE}(o)}{\mathrm{LDE}(p) + \frac{c}{k}\sum_{o\in k\mathrm{NN}(p)} \mathrm{LDE}(o)} \tag{9}$$

The complexity of LDF is the same as that of LOF.

KDEOS (Kernel Density Estimation Outlier Score) (Schubert et al. 2014a) also incorporates KDE into the LOF framework, but unlike LDE, it keeps the mathematical kernel density estimation intact. For comparison with neighbor densities, the KDE densities are standardized per point as z-scores with respect to the KDE densities of the kNN set, and averaged over different neighborhood sizes $k_{\min} \ldots k_{\max}$:

$$\mathrm{s}(p) = \underset{k_{\min} \ldots k_{\max}}{mean} \text{ z-score} \left(KDE_k(p), \{KDE_k(o)\}_{o \in k\mathrm{NN}(p)} \right) \tag{10}$$

Treating the resulting interim score s as normally distributed, the normal cumulative density function is used to obtain the final KDEOS score. For our study, we set $k_{\min} = k_{\max} = k$ for KDEOS, for ease of comparison with other outlier methods. The complexity of KDEOS is $O(k \cdot (k_{\max} - k_{\min} + 1))$, which is in general quadratic; however, in our formulation, the cost reduces to $O(k)$.

For a detailed theoretical analysis of both the kNN and the LOF families of density based outlier detection, see the work of Schubert et al. (2014a, b).

FastABOD (Fast Angle-Based Outlier Detection) was designed as a fast variant of angle-based outlier detection (ABOD) (Kriegel et al. 2008). The ABOD model computes, for each point as origin, the angles to all other pairs of points, and uses a weighted variance of these angles as an outlier factor—the complete computation requires $O(N^3)$ time. The more efficient FastABOD version considers only those pairs of points that are among the kNNs, and thus requires only $O(k^2)$ computations per point.

Both FastABOD and KDEOS have other parameters to tune for optimal results in addition to the neighborhood size k. FastABOD uses the kernel trick, and thus an appropriate kernel function must be chosen (we use the default polynomial kernel of degree 2). KDEOS performance benefits from choosing an intrinsic dimensionality and a kernel bandwidth multiplier h. LDF has a comparable kernel bandwidth multiplier constant h, and an additional score scaling constant c. We do not vary these default settings (we use $h = 1$, $c = 0.1$), though, as we evaluate all algorithms on the same terms, varying only the neighborhood size. For many datasets, there may be better results obtainable with either algorithm by further exploring the parameter space.

As we can see, all the methods presented above have a comparable runtime complexity, which is dominated by the database size $N = |DB|$: in addition to finding the kNN, all methods need $O(1)$, $O(k)$ or $O(k^2)$ computations per point, yielding an overall runtime of $O(N^2 + Nk^j)$ (for $j = 0 \ldots 2$, depending on the method as discussed above). If the dataset is amenable to indexing, this can be reduced to $O(N \log N + Nk^j)$. Assuming that only small neighborhoods of size $k^2 < \log N$ are of interest to the user, the complexity is that of k nearest neighbor search.

Efficiency issues have lately been addressed through the use of approximate neighborhood search (Wang et al. 2011; de Vries et al. 2010, 2012; Schubert et al. 2015b). In this study, however, we will focus on effectiveness, since for these fundamental methods, beyond the asymptotic complexities given above, an empirical study of efficiency is entirely dependent upon implementation details (Kriegel et al. 2015). For this reason, the efficiency of outlier methods is beyond the scope of our study.

2.2 Discussion

As stated above, we focus on a class of outlier detection models that are all based on k-nearest neighborhoods, for three main reasons: (i) the popularity of this very broad family of methods; (ii) the behavior of the algorithms can be compared across the various choices of this single parameter k; and (iii) the parameter k nevertheless has different interpretations and impact across the different models.

Within the family of kNN outlier detection methods, we selected the representative algorithms as listed above so as to have proponents of both the 'global' and 'local' paradigms, with different degrees of 'locality' (Schubert et al. 2014b). Second, we include variants of LOF for different choices of the typical components (Schubert et al. 2014b) of local outlier models: the notions of neighborhood, of (reachability) distance, of density estimate, of model comparison, and of score normalization. While they have been studied selectively in existing papers on specific datasets, ours is the first empirical study that evaluates them jointly and extensively on a large range of settings and datasets.

For these different variants, the impact of the choice of k is expected to depend on the role of locality of neighborhoods in the associated outlier model.

- The most straightforward connection between the choice of k and the outlier model is realized in kNN and kNNW.
- In the case of LOF and other 'local' methods, the method becomes more and more 'global' with increasing k, and density estimates become less local (and hence less discriminative)—for $k \approx N$, the 'local' model will be (almost) identical for all points.
- For FastABOD, k is an approximation parameter for speed-up, but effectively also results in a localization of the global ABOD model (i.e., the *context* is more local with smaller k). However, the with decreasing k increasingly local FastABOD model is still compared against *all* local models (i.e., the *reference* remains global). As the ABOD model is based on variance, we can expect FastABOD to deteriorate strongly with very small values of k, due to the problems of estimation over small sample sizes.
- For the kernel-based methods (LDF, KDEOS), the impact of k will again be different than for the more basic LOF variants.
- For INFLO, k constrains both the size of the reverse k-nearest neighborhood as well as the more usual k-nearest neighborhood.

In summary, although the parameter k determines the size of neighborhoods for all methods, the impact of the choice of k is expected to be different for different methods.

We have implemented all the aforementioned outlier methods in a common framework, ELKI (Achtert et al. 2013), and made the implementations publicly available.

3 Evaluation measures

The methods studied here return a full ranking of database objects based on their outlierness, scored according to the outlier model upon which the method is based. In practice, however, the user of an outlier detection method is generally interested in obtaining a manageably-small subset consisting of the top-ranked objects of the result. For those cases where a target number of outlier candidates n is specified in advance, the simplest measure of performance is the precision at n (which we denote by $P@n$), defined as the proportion of correct results in the top n ranks (Craswell 2009a). For a database DB of size N, consisting of outliers $O \subset DB$ and inliers $I \subseteq DB$ ($DB = O \cup I$), $P@n$ can be formalized as:

$$P@n = \frac{|\{o \in O \mid \text{rank}(o) \leq n\}|}{n}. \tag{11}$$

Here, we assume that the outlier ranking is unique; if instead two or more objects share a common outlierness score, we assume that the ties are broken arbitrarily but consistently.

When using $P@n$ to assess the general performance of an outlier ranking method, it is unclear how to fairly choose the parameter n. Setting it to the number of outliers in the ground truth, $n = |O|$, yields the popular R-Precision measure (Craswell 2009b). Whenever the number of outliers $n = |O|$ is very low relative to N, the values of $P@n$ that are typically obtained can be deceptively low, and not very informative as such. On the other hand, when $n = |O|$ is relatively large (of the same order as N), deceptively high values of $P@n$ can be obtained simply due to the relatively small number of inliers available. To make results with different outlier rates easier to compare, we suggest that the precision at n should be adjusted for chance.

Adjustment for chance is a principled procedure that was discussed by Hubert and Arabie (1985) for the validation of clusterings, but which can be applied to any index as follows:

$$\text{Adjusted Index} = \frac{\text{Index} - \text{Expected Index}}{\text{Maximum Index} - \text{Expected Index}} \tag{12}$$

The main purpose of adjustment for chance is to allow for comparisons between different settings where the expected values of the score (in our case, outlierness) may vary, by bringing scores that are close to their expected values into alignment.

For the index $P@n$, the maximum possible value is $|O|/n$ if $n > |O|$, and 1 otherwise. The expected value under the hypothesis of a completely random outlier ranking is $|O|/N$, which does not depend on n.[2] If $n \leq |O|$, we obtain the adjustment formula

$$\text{Adjusted } P@n = \frac{P@n - |O|/N}{1 - |O|/N}. \tag{13}$$

For larger n, the maximum $|O|/n$ must be used instead of 1 in Eq. 13.

An important challenge in both the design and evaluation of outlier methods is that of coping with the inherent imbalance between the numbers of inliers and outliers: we generally expect that $|I| \gg |O|$, and that $|I| \approx N$. While the $P@n$ and Adjusted $P@n$

[2] In fact, the number of true outliers expected to be ranked by chance among the top n positions is a fraction n/N of $|O|$, which yields $P@n = \frac{n \cdot |O|}{N} \big/ n = \frac{|O|}{N}$.

measures are easily interpreted, they are sensitive to the choice of n, particularly when n is small. For example, for a dataset with 10 outliers and 1 million inliers, an algorithm that assigns the true outliers to the (quite high) ranks 11–20 will nevertheless have a $P@10$ of 0, but a $P@20$ of 0.5. The Adjusted $P@n$ measure can be seen to suffer from a similar sensitivity with respect to the choice of n. The remaining outlier measures to be discussed in this section avoid such pitfalls by averaging across different values of n.

The use of the R-Precision ($P@|O|$) evaluation measure implicitly assumes that the user has some knowledge of the number of outliers in the dataset. However, this is not always the case, and thus attention has been given to measures that aggregate performance over a wide range of possible choices of n. One such measure, popular in information retrieval contexts, is the average precision (Zhang and Zhang 2009):

$$\mathrm{AP} = \frac{1}{|O|} \sum_{o \in O} P@\,\mathrm{rank}(o). \tag{14}$$

Instead of evaluating only at a single value of n, the values of $P@n$ are averaged over the ranks of all outlier objects $o \in |O|$. There exist other variants, such as evaluating over a sample of the ranks at fixed intervals. However, evaluation at positive (outlier) ranks appears to be both popular and well-suited for imbalanced datasets.

Average precision can be adjusted the same way as $P@n$. A perfect ranking yields a maximum value of 1, and the expected value of a random ranking is $|O|/N$:

$$\text{Adjusted AP} = \frac{\mathrm{AP} - |O|/N}{1 - |O|/N}. \tag{15}$$

For precision at n and for average precision, the adjustment for chance is not strictly necessary when the performance of two methods on the same dataset (that is, with the same proportion of outliers) are compared in relative terms. Adjustment for chance is helpful if the measure is to be interpreted in absolute terms. Adjustment for chance is strictly necessary if the performance is to be compared over different datasets with different proportions of outliers—for example, when studying the susceptibility of the method to the proportion of outliers. Such comparisons can be misleading when the measures are not adjusted.

The most popular evaluation measure in the literature on unsupervised outlier detection is based on a curve known as the Receiver Operating Characteristic (ROC), due to its origin in signal detection. The curve is obtained by plotting for all possible choices of n the true positive rate (the proportion of outliers correctly ranked among the top n) versus the false positive rate (the proportion of inliers ranked among the top n). A random outlier ranking would result in a curve close to the diagonal, whereas a perfect ranking (in which all outliers are ranked ahead of any inliers) would result in a curve consisting of a vertical line at false positive rate 0 and a horizontal line at the top of the plot (indicating a true positive rate of 1 for every false positive rate > 0).

