On the Internal Evaluation of Unsupervised Outlier Detection

Henrique O. Marques
University of São Paulo
São Carlos, SP, Brazil
hom@icmc.usp.br

Ricardo J. G. B. Campello
University of São Paulo
São Carlos, SP, Brazil
campello@icmc.usp.br

Arthur Zimek
Ludwig-Maximilians-Universität München
Munich, Germany
zimek@dbs.ifi.lmu.de

Jörg Sander
University of Alberta
Edmonton, AB, Canada
jsander@ualberta.ca

ABSTRACT

Although there is a large and growing literature that tackles the unsupervised outlier detection problem, the unsupervised *evaluation* of outlier detection results is still virtually untouched in the literature. The so-called internal evaluation, based solely on the data and the assessed solutions themselves, is required if one wants to statistically validate (in absolute terms) or just compare (in relative terms) the solutions provided by different algorithms or by different parameterizations of a given algorithm in the absence of labeled data. However, in contrast to unsupervised cluster analysis, where indexes for internal evaluation and validation of clustering solutions have been conceived and shown to be very useful, in the outlier detection domain this problem has been notably overlooked. Here we discuss this problem and provide a solution for the internal evaluation of top-n (binary) outlier detection results. Specifically, we propose an index called IREOS (Internal, Relative Evaluation of Outlier Solutions) that can evaluate and compare different candidate labelings of a collection of multivariate observations in terms of outliers and inliers. We also statistically adjust IREOS for chance and extensively evaluate it in several experiments involving different collections of synthetic and real data sets.

Categories and Subject Descriptors

H.2.8 [**Database Applications**]: Data mining

Keywords

Outlier detection, unsupervised evaluation, validation

1. INTRODUCTION

One of the central tasks of data mining is *outlier* or *anomaly* detection, the problem of discovering patterns that are exceptional in some sense. Detecting such patterns is relevant for two main reasons: (i) in some applications, such patterns represent spurious data (e.g., sensor failures or noise) that should be removed in a preprocessing step for further data analysis; or, more importantly, (ii) in many applications, such patterns represent extraordinary behaviors that deserve some special attention, such as genes associated with certain diseases, frauds in financial systems, employees with unusual productivity profiles, or customers with uncommon purchasing patterns.

Outlier detection techniques can be categorized in different ways. For instance, a common distinction is that between the methods that assign binary labels ("outlier" *vs.* "inlier" for those observations deemed anomalous *vs.* normal) and methods that assign a score or rank representing a degree to which an observation is considered to be outlier. Another distinction is that between supervised, semi-supervised, and unsupervised outlier detection techniques [11]. Supervised techniques assume that a set of observed instances labeled as inliers and outliers are available to train a classifier. In the semi-supervised scenario, labeled outliers are not available and only previously known inliers can be used in order to obtain a (one class) classification model. When no labeled data are available at all, it is necessary to use unsupervised techniques, which do not assume any prior knowledge about which observations are outliers and which are inliers.

In this work we focus on unsupervised outlier detection scenarios. In general, an outlier in this context can be described as "*an observation (or subset of observations) which appears to be inconsistent with the remainder of that set of data*" [4]. In this context, there is no generally applicable definition of "appearance of inconsistency"; its formalization rather depends on the application scenario and the detection method to be used. A common scenario is to apply some outlier detection method to a database with N observations, labeling a certain subset of n such observations as the n most likely outliers, while the remaining $N - n$ observations are labeled as inliers. This is referred to as the *top-*n outlier detection problem [1, 2, 5, 9, 15, 20, 23, 28, 29, 32], which is far from trivial, especially when dealing with multivariate data following complex, unknown distributions. Without labeled examples, the main complicating factor in this problem is that the notion of "outlierness" is not precisely and generally defined.

The subjectivity inherent in the unsupervised outlier detection scenario is one of the main reasons why a rich variety of detection methods has been developed, from classic parametric statistical methods [4, 13] to more recent database-oriented approaches con-

SSDBM '15, June 29 - July 01, 2015, La Jolla, CA, USA
© 20XX ACM. 978-1-4503-3709-0/15/06…$15.00
DOI: http://dx.doi.org/10.1145/2791347.2791352

ceived to deal with multivariate, possibly large databases. Considering the latter category, a plethora of detection algorithms has emerged in the past 15 years or so. Examples are DB-Outlier [18, 19], kNN Outlier [2, 32], LOF [7] and its many variants [16,21,22,30,37,41] (see, e.g., the work of Schubert et al. [35] for a discussion of these and many more variants), and ABOD [23], just to mention a few. Each of these algorithms, however, uses its own criterion to judge quantitatively the level of adherence of each observation with the concept of outlier, from a particular perspective. This complicates not only the selection of a particular algorithm and/or the choice of an appropriate configuration of parameters for this algorithm in a practical application, but also the assessment of the *quality* of the solutions obtained, especially in light of the problem of defining a measure of quality that is not tied to the criteria used by the algorithms themselves. These issues are interrelated and refer to the problems of model selection and assessment (evaluation or validation) of results in unsupervised learning. These problems have been investigated for decades in the area of unsupervised data clustering [14], but are rarely mentioned and are virtually untouched in the area of outlier detection [43].