Since the false positive rate is normalized by the number of false positives (the number of inliers), and the true positive rate is normalized by the number of positives (the number of outliers), ROC inherently adjusts for the imbalance of class sizes typical of outlier detection tasks.

A ROC curve can be summarized by a single value known as ROC AUC, defined as the area under the ROC curve (AUC). The ROC AUC value ranges between 0 and 1, and can be regarded as the average of the recall at n (true positive rate over the n top-ranked objects), with n taken over the ranks of all inlier objects in $|I|$. A perfect ranking of the database objects would result in a ROC AUC value of 1, whereas an inverted perfect ranking would produce a value approaching 0. A random ranking of the database objects would result in a ROC AUC value close to 0.5. Hanley and McNeil (1982) showed that the ROC AUC value corresponds to the probability of a pair (o, i), where o is some true outlier, and i is some inlier, being ordered correctly in the evaluated ranking (that is, with o appearing before i).

$$\text{ROC AUC} := \underset{o \in O, i \in I}{mean} \begin{cases} 1 & \text{if } \operatorname{score}(o) > \operatorname{score}(i) \\ \frac{1}{2} & \text{if } \operatorname{score}(o) = \operatorname{score}(i) \\ 0 & \text{if } \operatorname{score}(o) < \operatorname{score}(i) \end{cases} \tag{16}$$

The ROC AUC measure has therefore a straightforward probabilistic interpretation, and does not require adjustment for chance.

While ROC AUC is the most popular evaluation measure for unsupervised outlier detection methods that deliver a ranking, in this study we show its results alongside with the results from the other aforementioned measures, as they can highlight different aspects of the performance that one may wish to identify.

Note that all evaluation measures used in this study require external ground truth (i.e., labels identifying outliers vs. inliers). There is only one internal evaluation measure for outliers available in the literature so far, IREOS (Marques et al. 2015), which is, however, computationally very expensive.

4 Datasets

In this section, we give a systematic presentation of the datasets considered in this study on issues in outlier evaluation. The datasets have been organized into two groups: the first, presented in Sect. 4.2, consists of sets that have previously appeared in the research literature in the evaluation of outlier detection algorithms; the second group, presented in Sect. 4.3, consists of sets originally intended for the evaluation of classification methods, where one or more classes have a natural semantic interpretation as outliers. In total, we define, collect, and make publicly available a repository of approximately 1000 datasets (including variants), together with full details of their preprocessing.[3] We begin in Sect. 4.1 with a discussion of the issues surrounding the compilation, preparation, and description of datasets for the evaluation of outlier detection algorithms.

[3] http://www.dbs.ifi.lmu.de/research/outlier-evaluation/.

4.1 Dataset preparation

The UCI repository (Bache and Lichman 2013) is a valuable source of datasets for the evaluation of data mining algorithms. While most of them have been proposed for the evaluation of classification methods, they have also been widely used for unsupervised algorithms such as clustering. However, since the semantics of data clusters are often quite different from those of ground-truth classes, the appropriateness of such datasets for the evaluation of unsupervised learning methods is debatable (Färber et al. 2010). For the evaluation of outlier detection, the semantic mismatch is even more problematic, since outliers are assumed to be both rare and diverse.

In the following, we outline the main issues in converting classification datasets to outlier evaluation datasets, and discuss how these issues were handled in our study.

Downsampling A common approach in outlier detection research is to randomly downsample a particular class to produce outliers, while retaining all instances of the remaining classes to form the inlier set. Random downsampling often leads to great variation in the nature of the outliers produced. Therefore, to mitigate the impact of randomization when downsampling, we repeat the procedure for each dataset 10 times, resulting in 10 different variants for these datasets.

Duplicates The handling of duplicate instances in the dataset has received scant attention in the literature. However, the presence of duplicates is problematic for several methods. For example, for LOF and many of its variants, duplicate instances can lead to distance values of zero, which introduces numerical instability into the computation of local density estimates. For datasets containing duplicates, we generate two variants, one with the original duplicates, and one without duplicates. It should be noted, though, that removing duplicate records can drastically lower local density estimates in certain cases.

Categorical attributes Transformation of categorical attributes into numerical ones is another source of dataset variation. We employ two techniques:

- 1-of-n encoding, where a categorical attribute with n possible values is mapped into n binary attributes for which a value of 1 (or 0) represents the presence (or absence) of the corresponding categorical value;
- IDF, where a categorical attribute is encoded as the *inverse document frequency* $IDF(t) = \ln(N/f_t)$, where N is the total number of instances, and f_t is the frequency (number of occurrences) of the attribute value t.

For datasets with categorical attributes we thus have three variants: one where categorical attributes are removed, and two resulting from the transformations described above.

Normalization The normalization of datasets is expected to have considerable impact on the results, but is rarely discussed in the literature. A full exploration of this issue is beyond the scope of this study. However, for each dataset that does not already have normalized attributes, we include two variants: unnormalized, and attribute-wise linear normalization to the range [0, 1].

Missing values Standard outlier detection techniques as considered here cannot handle data with missing values. We determine for each dataset and each attribute the number of missing values. If an attribute has fewer than 10 % of instances with missing values, those instances are removed. Otherwise, the attribute itself is removed.

4.2 Datasets used in the literature

Table 1 lists those datasets of our study that are known to have appeared in the outlier detection literature. For each dataset, variants have been produced according to the guidelines set out in Sect. 4.1. The details shown in the table include the number of instances, outliers, and attributes after preprocessing missing attributes and downsampling but before the removal of duplicates. Full documentation of the datasets is available on our repository website.

Some of the datasets in our collection have been the basis of benchmarking in several publications. Unless the processed datasets were made publicly available [as it is the case for datasets used by Keller et al. (2012)[4]], some ambiguity may remain as to their construction, due to a lack of information as regards the issues listed in Sect. 4.1, or to the use of downsampling. As an additional complication, some publications do not give clear references to the datasets they use, or refer to datasets by ambiguous names. Often, it is not specified as to whether only the training set, only the test set, or both partitions of a classification dataset were used.[5] In contrast, processing datasets according to the guidelines set out earlier, as well as making them publicly available together with their full documentation, promotes the reproducibility and ease of comparison of future experimentation with outlier methods.

4.3 Semantically meaningful outlier datasets

Semantically meaningful datasets for outlier evaluation are those in which certain classes can be reasonably assumed to be associated with real-world instances that are both rare and deviating—for example, ‘sick’ patients within a population dominated by ‘healthy’ individuals. However, it is sometimes the case that outliers within the real-world population may be overrepresented within a given classification dataset. For such datasets, we create several variants by downsampling the outlier class at several different rates: 20, 10, 5, and 2 % of outliers.

As semantically meaningful datasets, we selected and processed the following UCI repository datasets (Bache and Lichman 2013), together with an additional classification dataset, Stamps (Micenková et al. 2012), combining training and test sets whenever both exist (summarized in Table 2).

- *Annthyroid* Medical data on hypothyroidism. Three classes relate to the conditions ‘normal’, ‘hyperfunction’, and ‘subnormal functioning’. Classes other than ‘normal’ were defined as outliers here.

[4] Available at: http://www.ipd.kit.edu/~muellere/HiCS/realworld.zip. Note that we have supplemented our collection with some of these datasets, without further preprocessing.

[5] For unsupervised learning, both training and test sets can be used together, and we assume this is the case unless otherwise specified.

Table 1 Datasets used in the literature, with different variants of preprocessing and of outlier definitions

Dataset	Preprocessing	N	|O|	Attributes		Version used by
				Num	Cat	
ALOI	50,000 images, 27 attr.	50,000	1508	27		Kriegel et al. (2011a) and Schubert et al. (2012)
	24,000 images, 27,648 attr.					de Vries et al. (2012)
Glass	Class 6 (*out.*) vs. others (*in.*)	214	9	7		Keller et al. (2012)
Ionosphere	Class ‘b’ (*out.*) vs. class ‘g’ (*in.*)	351	126	32		Keller et al. (2012)
KDDCup99	U2R (*out.*) vs. Normal (*in.*)	60,632	246	38	3	Nguyen and Gopalkrishnan (2010), Nguyen et al. (2010), Kriegel et al. (2011a) and Schubert et al. (2012)
Lymphography	Classes 1 and 4 (*out.*) vs. others (*in.*)	148	6	3	16	Lazarevic and Kumar (2005), Nguyen et al. (2010) and Zimek et al. (2013b)
PenDigits	Downsampling class ‘4’ to 20 objects (*out.*)	9868	20	16		Kriegel et al. (2011a) and Schubert et al. (2012)
	Downsampling class ‘0’ to 10 % (*out.*)					Keller et al. (2012)

Table 1 continued

Dataset	Preprocessing	*N*	|O|	Attributes		Version used by
				Num	*Cat*	
Shuttle	Classes 2, 3, 5, 6, 7 (*out.*) vs. class 1 (*in.*)					Lazarevic and Kumar (2005), Abe et al. (2006) and Nguyen et al. (2010)
	Class 2 (*out.*) vs. downs. others to 1000 obj. (*in.*)	1013	13	9		Zhang et al. (2009)
	Downs. classes 2, 3, 5, 6, 7 (*out.*) vs. others (*in.*)					Gao and Tan (2006)
Waveform	Downsampling class '0' to 100 objects (*out.*)	3443	100	21		Zimek et al. (2013b)
WBC	'*malignant*' (*out.*) vs. '*benign*' (*in.*)					Gao and Tan (2006)
	Downs. class '*malignant*' to 10 objects (*out.*)	454	10	9		Kriegel et al. (2011a), Schubert et al. (2012) and Zimek et al. (2013b)
WDBC	Downs. class '*malignant*' to 10 objects (*out.*)	367	10	30		Zhang et al. (2009)
	'*malignant*' (*out.*) vs. '*benign*' (*in.*)					Keller et al. (2012)
WPBC	Class 'R' (*out.*) vs. class 'N' (*in.*)	198	47	33		Keller et al. (2012)

For those variants used in our collection, we detail also the numbers of instances, outliers, and attributes

Table 2 Datasets with semantic outliers (numbers without downsampling)

Dataset	Semantics	N	$\|O\|$	Attributes	
				Num.	Binary
Annthyroid	Two types of hypothyroidism vs. healthy	7200	534	21	
Arrhythmia	12 Types of cardiac arrhythmia vs. healthy	450	206	259	
Cardiotocography	Pathologic, suspect vs. healthy	2126	471	21	
HeartDisease	Heart problems vs. healthy	270	120	13	
Hepatitis	Survival vs. fatal	80	13	19	
InternetAds	Ads vs. other images	3264	454		1555
PageBlocks	Non-text vs. text	5473	560	10	
Parkinson	Healthy vs. Parkinson	195	147	22	
Pima	Diabetes vs. healthy	768	268	8	
SpamBase	Non-spam vs. spam	4601	1813	57	
Stamps	Genuine vs. forged	340	31	9	
Wilt	Diseased trees vs. other	4839	261	5	

- *Arrhythmia* Patients classified as normal or as exhibiting some type of cardiac arrhythmia. In total, there are 14 types of arrhythmia and 1 type that brings together all the other different types. However, 3 types of arrhythmia have no data. Again, we treat healthy people as inliers and patients suffering from arrhythmia as outliers.
- *Pima* Medical data on diabetes. Patients suffering from diabetes were considered outliers.
- *Cardiotocography* Related to heart diseases, describing 3 classes: normal, suspect, or pathological. Normal patients are treated as inliers and the remaining as outliers.
- *Heart Disease* Medical data on heart problems, with affected patients considered outliers and healthy people considered inliers.
- *Hepatitis* Prediction of whether a patient suffering from hepatitis will die (outliers) or survive (inliers).
- *Internet Ads* Images from web pages, classified as ads or not, for the purpose of learning to remove ads automatically from web pages while retaining regular images. Ads are considered outliers.
- *Page Blocks* Types of blocks in document pages, relating to an essential step in document analysis, namely to separate text from pictures or graphics. If the block content is text, it was labeled here as inlier, otherwise it was labeled as outlier.
- *Parkinson* Medical data distinguishing healthy people from those suffering from Parkinson's disease. The latter were labeled as outliers.
- *Spam Base* Emails classified as spam (outliers) or non-spam.
- *Stamps* Learning to identify forged (photocopied and printed) stamps from genuine (ink) stamps based on color and printer properties. The former are outliers.

- *Wilt* Differentiating diseased trees from other land covers. The former are considered outliers here.

5 Experimental results

Our study centers around the following three questions: Are the evaluation measures capable of revealing the performance characteristics of outlier methods? How do outlier methods perform over a broad range of datasets and parameter settings? How can one better understand and properly characterize datasets in light of the outlier detection evaluation task?

In our experimentation, each of the 12 methods was executed over multiple variants of the 23 datasets, determined according to whether the data were normalized, whether duplicates were removed, on the treatment of categorical attributes, and on which of 4 downsampling rates was employed. Each experiment was performed for each meaningful choice of parameter value k between 1 and 100 (or the number of data instances, if less than 100).[6] In total, 1,300,758 experimental runs were performed. The complete results, including all plots for all datasets, are available on our repository website.[7]

5.1 Evaluation measures

In order to assess the quality of outlier methods, we first investigate the behavior of the three base evaluation measures identified in Sect. 3: precision at n [$P@n$, where we set $n = |O|$ (Craswell 2009a, b)], average precision (AP), and ROC AUC. For each evaluation measure, roughly 1000 plots were produced across the range of values of k, of which examples are shown in Fig. 1 for two datasets, Annthyroid (7.4 % outliers) and HeartDisease (44.4 % outliers). On these two examples, we can already observe significant variations in performance trends across differing combinations of outlier algorithms, datasets, parameter choices and evaluation methods (please see the web repository for the complete results).

As noted in Sect. 3, ROC AUC is expected to be less sensitive to variation in the number of true outliers than the other evaluation measures. This tendency is confirmed by our experimental results, as demonstrated by the examples shown in Fig. 1. The ROC AUC scores achieved by the methods were consistently high across the datasets, while still being able to discriminate among the different outlier methods. In contrast, the $P@n$ scores were considerably lower for those datasets with smaller proportions of outliers (as seen here for Annthyroid). Although the behavior of AP more closely resembles that of ROC AUC, in that it assesses the ranks of all outliers, AP scores also tend to be low when, as it is the typical case in outlier detection scenarios, the numbers of inliers and outliers are not balanced.

[6] FastABOD requires at least a set of 3 neighbors, as it computes variances of angles to neighbors. LDOF, KDEOS, and ODIN require at least 2 neighbors.

[7] http://www.dbs.ifi.lmu.de/research/outlier-evaluation/.

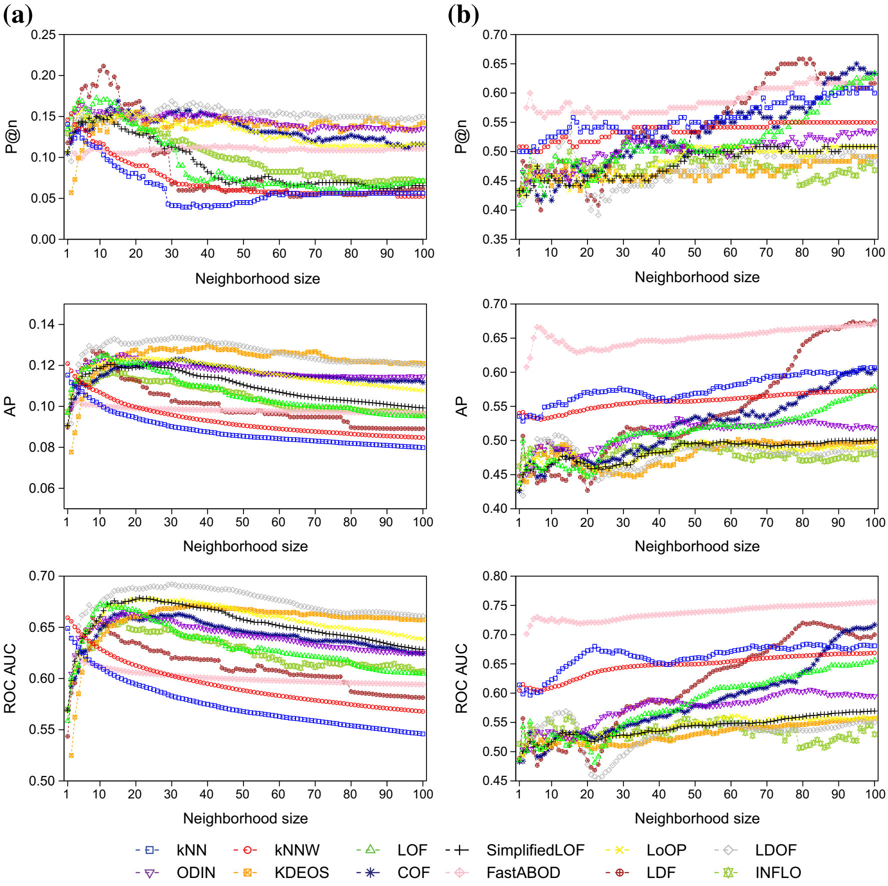

Fig. 1 Results on two datasets (examples, without duplicates, normalized, no downsampling), comparing precision at *n* (*P@n*), average precision (AP), and ROC AUC (complete results for all datasets are available on our web repository). **a** Annthyroid. **b** HeartDisease

P@n takes into account only the number of true outliers among the *n* top-ranked items, and thus its behavior is quite different from that of ROC AUC. As observed by Davis and Goadrich (2006), the *P@n* measure can therefore be helpful in discriminating between methods that perform more or less equally well in terms of ROC AUC. Following their lead, we will rely mainly on ROC AUC scores in judging the effectiveness of the outlier methods, while turning occasionally to *P@n* and AP for further insights.

When *P@n* and AP are adjusted for chance, we obtain curves analogous in shape to those of their unadjusted counterparts, but with an upward shift in the quality scores. As detailed in Sect. 3, although adjustment for chance is not necessary when comparing the performance of different methods on a single dataset, it is beneficial (if not indispensable) when comparing across datasets with different proportions of outliers.

5.2 Characterization of the outlier methods

In this section, we characterize the performance of the selected outlier detection methods using datasets introduced in Sect. 4. These datasets cover a wide range of proportions of points considered outliers in the ground truth of the datasets—as high as 75 % for the Parkinson dataset (without downsampling). One can argue that ground-truth datasets with a large proportion of outliers are not appropriate for the evaluation of outlier detection methods, since outliers are by definition assumed to be exceptions in the data, and thus much less common than inliers. Accordingly, most outlier detection models either implicitly or explicitly assume that outliers are relatively rare — a characteristic which is confirmed by our experimental results, in the sense that all methods perform worse as more and more objects are included as outliers in a dataset. The conclusions about *relative* performance of the methods are however largely unaffected by the proportion of outliers in the dataset.

To compare the methods according to their *quality scores* we will consider initially (1) the *average performance* over the range of given values of k (representing an expected performance if the users have no prior knowledge about k), and (2) the best-case performance, selecting the k for which the performance of a method on a dataset is maximal (representing the optimistic case where the optimal value of k for a method is known in advance). To show how the number of outliers in a dataset affects the performance of the methods, we show here results that are aggregated over *all* datasets, as well as results that are aggregated over only the datasets that contain *up to 5 %* outliers.

In Fig. 2a, b, we show for each method the mean and standard error of the best-case ROC AUC values (aggregated over all datasets), for normalized and unnormalized datasets, respectively; Fig. 2c, d shows the same statistics aggregated over datasets that contain only up to 5 % outliers.

When comparing these figures, we can make two general observations: (1) normalization, on average, leads to better performance for all methods; and (2) having only a small percentage of objects labeled as outliers in the ground truth leads to better performance for all methods, and while some methods are more affected than others, their relative performance does not change dramatically. Because of observation (1), we will focus in the following on results for normalized datasets. Because of observation (2), we will focus on datasets with a small percentage of outliers in most cases while occasionally comparing results over different percentages of outliers. However, results for unnormalized data, as well as aggregated values for different maximum percentages of outliers (up to 10 % and up to 20 %) are available online; the main conclusions are the same across all these results.