In the data clustering domain, the related problems of evaluation and model selection are tackled by using some kind of quantitative index, called validation criterion [14]. In practice, when labels are not available, *internal* validation indexes can be used. These indexes are called internal as they do not make use of any external information (such as class labels) in the evaluation of a solution. Instead, internal indexes measure the quality of an obtained clustering solution based only on the solution and the data objects. Most such indexes are also *relative* in the sense that they can be employed to compare different clustering solutions pointing out which one is better in relative terms. Therefore they can also be used for model selection. Internal, relative indexes have been shown to be effective and useful tools for the unsupervised clustering evaluation and model selection tasks — e.g. see [10, 26, 39, 40] and references therein.

The areas of clustering and outlier detection are related to each other and, from a certain perspective, they can even be seen as two sides of the same coin. In fact, when referring to an outlier as "*an observation which deviates so much from other observations as to arouse suspicions that it was generated by a different mechanism*", as in Hawkins' [13] definition, it is implicitly assumed that there are one or more mechanisms responsible for generating the normal, "unsuspicious" observations. Clusters are possible candidates to model such mechanisms. Surprisingly, although the internal evaluation problem has been extensively studied in data clustering, it has been completely neglected in outlier detection. In this paper we step towards bridging this gap by proposing an internal, relative evaluation measure for unsupervised outlier detection. We start from a definition of "outlierness" that is not tied to any particular criterion used by outlier detection algorithms. Rather, it follows the same, common intuition as a multitude of these algorithms and criteria: an outlier is an observation that is to some extent farther away and can therefore be more easily separated from other observations than an inlier. We formulate separability for outlier detection in an objective and principled way, leading to a natural definition of the proposed index.

In summary, we make the following contributions in this paper:

- We introduce the first internal, relative validation measure for evaluation of outlier detection results, IREOS (Internal, Relative Evaluation of Outlier Solutions). In its most general form, IREOS can evaluate given solutions in polynomial time, but searching the solution space using the measure itself as the basis for a detection algorithm would, in principle, not be computationally feasible in the general case (which is a common property of truly independent measures for evaluation).
- We propose furthermore an improved version of IREOS that is adjusted for chance by removing from the index the theoretical offset that is expected to be observed when evaluating random solutions. In addition to the adjusted index, we also devise means to return p-values with respect to the null hypothesis of a random solution.
- Since the exact procedure to adjust the index for chance can be computationally demanding for large data sets, we also provide a faster version of the proposed procedure, based on Monte Carlo experiments.
- We extensively evaluate IREOS using different collections of synthetic and real data sets, both in controlled experiments as well as in practical experiments of model selection.

The remainder of this paper is organized as follows. In Section 2, we discuss the typical approaches for *external* evaluation and the different requirements and use cases for *internal* evaluation. In Section 3, we introduce IREOS, discussing requirements and solutions, adjustment for chance, statistics, and algorithmic properties. We evaluate this index in Section 4 and conclude the paper in Section 5.

2. RELATED WORK

In the literature so far, the evaluation of results in unsupervised outlier detection has been mostly restricted to controlled experiments in research papers that make use of labeled data sets to evaluate how algorithms compare to other algorithms when trying to assess, in an unsupervised way, observations previously known to be inliers or outliers according to a particular intuition or semantic (e.g., normal patients versus patients with an uncommon pathology). In this scenario, referred to as external evaluation or validation, the labels are not used by the algorithms, but rather to assess their results only [43,46].

For the external evaluation of a top-n outlier detection solution, one is given a data set with n known outliers (ground truth) as well as the observations ranked top-n by the given solution. *Precision-at-n* (prec@n for short) measures the fraction of the true outliers (i.e., labeled in the ground truth as outlier) among the top-n objects of the given solution [8]. If an outlier ranking is to be evaluated beyond the top-n ranks, one could also decide to measure prec@$2n$, prec@$3n$, or precision at some other point in the ranking, but the typical choice is to use the number of true outliers as cutoff value for measuring precision [43]. A common alternative is the Receiver Operating Characteristic (ROC), which compares the candidate ranking against the binary ground truth by plotting the true positive rate against the false positive rate. Variants of these measures and more in-depth considerations about the external evaluation of unsupervised outlier detection results have been discussed, e.g., by Schubert et al. [34].