Regarding the relative performance of the methods under their best parameter setting, we can see that KDEOS (along with FastABOD on unnormalized data) scores below the others in terms of ROC AUC performance. Although no one method consistently and significantly outperformed the others in all experiments, Fig. 2b does show that a group consisting of the methods kNN, kNNW, and LOF as well as the LOF-variant LDF does stand out to some extent when averaging best performance over all datasets. As shown in Fig. 2d, when averaging over the datasets containing

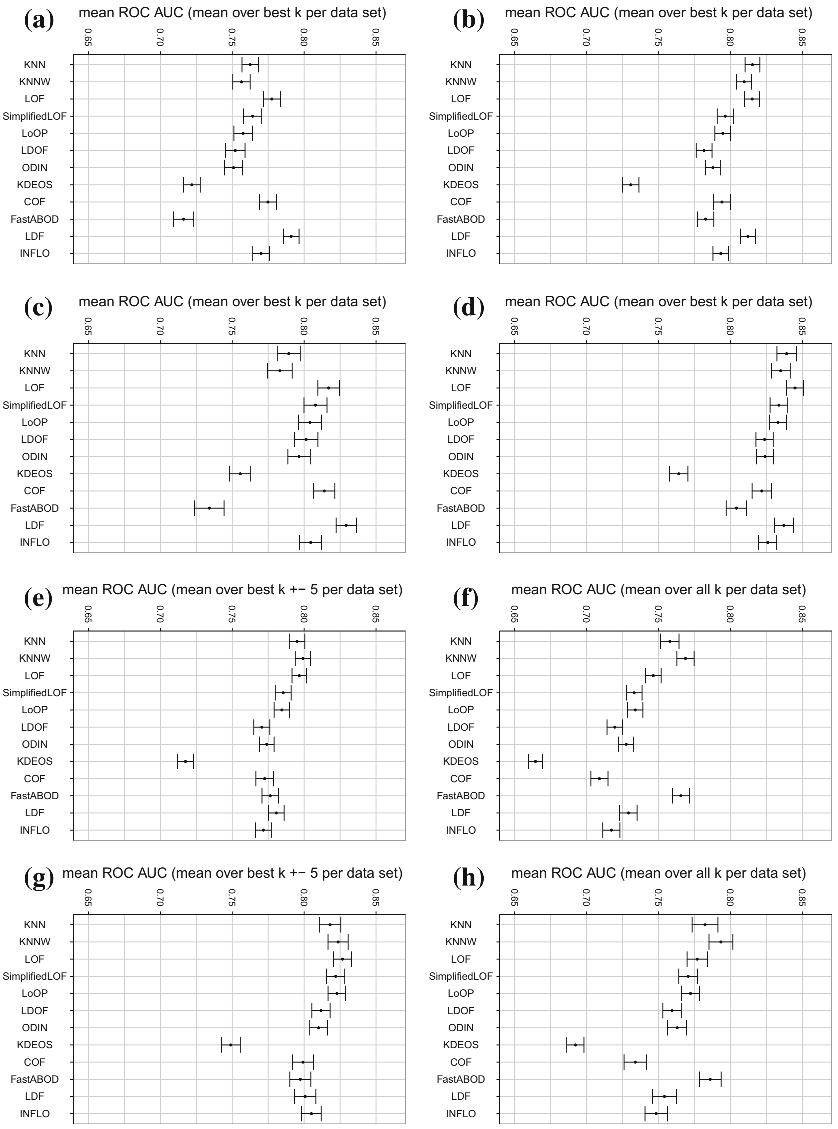

Fig. 2 Comparison of the achieved ROC AUC values per method, over different collections of datasets without duplicates. **a** Average over the maximum ROC AUC per method (selecting the optimal *k* for each method)—unnormalized data, aggregated over all datasets. **b** Average over the maximum ROC AUC per method (i.e., selecting the optimal *k* for each method)—normalized data. **c** Average over the maximum ROC AUC per method (selecting the optimal *k* for each method)—unnormalized data, aggregated over datasets with up to 5 % outliers. **d** Average over the maximum ROC AUC per method (selecting the optimal *k* for each method)—normalized data, aggregated over datasets with up to 5 % outliers. **e** Average ROC AUC over a window of size 10 around the optimal value of *k*—normalized data, aggregated over all datasets. **f** Average ROC AUC over all *k*—normalized data, aggregated over all datasets. **g** Average ROC AUC over a window of size 10 around the optimal value of *k*—normalized data, aggregated over datasets with up to 5 % outliers. **h** Average ROC AUC over all *k*—normalized data, aggregated over datasets with up to 5 % outliers

only up to 5 % of outliers, in addition to LOF, LDF, kNN, and kNNW, two more close variants of LOF also stand out: SimplifiedLOF, and LoOP.[8]

To study the stability of performance with respect to the choice of k, we consider the average performance within a range of ±5 around the optimal value: the less stable the method, the greater the expected degradation in performance when aggregating over this larger window. If necessary, the window may be shifted so that it fits entirely within the allowable range for k for the dataset in question. The results aggregated over all data sets are shown in Fig. 2e, and the results aggregated over datasets with up to 5 % outliers are shown in Fig. 2g. We observe that for all methods, the overall performance is degraded to some extent; however, this degradation is greater for some methods than for others (indicating that the method is less stable). The method LDF is shown to be the least stable, FastABOD shows a more stable behavior than the other methods, whereas the stabilities of the other methods are more or less comparable. When averaging best performance over all datasets, the methods kNN, kNNW, and LOF stand out to some extent, and when averaging over the datasets containing only up to 5 % of outliers, the best performing group also contains SimplifiedLOF and LoOP.

If we widen the window to include all values of k in the range 1 to 100, the performances of all methods degrade even further, as one would expect (Fig. 2f, h). Here, the top group of methods least affected by the variation of k are the distance-based methods kNN and kNNW, as well as FastABOD, followed by LOF, when averaging over all datasets, and also followed by SimplifiedLOF and LoOP, when averaging over the datasets containing only up to 5 % of outliers.

We applied the Friedman test (Friedman 1937) to examine whether there is a significant difference between the results of the algorithms on collected datasets. The null hypothesis for this test assumes that there is no significant difference between the algorithms. If the calculated probability is low (p-value less than the selected significance level) the null hypothesis is rejected, which indicates that at least two algorithms are significantly different from each other. The Nemenyi post-hoc test (Nemenyi 1963) can be applied in this scenario so as to reveal which pairs of algorithms exhibit such differences (if any). In the usage of both tests, we follow Demšar (2006).

The Friedman test was applied to the collection of datasets normalized and without duplicates, with the exception of ALOI and KDDCup99, since not all algorithms have provided results for these two datasets. We base the test on the best achieved quality in terms of ROC AUC (i.e., we chose for each method the best-performing parameter setting (k) for each dataset independently). Due to the assumption of independence between variables of the test, the results for those datasets with multiple subsampled variants were grouped by averaging the best results over all variants for each method.

The Friedman test returned a p-value of approximately 2.891E-10, which suggests that the null hypothesis is extremely unlikely. The results for the Nemenyi post-hoc

[8] We see the same overall tendency (although much weaker due to overall low values) if we use $P@n$ and AP (both adjusted and unadjusted) instead of ROC AUC. This is expected since (Adjusted) $P@n$ and (Adjusted) AP can yield additional insights when comparing results that are very good in terms of ROC AUC. In this aggregated evaluation, however, many results with weak scores are included. The corresponding plots are available online.

Table 3 Statistical differences among the 12 studied algorithms measured by the Nemenyi post-hoc test, using normalized datasets without duplicates (ALOI and KDDCup99 removed)

	kNN	kNNW	LOF	SimplifiedLOF	LoOP	LDOF	ODIN	KDEOS	COF	FastABOD	LDF	INFLO
kNN	=							−−				
kNNW		=						−−				
LOF			=			−	−−	−−		−−		
SimplifiedLOF				=				−−				
LoOP					=			−−				
LDOF			+			=						
ODIN			++				=					
KDEOS	++	++	++	++	++			=	++		++	++
COF								−−	=			
FastABOD			++							=	+	
LDF								−−		−	=	
INFLO								−−				=

The symbols indicate that the column method is better than the row method at 90 % (‘+’) and 95 % (‘++’) confidence levels. Worse performance is indicated for 90 % (‘−’) and 95 % (‘−−’) confidence test are shown in Table 3. The symbols '+' or '++' indicate that the column method is better than the row method with 90 % ('+') and 95 % ('++') confidence, the symbols '−' or '−−' indicate significantly worse performance. Two main observations can be made: (i) KDEOS is statistically worse than kNN, kNNW, LOF, SimplifiedLOF, LoOP, COF, LDF, and INFLO at the 95 % confidence level; and (ii) LOF is statistically better than three competitors (ODIN, KDEOS, and FastABOD) at the 95 % confidence level, better than LDOF at the 90 % confidence level, and is the "winner" from this particular perspective.

These findings also shed some light as to how the evaluation of new methods could be performed. It is always possible to find cases (specific parameter settings for specific datasets) where one particular method outperforms its competitors. As was demonstrated here, best practice dictates that the behavior of outlier detection methods be studied across a range of parameter settings, as the results for different parameter values can vary widely. Even if the methods to be compared share a seemingly analogous parameter (such as a neighborhood size k), setting it to the same values for all methods may still not allow for a direct comparison. As indicated in Fig. 1, the methods may depend on the parameter in different ways, and reach their peak performances for different choices of a seemingly identical parameter such as neighborhood size. Surveying the research literature would suggest that best practice is not always followed. There are many publications in which methods are compared and conclusions are made based on only a single, arbitrary choice of an important parameter (see the work of Müller et al. (2011, 2012), Liu et al. (2012), Keller et al. (2012), Ting et al. (2013) for some recent examples published at high quality venues).

To summarize these findings, we may conclude that after about 15 years of research, in general, the seminal methods kNN, kNNW, and LOF remain the state of the art, especially in datasets with possibly larger amounts of outliers. The methods SimplifiedLOF and LoOP that are closely related to LOF perform similarly well in datasets with few outliers, but not better than LOF, on average. LDF shows a relatively good peak performance but is rather unstable w.r.t. the choice of k. The peak performance of FastABOD is a bit below average but FastABOD is very stable w.r.t. k. It seems that none of the more recent developments offer *comprehensive* improvement over the classic methods, on average. It would indeed seem that there is no free lunch for unsupervised outlier detection.

5.3 Characterization of the datasets

We now characterize the properties of the studied datasets, and discuss their suitability for the evaluation of outlier detection methods. For this analysis, we examine the collective performance of our representative set of outlier detection methods on these datasets, based on notions of 'difficulty' of outlier detection, and a notion of 'diversity' of the results.