For internal evaluation, which is the focus of our paper, we are not aware of the existence of any internal validation index for unsupervised outlier detection. This has been noted as a gap in the literature with respect to the development of advanced ensemble selection methods [43], but the potential applications of internal measures are far more diverse. Most fundamental is the practical application of outlier detection methods where users would benefit from unsupervised estimates of the quality of a solution provided by some method. After all, the availability of labeled data required by external evaluation measures is not consistent with the premises of unsupervised learning, and the commonly practiced external evaluation of unsupervised outlier detection algorithms makes sense only when comparing performances of algorithms in controlled experiments.

3. INTERNAL EVALUATION OF OUTLIER DETECTION

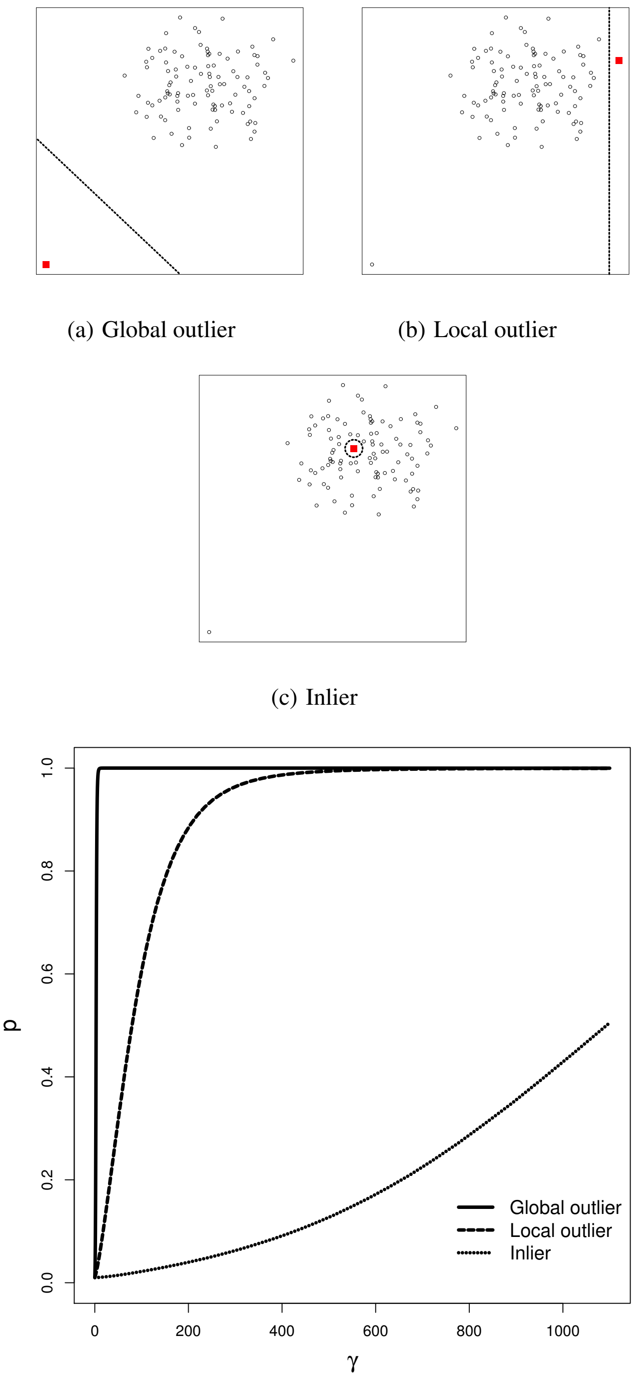
3.1 Problem Statement

Let $\mathbf{X} = \{\mathbf{x}_1, \cdots, \mathbf{x}_N\}$ be an unlabeled data set containing N d-dimensional feature vectors, $\mathbf{x}_i$, and assume that one or more unsupervised outlier detection algorithms will produce, for this data set, candidate solutions of top-n outliers, which one wants to evaluate in the absence of labels. Formally, a solution can be seen as a subset $\mathbf{S} \subset \mathbf{X}$, $|\mathbf{S}| = n$, containing the objects labeled as outliers. Given a collection of such candidate solutions, we want to independently and quantitatively measure the quality of each individual candidate solution, e.g., in order (i) to assess their statistical significance when compared to the null hypothesis of a random solution; or (ii) to compare them in relative terms so that the best candidates, corresponding to more suitable models (algorithms, parameters), can be selected.

3.2 Preliminary Attempt: A Baseline Index

We start from the intuition that an outlier is an observation that is to some extent farther off and can therefore be more easily separated (discriminated) from other observations than an inlier. The labeling of n data objects as outliers corresponding to a better (worse) unsupervised outlier detection solution $\mathbf{S}$ is expected to be more (less) according to this intuition. The basic problem is then to quantify how easy or difficult it is to separate each object $\mathbf{x}_i \in \mathbf{S}$ from other objects. In a good solution $\mathbf{S}$, consisting mostly of genuine outliers correctly detected by some method, the average degree of separability is expected to be high, whereas in a poor solution containing many false positives this average degree of separability should be lower.

We propose to assess the separability of individual data objects using a classifier. We advocate the use of a maximum margin classifier [36, 42], as this type of classifier is able to quantify how distant each object is from the decision boundary while trying to maximize the margin of separability between this boundary and the instances of different classes. This idea is illustrated in Figure 1. Figures 1(a), 1(b), and 1(c) highlight different objects labeled as an outlier (red square) in different hypothetical outlier detection solutions. In Figure 1(a), the highlighted object, a genuine global outlier, is far away from a maximum margin classification boundary (dashed line) that discriminates it from the other objects. In Figure 1(b), the highlighted object is arguably a local outlier (w.r.t. the neighboring cluster) and the margin is narrower but still wider than that in Figure 1(c). In the case of Figure 1(c), the highlighted object is undoubtedly an inlier and not only the margin is very narrow but also the decision boundary needs to be nonlinear (i.e., more complex).


(a) Global outlier

(b) Local outlier

(c) Inlier

(d) Separability

Figure 1: Illustrative data set: (a − c) three different objects labeled as outliers; (d) curves of separability for a maximum margin classifier for each of these labeled outliers.

The fact that the decision boundary needs to be nonlinear to separate certain objects (as in the example in Figure 1(c)) implies that a nonlinear maximum margin classifier is required for our purpose, such as Nonlinear SVMs or Kernel Logistic Regression [36, 42]. These classifiers use a kernel function to transform the original (possibly non-linearly separable) problem into a linearly separable one. One of the most effective and popular kernel functions is the radial basis kernel, given by:

$$K(\mathbf{x}_i, \mathbf{x}_j) = e^{-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2}.$$

The term γ, which is inversely proportional to the width of such a Gaussian-shaped kernel, is positively related to the flexibility (degree of nonlinearity) of the decision boundary of the corresponding classifiers. In other words, the discrimination capacity of a kernel-based classifier is positively dependent on γ. As a special case it approaches a linear classifier as γ approaches zero. The effect of γ is similar to that of the order of a polynomial kernel function, starting from linear for first order and getting more and more non-linear as the order increases.

In practical classification tasks, γ can be used to control the compromise between the performance of the classifier on the training data versus on test data. Here, however, we are not interested at all in the classifier itself or its performance on new, unseen data. We use a classifier merely to measure the degree of difficulty when trying to discriminate between one individual data object and the other data objects. The key observation to achieve this without having to specify a particular value for γ as a parameter is that our original premise tends to hold true, to a lesser or greater extent, no matter the value of γ. In other words, the fundamental assumption "*the more outlierish an object is, the easier is it to discriminate from others*" is expected to be observed for different values of γ, although the contrast between easier and more difficult cases may change. This is illustrated in Figure 1(d). We vary the value of γ from zero up to a maximum value γ_{max} (for which all the objects labeled as outliers (a, b, c) can be individually discriminated from all the others by using a kernel-based classifier). The values along the curves (vertical axis) stand for a measure $p(\mathbf{x}_j, \gamma)$ that quantifies in a normalized interval how far each object $\mathbf{x}_j$ is from the decision boundary. For all values of γ, the two outliers are distinctly farther away from the decision boundary than the inlier.

Thus, we do not need to choose a particular value of γ. Instead, we can measure the overall separability of an object $\mathbf{x}_j$ by computing the *area under the curve* (AUC) over the interval of γ values, i.e.,

$$\int_{\gamma=0}^{\gamma_{\text{max}}} p(\mathbf{x}_j, \gamma).$$

Our final goal is, though, to evaluate the separability across the *collection* of data objects labeled as outliers in a given solution $\mathbf{S}$. We therefore take the average curve of separability for those objects in $\mathbf{S}$, i.e.,

$$\bar{p}(\gamma) = \frac{1}{n} \sum_{\mathbf{x}_j \in \mathbf{S}} p(\mathbf{x}_j, \gamma),$$

and then compute the area under this curve to get a single number, i.e.,

$$\int_{\gamma=0}^{\gamma_{\text{max}}} \bar{p}(\gamma).$$

This value can be trivially normalized in $[0, 1]$ by dividing it by its maximum possible value, γ_{max}, thus giving rise to a first, preliminary index,

$$I(\mathbf{S}) = \frac{1}{\gamma_{\text{max}}} \int_{\gamma=0}^{\gamma_{\text{max}}} \bar{p}(\gamma).$$