To obtain a better understanding of the relative difficulty of the datasets, we first consider performance scores for each dataset, aggregated over all methods. For this purpose, for a given dataset, we determine for each method the best performance score obtained over a range of parameter settings, and then average these scores over all methods. For each dataset to which random downsampling is applied over multiple

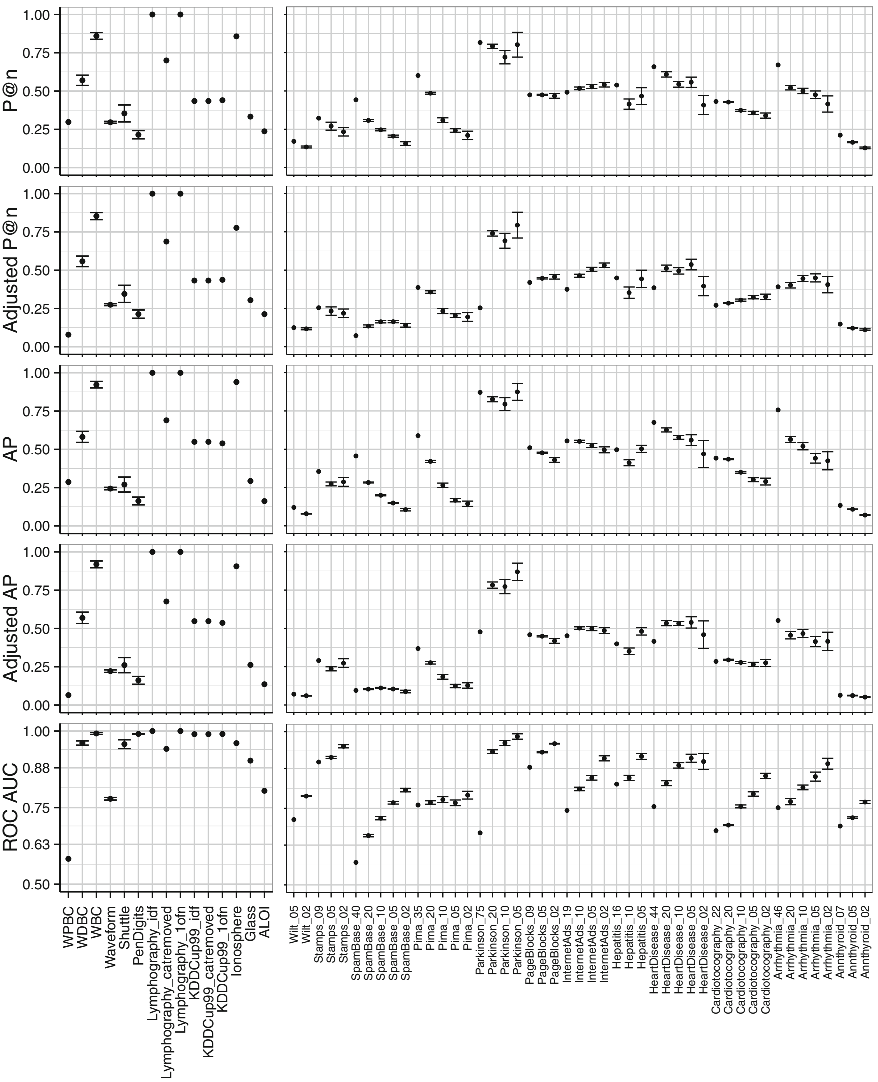

Fig. 3 Comparison of the best achieved quality per dataset, over all datasets (without duplicates, normalized)

runs, we also show the standard error as an interval about the mean score. Figure 3 shows the results for the datasets from Tables 1 and 2, for each of the performance measures considered.

The first (top) row in Fig. 3 shows the results using Precision at n ($P@n$). One can clearly observe a wide variation in $P@n$ across the different datasets. For example, on the Parkinson dataset variants, the methods achieve an average $P@n$ of approximately 0.75, whereas on the Annthyroid variants the scores fall below 0.2. Overall, we note a trend towards higher $P@n$ scores as the proportion of outliers in a dataset increases (for these proportions, see Tables 1 and 2; for the datasets in Table 2, we also show results for variants with smaller downsampling rates). As discussed earlier, this trend is due at least in part to the fact that a random ranking leads to an expected $P@n$ score of $|O|/N$, independently of the value of n. Increasing the proportion of outliers therefore increases the expected $P@n$ score for random rankings.

In order to control for differences in the proportions of outliers, we introduced the adjusted precision at n, Adjusted $P@n$, which is shown in the second row of Fig. 3. Here, the trend regarding increasing numbers of outliers is attenuated (or even reversed in some cases), in particular when comparing variants of the same dataset across different downsampling rates. Indeed, the results for Adjusted $P@n$ indicate that an increasing number of outliers can lead to a lower quality score (as in the case of the Cardiotocography set), which suggests that the outlier class may begin to form a cluster as the sample size increases. For datasets where the scores are already very low, such as Annthyroid, Pima, and SpamBase, increasing the number of outliers does not have a significant effect. As discussed earlier, both $P@n$ and Adjusted $P@n$ consider only the first n positions of a ranking. If n is low, the resulting scores may be very low, or highly variable, and thus inconclusive.

The third row in Fig. 3 shows the Average Precision, AP, for each dataset. As discussed in Sect. 3, AP attempts to overcome the deficiencies of $P@n$ by computing scores over multiple choices of n. However, the figure shows that as with $P@n$, AP scores tend to be higher for datasets with larger proportions of outliers. To a large extent, this effect can be explained by the increases in the expected $P@n$ values of which AP is the average.

The fourth row in Fig. 3 shows Adjusted AP. Again, the adjusted AP scores tend to be more stable than the unadjusted AP scores when the proportion of outliers is increased.

The fifth row in Fig. 3 shows the results obtained with the most commonly-used performance measure for outlier detection methods, ROC AUC. Like the Average Precision (both adjusted and unadjusted), ROC AUC takes the entire outlier ranking into account. The ROC AUC scores show a clear decreasing trend as the proportion of outliers is increased. This trend is clear even for those datasets (such as Spambase) where Adjusted $P@n$ was less discriminative. However, it is also clear that a relatively high ROC AUC score indicates only that, in the overall ranking, outliers are more likely to be ranked ahead of inliers; it does not necessarily mean that the top rankings are dominated by outliers. We would therefore argue that one cannot rely solely on ROC AUC scores in judging the quality of an outlier method—rather, ROC AUC and Adjusted$P@n$ complement each other, as they reveal different aspects of an outlier ranking, both of which are relevant in practice.

To eliminate the percentage of outliers as a factor that can influence the relative performance of methods, we will restrict the following analysis to datasets with a comparable proportion of outliers, selecting a variant from those datasets with between 3 and 5 % of outliers. Since every dataset from Table 2, and the datasets ALOI, Glass, Lymphography (for which we select Lymphography_idf), Waveform, WBC, and WDBC from Table 1, all have a variant where the proportion of outliers falls in this range, this will still include a variant of the majority of the different types of data.

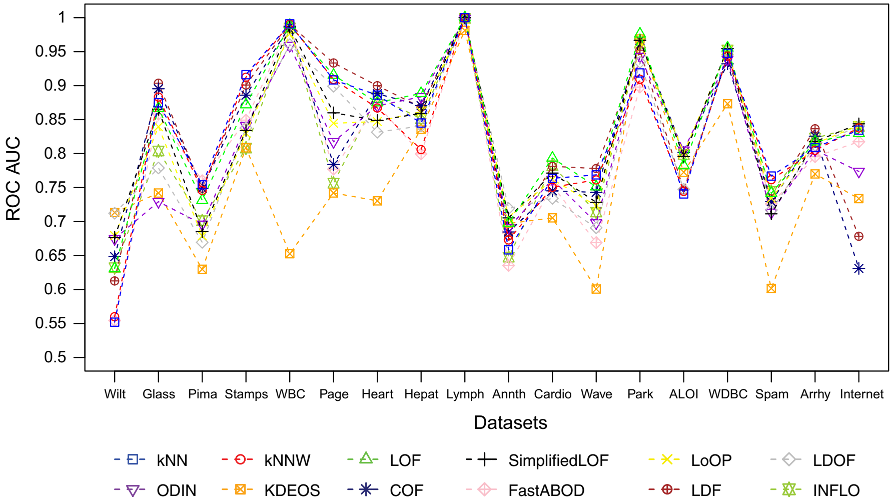

Fig. 4 Average best ROC AUC scores for datasets with between 3 and 5 % of outliers (without duplicates, normalized), arranged from left to right by increasing dimensionality

Since it is often presumed that the dimensionality of a dataset can be a contributing factor in how well a method performs on the dataset (Houle et al. 2010; Zimek et al. 2012), we first address the question whether this is the case for the given datasets and methods. In Fig. 4, we show the ROC AUC scores for each method (using the best k parameter for each method) on each of the datasets with between 3 and 5 % of outliers, averaged over the different variants where available. The datasets are arranged on the x-axis of the plot from left to right in order of increasing dimensionality (see Tables 1 and 2 for the dimensionalities of the datasets). Values that are not available (FastABOD on ALOI) are omitted, i.e. the function of FastABOD is discontinuous. One can see that there is no clear pattern that would indicate an effect of the dimensionality for these datasets. In fact the lowest dimensional dataset Wilt (5 attributes) is among the most ‘difficult’, and InternetAds, even though very high-dimensional (1555 attributes) is of intermediate difficulty for most of the methods. Furthermore, all methods show a similar behavior for most datasets, suggesting that the difficulty of the datasets is independent of their dimensionality.

To further analyze the suitability of the datasets for outlier detection, we consider now first one variant for each of the datasets: for ALOI, Glass, Lymphography, and Wilt these are just the given ‘base’ datasets; for the other datasets, one of the 10 downsampled versions is chosen at random. As always, the full details for all datasets can be found in the web repository.

In Fig. 5, for each of the selected datasets, a heat map is shown that represents for each outlier detection method (x-axis) the (binned) rank it gives to the outliers in the ground truth (y-axis)—using the best overall solution (the best value for k) according to ROC AUC. In other words, the (x, y) position in a plot for a dataset represents the (binned) rank that the method x has given to ground truth outlier y. The outliers for each dataset are ordered along the y-axis of the corresponding plots according to the average of ranks achieved over all methods, with the top-ranked outliers appearing at

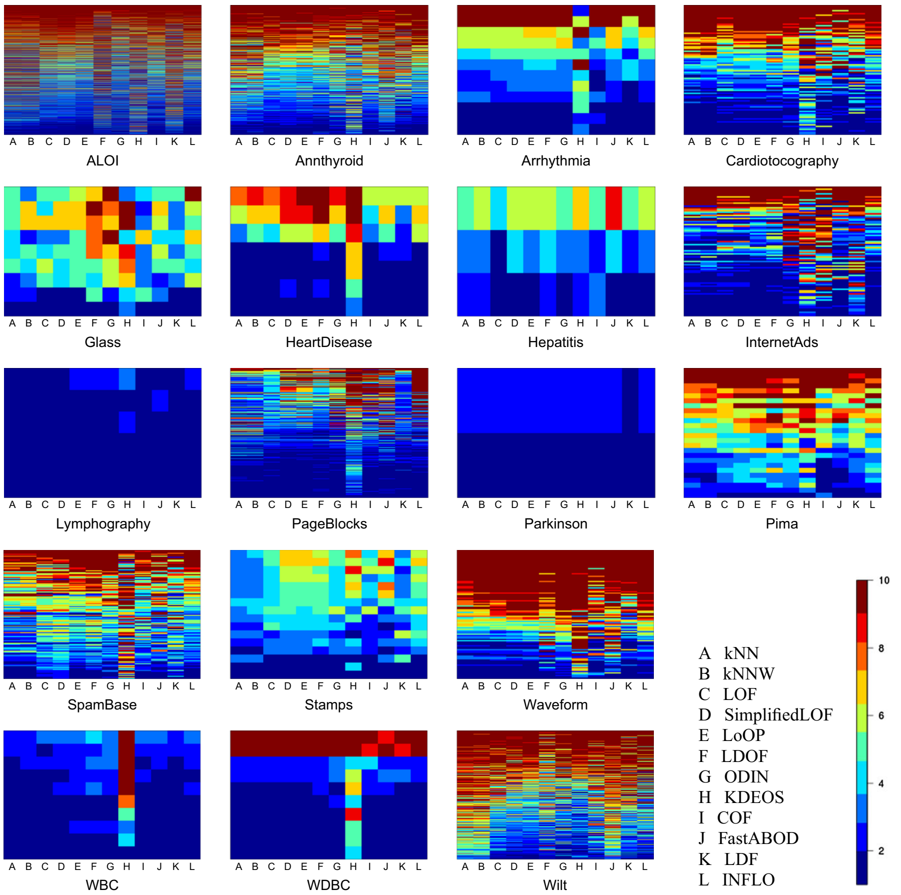

Fig. 5 Diversity of outlier ranks in 17 datasets (without duplicates, normalized, with 3–5 % outliers). The (binned) ranks refer to the best solution obtained by each method with respect to k, according to ROC AUC. Note that FastABOD is missing for ALOI (due to excessive space requirements for this large set)

the bottom of the heat map. For the heat maps, the ranks are binned and color coded in the following way. Given $n = |O|$ outliers in the ground truth of a dataset, and a ranking of all N points in that dataset by an outlier method, the first n rank positions are assigned to Bin 1 (thus outliers whose rank falls into this bin would have contributed to the $P@n$ score for that method), the second n positions are assigned Bin 2 (outliers falling in this bin would have contributed to the $P@2n$ for that method but not to the $P@n$ score), and so on, up to Bin number 9,[9] and all ranks higher than $9 \cdot n$ are assigned to bin 10. Bin 1 is assigned the color dark blue, Bin 10 is assigned the color dark red, and the other bin colors are assigned in consecutive order within the color spectrum, as indicated in the legend of Fig. 5.

[9] Therefore, as a side effect, such heat maps can also serve to visualize the profile of performance in terms of $P@(x \cdot n)$ for $x = 1, \ldots, 9$.

Since for each method we take account only of the best result achieved, these plots tell us less about the overall performance by a method on a given dataset, but more about how difficult the individual outliers are to detect (with blue outliers being easiest to detect), and the differences in the level of (best-case) difficulty across the various methods. For several of these datasets, most of the outliers are relatively easy for the majority of methods to detect. The Parkinson dataset variant shown is an extreme example in which all of the methods tested place all n outliers within the top $2n$ ranks. In the WDBC variant, most outliers are placed in the top ranks by most methods, with only two outliers that all methods essentially failed to identify. In the Waveform variant, roughly one third of the outliers are easily identified by most methods, while roughly half are not identified in almost all cases. Overall, we see a wide spectrum of difficulty of detection, where some outliers are easily identified by all methods, some are ranked highly by some methods but not others, and some outliers are not detected at all among the top-$9n$ by any of the methods.

To characterize the properties of the datasets across different downsampling rates, and to facilitate the discussion of their suitability for evaluating outlier detection methods, we formalize notions of 'difficulty' and 'diversity' of a specific dataset variant over a given set of representative outlier detection methods.

Difficulty of a dataset is simply defined as the average of the (binned) ranks of all outliers in the dataset reported by the given set of outlier methods (for each variant shown in Fig. 5, this is the average bin number depicted in the corresponding plot). Datasets with low difficulty score contain outliers that are relatively easy to detect by the majority of methods. A high difficulty score indicates that most or all methods have difficulty in finding the outliers.

Diversity characterizes the agreement of a given collection of outlier methods with respect to the scores they give to the outliers. For each individual outlier o, the diversity score of o is defined as the standard deviation of the (binned) ranks reported for this outlier by the different methods. The diversity score for a dataset is then computed as the Root Mean Square (RMS) of the diversity scores of all outliers in the dataset. If a dataset has a low diversity score, the methods largely agree on the difficulty of identifying the outliers of the dataset. A high diversity score indicates a large disagreement on the ranking of the outliers.

Figure 6 shows the position of each dataset with between 3 and 5 % outliers in the space of diversity vs. difficulty. For each dataset with downsampled variants, the 95 % confidence ellipse is also shown to indicate the extent to which the difficulty and diversity can vary with downsampling. The datasets with only one variant are shown as filled dots. The plot also includes artificial points that indicate certain boundary cases: (1) in the lower left corner, a point representing a 'perfect' result, with a diversity score of 0 (all methods agree on the binned ranks of all outliers) and a difficulty score of 1 (all methods identify all n outliers within the top n ranks); (2) in the upper right corner, points representing the difficulty and diversity scores that would be obtained if each of the 12 methods returned a uniform random ranking of all dataset objects; (3) in the lower right corner, points representing the results for a set of 12 *identical* random rankers, resulting always in a diversity score of 0, but varying in difficulty by chance.

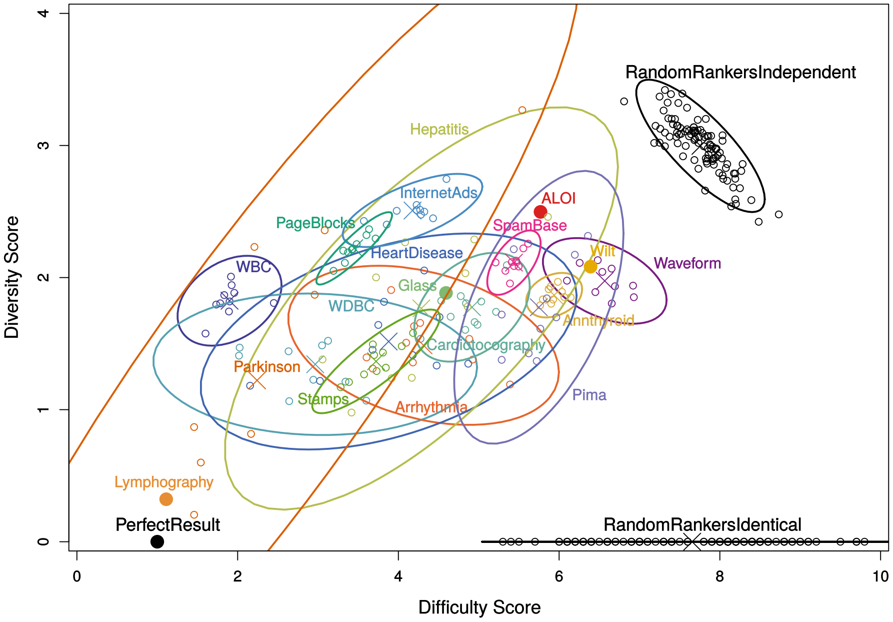

Fig. 6 Diversity versus difficulty of datasets with 3–5 % outliers (without duplicates, normalized). For each dataset with different downsampled variants, the 95 % confidence ellipse is shown as well as the individual points for each variant. The means of the ellipses, which do not correspond to actual data instances, are indicated by an '×' labeled with the corresponding base dataset. The scores of random rankers are the results of 100 simulations with a dataset of 200 points, of which 10 are outliers

Note that diversity and difficulty are not completely independent of each other. On datasets of very low difficulty, it follows from the definition that outlier methods must then also tend to agree on the rankings of the outliers. It is impossible to have at the same time high diversity in outlier rankings and a low difficulty score, and thus one would not expect to find datasets in the upper left area of the plot. Also, for the model of identical random rankers, if by chance a larger number of outliers is found in top positions of the ranking, which the majority of the rankers agree on, we can also expect to see a lower difficulty score. While individual random rankers may occasionally, by chance, obtain a high ranking, it is much less likely for many independent rankers to achieve it simultaneously. As one can observe in the figure, this leads to a much lower variance in difficulty scores for the 12 independent random rankers than for the 12 identical rankers.

From Fig. 6 we can also observe that most of the datasets considered in this study are of moderate difficulty and diversity. Such datasets are of greatest interest in outlier evaluation, as they have significant variety in their performance characteristics, and allow for the possibility of a new method to demonstrate its general superiority over existing methods by detecting the outliers in these datasets more consistently.

The observed scores also indicate that the outliers in the given datasets exhibit at least some of the properties that the methods attempt to model. The 12 studied methods lead, in general, to difficulty scores that are much lower than that of the random rankers,

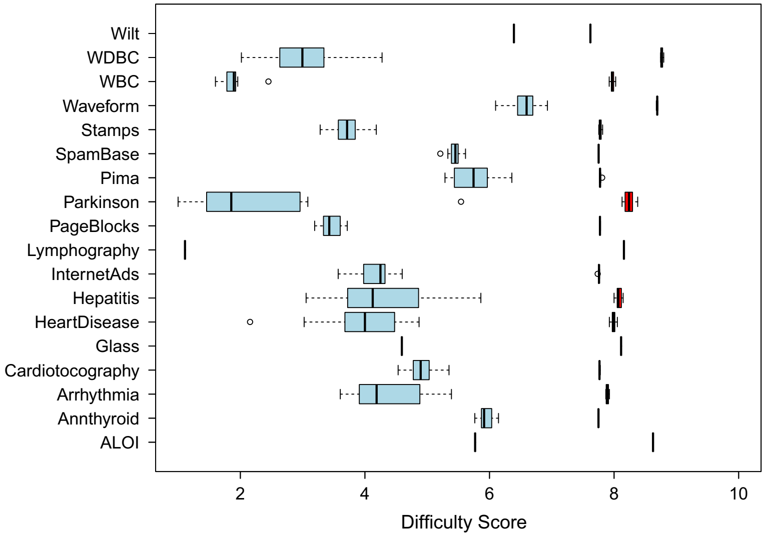

Fig. 7 On the *left* the distribution of observed difficulty scores for each dataset. On the *right* for each dataset the distribution of difficulty scores obtained after randomly permuting the ground truth labels. For datasets where only a single variant has between 3 and 5 % of outliers, only a single value is plotted

indicating that a significant proportion of the objects labeled as outliers can indeed be considered as outliers by the outlier models corresponding to the different methods.

Datasets that are centrally-located in Fig. 6 can potentially offer insights into the strengths and weaknesses of different outlier methods. However, in most cases we observe a large variance in both diversity and difficulty scores for different downsamplings of the same base dataset (indicated by the error ellipses). This variance can be extreme (as in the case of Parkinson[10]), or more moderate (as in Annthyroid). The large observed variances indicate that, in general, the use of a single downsampled dataset is not appropriate when evaluating the performance of outlier detection methods.

To investigate whether the observed results are significantly different from those of a set of random rankers with the same dependency between them, we also performed experiments in which we computed the difficulty score for each dataset based on the ranking by the 12 methods, but after a random permutation of the ground truth labels. The results are shown in Fig. 7. The figure shows for each dataset the distribution of observed difficulty scores as a boxplot (in light blue), together with a boxplot of the distribution of difficulty scores obtained after randomizing the ground truth labels (in red), combining 1000 randomizations per dataset variant. The results clearly demonstrate that (at least some of) the objects labeled as outliers agree with the outlier model. For each base dataset with 10 different downsamplings, not a single random result in 10,000 is even close to the observed difficulty values; for the datasets with only a single variant none of the 1,000 random results is close to the observed value.

[10] This is not surprising given the relatively large amount of outliers (≈75 %) in the base dataset.

6 Conclusions

In this experimental study, we addressed a constant challenge in unsupervised outlier detection: the evaluation of algorithms in terms of *effectiveness*. We have discussed the notorious lack of commonly accepted, well-understood benchmark datasets with annotated ground truth. We also elaborated on commonly used evaluation measures, their strengths and weaknesses, and how several measures can be used in combination to provide insights into the relative performance of outlier methods. For precision at n and for average precision, we proposed an adjustment for chance, which allows meaningful comparisons of the performances of methods on different datasets.

Using the study of evaluation measures as a foundation, we performed an extensive experimental analysis of a representative set of both classical and recent unsupervised outlier detection methods, on a large collection of datasets.

Papers proposing a novel method often justify its performance based on a specific evaluation measure, on few datasets, and for few parameter settings. By using a diverse collection of datasets, several evaluation measures, and a broad range of parameter settings, we argue here that it is typically pointless and unjustified to state the superior behavior of any method *for the general case*. For optimization problems in machine learning, this fact is captured in the 'no free lunch' theorem (Wolpert 1996). For unsupervised learning approaches, it has been conjectured that the quest for a truly general and superior method is futile [at least for clustering, this has been discussed by Estivill-Castro (2002), Kriegel et al. (2009b), and von Luxburg et al. (2012)], but there is no common understanding of the implications of this conjecture. We therefore show here that in the evaluation of new algorithms for outlier detection, the goal should be to analyze where their strengths and (perhaps more importantly) weaknesses lie, when confronted with datasets of different characteristics.

The gist of our findings is that, when considering the totality of results produced in a systematic way across different parameter settings and a diverse collection of datasets (rather than specific parameter settings for specific datasets), the seminal methods kNN, kNNW, and LOF still remain the state of the art—none of the more recent methods tested offer any *comprehensive* improvement over those classics, while two methods in particular (LDF and KDEOS) have been found to be noticeably less robust to parameter choices. However, by picking appropriate parameter values, one may cast any of the methods tested in a favorable light, which emphasizes the importance of systematic testing across a range of parameter values.[11]

These findings should be taken with a grain of salt, as our selection of methods included—among general, rather abstract methods such as kNN and LOF—also rather specialized methods such as FastABOD (high dimensional data) or KDEOS (kernel-based, where the choice of a kernel allows for adaptation to very specific application problems). Recall (Sect. 2) that both FastABOD and KDEOS (and also LDF) require other parameters in addition to the neighborhood size—parameters that have not been studied here but that have reasonable default settings. Furthermore,

[11] Prima facie, this conclusion is valid, based on our experiments, for the dependency of related methods on a parameter choice regarding cardinality of a local neighborhood. Common sense suggests that we can have a similar expectation, mutatis mutandis, for other types of parameters for other kinds of methods.

KDEOS was designed as ensemble method and was tested here as individual method, putting it at a disadvantage. The evaluation of ensemble methods for outlier detection (Zimek et al. 2013a) poses different questions and challenges for future work.

Thus our findings do not allow us to conclude that these two methods perform worse than the others *in general*. If a method does not compete well on arbitrary datasets with classic, generic solutions like kNN or LOF, this does not at all mean that the same method cannot excel on particular domains. Our findings do suggest, however, that novel methods should not be proposed without also indicating those domains or application scenarios where this method is particularly well suited. Merely demonstrating that the method excels on a few datasets for a few parameter settings does not suffice. Most importantly, broad ranges of parameter choices should be tested for the competitors.

Another source of arbitrariness in the outlier evaluation research literature is the very common practice of producing datasets with outlier ground truth by means of class downsampling (along with other preprocessing steps). Our study shows that observations based on downsampling can vary considerably from sample to sample, and thus experimentation on only a small number of downsampled sets may not produce meaningful outcomes. Since for some sets there may be significant variance even when many downsampled variants have been considered, for the sake of reproducibility, the dataset samples that are adopted in an evaluation should be made publicly available, as we have done in our web repository.

On the positive side, our experimental study has provided a better understanding of the characteristics of datasets in current use, according to their suitability for evaluating outlier detection methods. Our characterizations can in principle be extended to other datasets, and thus our methodology could eventually serve to establish a commonly accepted collection of benchmark datasets for the evaluation of outlier detection methods. The extensive collection of results in our web repository can serve as a basis of comparison between established outlier methods and any new methods that may be proposed in the future, over a variety of datasets and a broad range of parameter settings, while avoiding the need to run new experiments on the established methods.

For a typical scenario where this study can be useful for future research in this domain, let us consider the situation in which researchers have developed a new outlier detection method, and have available to them for the evaluation of the method some dataset with annotated ground truth. Such researchers can make twofold use of our results.

1. They can test their method on the datasets provided in our repository and directly compare its performance with the results of the 12 standard methods we used. In addition to summaries and statistics, we provide also all raw results on our webpage. If the new method is competitive on our datasets, and if the authors can identify a scenario where their new method is particularly well suited (be it more generally applicable across many types of data—such as categorical, graph, or sparse vector data—or be it adapted to more specific purposes—such as high-dimensional data, or a particular domain), much more evidence can be generated for the advantages and disadvantages of the novel method than can be found in many publications today.

2. They can run the 12 standard methods from our collection on their new dataset, and perform the analysis as presented in Sect. 5.3. In this way, the new dataset can be situated within the space of diversity vs. difficulty (cf. Fig. 6), and the ground truth can be compared with random labelings as in our analysis in Fig. 7. Ideally, a new dataset adds value to the portfolio by providing different but equally reasonable challenges.

In this paper, we do not claim to have delivered the ultimate benchmark dataset collection for outlier detection. Furthermore, the selection of outlier detection methods used in our study is not exhaustive. The online repository could be extended to accommodate both novel methods and additional datasets. We provide online all scripts and implementations required to repeat our experiments. The same scripts and implementations, using ELKI 0.7 (Schubert et al. 2015a), can be easily used to extend the experiments, including more methods and more datasets. We plan to extend the repository in both directions and offer to include also methods and datasets as suggested or provided by users.

This study focused on representative unsupervised outlier detection models based on neighborhoods in Euclidean space. Future extensions of our study could include approximation methods [assessing speed-up as well as approximation quality, extending the work of Orair et al. (2010)], special methods for high-dimensional data (Zimek et al. 2012), or recently-developed ensemble techniques (Zimek et al. 2013a).

Acknowledgments This project was partially funded by FAPESP (Brazil—Grant #2013/18698-4), CNPq (Brazil—Grants #304137/2013-8 and #400772/2014-0), NSERC (Canada), and the Danish Council for Independent Research—Technology and Production Sciences (FTP) (Denmark—Grant 10-081972).

References

Abe N, Zadrozny B, Langford J (2006) Outlier detection by active learning. In: Proceedings of the 12th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD), Philadelphia, pp 504–509. doi:10.1145/1150402.1150459

Achtert E, Kriegel HP, Schubert E, Zimek A (2013) Interactive data mining with 3D-parallel-coordinate-trees. In: Proceedings of the ACM international conference on management of data (SIGMOD), New York, pp 1009–1012. doi:10.1145/2463676.2463696

Aggarwal CC (2013) Outlier analysis. Springer, Berlin

Akoglu L, Tong H, Koutra D (2015) Graph-based anomaly detection and description: a survey. Data Mining Knowl Discov 29(3):626–688. doi:10.1007/s10618-014-0365-y

Angiulli F, Pizzuti C (2002) Fast outlier detection in high dimensional spaces. In: Proceedings of the 6th European Conference on Principles of Data Mining and Knowledge Discovery (PKDD), Helsinki, pp 15–26. doi:10.1007/3-540-45681-3_2

Angiulli F, Pizzuti C (2005) Outlier mining in large high-dimensional data sets. IEEE Trans Knowl Data Eng 17(2):203–215. doi:10.1109/TKDE.2005.31

Bache K, Lichman M (2013) UCI machine learning repository. http://archive.ics.uci.edu/ml

Barnett V, Lewis T (1994) Outliers in statistical data, 3rd edn. Wiley, New York

Breunig MM, Kriegel HP, Ng R, Sander J (2000) LOF: identifying density-based local outliers. In: Proceedings of the ACM international conference on management of data (SIGMOD), Dallas, pp 93–104. doi:10.1145/342009.335388

Chandola V, Banerjee A, Kumar V (2009) Anomaly detection: a survey. ACM Comput Surveys 41(3):1–58. doi:10.1145/1541880.1541882

Craswell N (2009a) Precision at n. In: Liu L, Özsu MT (eds) Encyclopedia of database systems. Springer, Berlin, pp 2127–2128. doi:10.1007/978-0-387-39940-9_484

Craswell N (2009b) R-precision. In: Liu L, Özsu MT (eds) Encyclopedia of database systems. Springer, Berlin, p 2453. doi:10.1007/978-0-387-39940-9_486

Dang XH, Micenková B, Assent I, Ng R (2013) Outlier detection with space transformation and spectral analysis. In: Proceedings ofthe 13th SIAM international conference on data mining (SDM), Austin, pp 225–233

Dang XH, Assent I, Ng RT, Zimek A, Schubert E (2014) Discriminative features for identifying and interpreting outliers. In: Proceedings of the 30th International Conference on Data Engineering (ICDE), Chicago, pp 88–99. doi:10.1109/ICDE.2014.6816642

Davis J, Goadrich M (2006) The relationship between precision-recall and ROC curves. In: Proceedings of the 23rd international conference on machine learning (ICML), Pittsburgh, pp 233–240

de Vries T, Chawla S, Houle ME (2010) Finding local anomalies in very high dimensional space. In: Proceedings of the 10th IEEE International Conference on Data Mining (ICDM), Sydney, pp 128–137. doi:10.1109/ICDM.2010.151

de Vries T, Chawla S, Houle ME (2012) Density-preserving projections for large-scale local anomaly detection. Knowl Inf Syst 32(1):25–52. doi:10.1007/s10115-011-0430-4

Demšar J (2006) Statistical comparisons of classifiers over multiple data sets. J Mach Learn Res 7:1–30

Emmott AF, Das S, Dietterich T, Fern A, Wong WK (2013) Systematic construction of anomaly detection benchmarks from real data. In: Workshop on outlier detection and description, held in conjunction with the 19th ACM SIGKDD international conference on knowledge discovery and data mining, Chicago, pp 16–21

Estivill-Castro V (2002) Why so many clustering algorithms—a position paper. ACM SIGKDD Explor 4(1):65–75. doi:10.1145/568574.568575

Friedman M (1937) The use of ranks to avoid the assumption of normality implicit in the analysis of variance. J Am Stat Assoc 32(200):675–701. doi:10.1080/01621459.1937.10503522

Färber I, Günnemann S, Kriegel HP, Kröger P, Müller E, Schubert E, Seidl T, Zimek A (2010) On using class-labels in evaluation of clusterings. In: MultiClust: 1st international workshop on discovering, summarizing and using multiple clusterings held in conjunction with KDD 2010, Washington, DC

Gao J, Tan PN (2006) Converting output scores from outlier detection algorithms into probability estimates. In: Proceedings of the 6th IEEE international conference on data mining (ICDM), Hong Kong, pp 212–221. doi:10.1109/ICDM.2006.43

Hanley JA, McNeil BJ (1982) The meaning and use of the area under a receiver operating characteristic (ROC) curve. Radiology 143:29–36

Hautamäki V, Kärkkäinen I, Fränti P (2004) Outlier detection using k-nearest neighbor graph. In: Proceedings of the 17th international conference on pattern recognition (ICPR), Cambridge, pp 430–433. doi:10.1109/ICPR.2004.1334558

Hawkins D (1980) Identification of outliers. Chapman and Hall, London

Houle ME, Kriegel HP, Kröger P, Schubert E, Zimek A (2010) Can shared-neighbor distances defeat the curse of dimensionality? In: Proceedings of the 22nd international conference on scientific and statistical database management (SSDBM), Heidelberg, pp 482–500. doi:10.1007/978-3-642-13818-8_34

Hubert L, Arabie P (1985) Comparing partitions. J Classif 2(1):193–218

Jin W, Tung AKH, Han J, Wang W (2006) Ranking outliers using symmetric neighborhood relationship. In: Proceedings of the 10th Pacific-Asia conference on knowledge discovery and data mining (PAKDD), Singapore, pp 577–593. doi:10.1007/11731139_68

Keller F, Müller E, Böhm K (2012) HiCS: high contrast subspaces for density-based outlier ranking. In: Proceedings of the 28th international conference on data engineering (ICDE), Washington, DC, pp 1037–1048. doi:10.1109/ICDE.2012.88

Knorr EM, Ng RT (1997) A unified notion of outliers: properties and computation. In: Proceedings of the 3rd ACM international conference on knowledge discovery and data mining (KDD), Newport Beach, pp 219–222. doi:10.1145/782010.782021

Knorr EM, Ng RT (1998) Algorithms for mining distance-based outliers in large datasets. In: Proceedings of the 24th international conference on very large data bases (VLDB), New York, pp 392–403

Kriegel HP, Schubert M, Zimek A (2008) Angle-based outlier detection in high-dimensional data. In: Proceedings of the 14th ACM international conference on knowledge discovery and data mining (SIGKDD), Las Vegas, pp 444–452. doi:10.1145/1401890.1401946

Kriegel HP, Kröger P, Schubert E, Zimek A (2009a) LoOP: local outlier probabilities. In: Proceedings of the 18th ACM conference on information and knowledge management (CIKM), Hong Kong, pp 1649–1652. doi:10.1145/1645953.1646195

Kriegel HP, Kröger P, Zimek A (2009b) Clustering high dimensional data: a survey on subspace clustering, pattern-based clustering, and correlation clustering. ACM Trans Knowl Discov Data 3(1):1–58. doi:10.1145/1497577.1497578

Kriegel HP, Kröger P, Schubert E, Zimek A (2011a) Interpreting and unifying outlier scores. In: Proceedings of the 11th SIAM international conference on data mining (SDM), Mesa, pp 13–24. doi:10.1137/1.9781611972818.2

Kriegel HP, Schubert E, Zimek A (2011b) Evaluation of multiple clustering solutions. In: 2nd MultiClust Workshop: Discovering, Summarizing and Using Multiple Clusterings Held in Conjunction with ECML PKDD 2011, Athens, Greece, pp 55–66

Kriegel HP, Schubert E, Zimek A (2015) The (black) art of runtime evaluation: Are we comparing algorithms or implementations? submitted

Latecki LJ, Lazarevic A, Pokrajac D (2007) Outlier detection with kernel density functions. In: Proceedings of the 5th international conference on machine learning and data mining in pattern recognition (MLDM), Leipzig, pp 61–75. doi:10.1007/978-3-540-73499-4_6

Lazarevic A, Kumar V (2005) Feature bagging for outlier detection. In: Proceedings of the 11th ACM international conference on knowledge discovery and data mining (SIGKDD), Chicago, pp 157–166. doi:10.1145/1081870.1081891

Liu FT, Ting KM, Zhou ZH (2012) Isolation-based anomaly detection. ACM Trans Knowl Discov Data 6(1):31–39

Marques HO, Campello RJGB, Zimek A, Sander J (2015) On the internal evaluation of unsupervised outlier detection. In: Proceedings of the 27th international conference on scientific and statistical database management (SSDBM), San Diego, pp 7:1–12. doi:10.1145/2791347.2791352

Micenková B, van Beusekom J, Shafait F (2012) Stamp verification for automated document authentication. In: 5th International workshop on computational forensics

Müller E, Schiffer M, Seidl T (2011) Statistical selection of relevant subspace projections for outlier ranking. In: Proceedings of the 27th international conference on data engineering (ICDE), Hannover, pp 434–445. doi:10.1109/ICDE.2011.5767916

Müller E, Assent I, Iglesias P, Mülle Y, Böhm K (2012) Outlier ranking via subspace analysis in multiple views of the data. In: Proceedings of the 12th IEEE international conference on data mining (ICDM), Brussels, pp 529–538. doi:10.1109/ICDM.2012.112

Nemenyi P (1963) Distribution-free multiple comparisons. PhD thesis, New Jersey

Nguyen HV, Gopalkrishnan V (2010) Feature extraction for outlier detection in high-dimensional spaces. J Mach Learn Res Proc Track 10:66–75

Nguyen HV, Ang HH, Gopalkrishnan V (2010) Mining outliers with ensemble of heterogeneous detectors on random subspaces. In: Proceedings of the 15th international conference on database systems for advanced applications (DASFAA), Tsukuba, pp 368–383. doi:10.1007/978-3-642-12026-8_29

Orair GH, Teixeira C, Wang Y, Meira W Jr, Parthasarathy S (2010) Distance-based outlier detection: consolidation and renewed bearing. Proc VLDB Endow 3(2):1469–1480

Radovanović M, Nanopoulos A, Ivanović M (2014) Reverse nearest neighbors in unsupervised distance-based outlier detection. IEEE Trans Knowl Data Eng. doi:10.1109/TKDE.2014.2365790

Ramaswamy S, Rastogi R, Shim K (2000) Efficient algorithms for mining outliers from large data sets. In: Proceedings of the ACM international conference on management of data (SIGMOD), Dallas, pp 427–438. doi:10.1145/342009.335437

Schubert E, Wojdanowski R, Zimek A, Kriegel HP (2012) On evaluation of outlier rankings and outlier scores. In: Proceedings of the 12th SIAM international conference on data mining (SDM), Anaheim, pp 1047–1058. doi:10.1137/1.9781611972825.90

Schubert E, Zimek A, Kriegel HP (2014a) Generalized outlier detection with flexible kernel density estimates. In: Proceedings of the 14th SIAM International Conference on Data Mining (SDM), Philadelphia, pp 542–550. doi:10.1137/1.9781611973440.63

Schubert E, Zimek A, Kriegel HP (2014b) Local outlier detection reconsidered: a generalized view on locality with applications to spatial, video, and network outlier detection. Data Min Knowl Discov 28(1):190–237. doi:10.1007/s10618-012-0300-z

Schubert E, Koos A, Emrich T, Züfle A, Schmid KA, Zimek A (2015a) A framework for clustering uncertain data. Proc VLDB Endow 8(12):1976–1979

Schubert E, Zimek A, Kriegel HP (2015b) Fast and scalable outlier detection with approximate nearest neighbor ensembles. In: Proceedings of the 20th international conference on database systems for advanced applications (DASFAA), Hanoi, Vietnam, pp 19–36. doi:10.1007/978-3-319-18123-3_2

Tang J, Chen Z, Fu AWC, Cheung DW (2002) Enhancing effectiveness of outlier detections for low density patterns. In: Proceedings of the 6th Pacific-Asia conference on knowledge discovery and data mining (PAKDD), Taipei, pp 535–548. doi:10.1007/3-540-47887-6_53

Ting KM, Zhou GT, Liu FT, Tan SC (2013) Mass estimation. Mach Learn 90(1):127–160. doi:10.1007/s10994-012-5303-x

Vendramin L, Campello RJGB, Hruschka ER (2010) Relative clustering validity criteria: a comparative overview. Stat Anal Data Min 3(4):209–235. doi:10.1002/sam.10080

von Luxburg U, Williamson RC, Guyon I (2012) Clustering: science or art? JMLR Workshop Conf Proc 27:65–79

Vreeken J, Tatti N (2014) Interesting patterns, chapter 5. In: Aggarwal CC, Han J (eds) Frequent pattern mining. Springer, Berlin, pp 105–134. doi:10.1007/978-3-319-07821-2_5

Wang Y, Parthasarathy S, Tatikonda S (2011) Locality sensitive outlier detection: a ranking driven approach. In: Proceedings of the 27th international conference on data engineering (ICDE), Hannover, pp 410–421. doi:10.1109/ICDE.2011.5767852

Wolpert DH (1996) The lack of a priori distinctions between learning algorithms. Neural Comput 8(7):1341–1390. doi:10.1162/neco.1996.8.7.1341

Yang J, Zhong N, Yao Y, Wang J (2008) Local peculiarity factor and its application in outlier detection. In: Proceedings of the 14th ACM international conference on knowledge discovery and data mining (SIGKDD), Las Vegas, pp 776–784. doi:10.1145/1401890.1401983

Zhang E, Zhang Y (2009) Average precision. In: Liu L, Özsu MT (eds) Encyclopedia of database systems. Springer, Berlin, pp 192–193. doi:10.1007/978-0-387-39940-9_482

Zhang K, Hutter M, Jin H (2009) A new local distance-based outlier detection approach for scattered real-world data. In: Proceedings of the 13th Pacific-Asia conference on knowledge discovery and data mining (PAKDD), Bangkok, pp 813–822. doi:10.1007/978-3-642-01307-2_84

Zimek A, Vreeken J (2015) The blind men and the elephant: on meeting the problem of multiple truths in data from clustering and pattern mining perspectives. Mach Learn 98(1–2):121–155. doi:10.1007/s10994-013-5334-y

Zimek A, Schubert E, Kriegel HP (2012) A survey on unsupervised outlier detection in high-dimensional numerical data. Stat Anal Data Min 5(5):363–387. doi:10.1002/sam.11161

Zimek A, Campello RJGB, Sander J (2013a) Ensembles for unsupervised outlier detection: challenges and research questions. ACM SIGKDD Explor 15(1):11–22

Zimek A, Gaudet M, Campello RJGB, Sander J (2013b) Subsampling for efficient and effective unsupervised outlier detection ensembles. In: Proceedings of the 19th ACM international conference on knowledge discovery and data mining (SIGKDD), Chicago, pp 428–436. doi:10.1145/2487575.2487676

Zimmermann A (2014) The data problem in data mining. ACM SIGKDD Explor 16(2):38–45. doi:10.1145/2783702.2783706