As in practice classifiers need to be trained to compute $\bar{p}(\gamma)$ for each γ, we discretize the interval $[0, \gamma_{\text{max}}]$ into a finite number of values for γ, from $\gamma_1 = 0$ to $\gamma_{n_\gamma} = \gamma_{\text{max}}$. A baseline index can thus be computed (within $[0, 1]$) as:

$$I(\mathbf{S}) = \frac{1}{n_\gamma} \sum_{l=1}^{n_\gamma} \left(\frac{1}{n} \sum_{\mathbf{x}_j \in \mathbf{S}} p(\mathbf{x}_j, \gamma_l) \right). \tag{1}$$

3.3 IREOS Index

Intuitions Missing in the Baseline Index: Our preliminary, baseline index introduced in Section 3.2 may work satisfactorily in various application scenarios. Conceptually, however, it does not capture two basic intuitions that we judge important in the realm of outlier detection. Both are related to the possible presence of *clumps* of data objects in the data set. Clumps, or particles, are subsets of objects lying in the same region of the data space, relatively closer to each other than they are from other objects, but too small to be deemed a cluster. They may exist for different reasons, mainly: (i) just by chance, e.g., in data sets with background noise following a Poison process; or (ii) as a result of anomalies whose instances are relatively rare but tend to be somewhat similar to each other, e.g., some genetic mutations or certain types of frauds. Although the semantics behind the possible interpretation of such clumps as outliers would be different, namely, noise in the first case and micro-clusters in the second, in both cases the analyst may not want to miss the corresponding objects as potential outliers for further investigations.

In principle, an issue with the idea of considering clumps as part of our evaluation model is that the interpretation of this concept may depend strongly on both the application domain and the users' personal expectations. The point is that, without a mechanism that allows different users in varied application scenarios to explicitly express what they judge "too small" to be interpreted as a cluster, an evaluation measure will end up being hooked on a single, rigid, and very particular evaluation perspective. Therefore, in contrast to the common practice of avoiding any parameters in evaluation indexes for unsupervised learning, here we advocate that for outlier detection it is actually important to provide the users with an optional control mechanism to adjust their expectations about clump sizes. Given a certain expectation about what a maximum clump size should be, and beyond what the user believes a somewhat isolated group of objects is more of a cluster nature rather than a clump of potential outliers, we support that an evaluation index should be able to differentiate between weak candidate outliers as objects inside clusters from moderate candidate outliers as objects in isolated clumps (and these from strong candidate outliers in the form of isolated objects). This is the first intuition that is missing in our baseline index as it was defined in Section 3.2. In order to capture this intuition, we define a maximum clump size, m_{cl}, as an optional control parameter for exploratory data analysis, to be incorporated in our index.

A second, related intuition is not captured by the preliminary index either. While it is clear that the evaluation of each object labeled as an outlier and, accordingly, the whole index, should be negatively affected by the presence of other objects nearby (e.g., in a clump), it is intuitive that such a negative impact should be more severe if the nearby objects are assigned a different label (i.e., they are actually deemed inliers). Consider the example in Figure 2, which corresponds essentially to the same data set as Figure 1 except for an additional object placed near the global outlier in the left bottom corner. The difference between the subfigures respectively). On the left, this label appears to be inconsistent with the label of the origi-

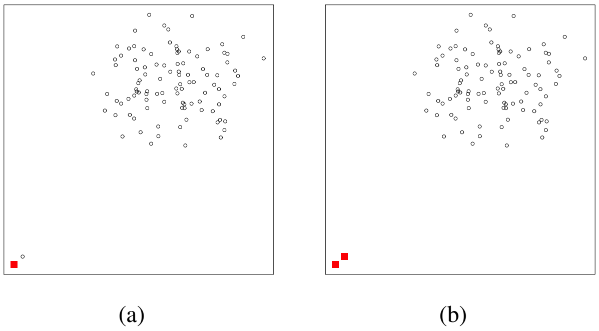


Figure 2: Data set of Fig. 1 with an additional object at the lower left corner: (a) labeled inlier; (b) labeled outlier.

nal object close by, and the index should be negatively affected. On the right, even though the original object is less of an outlier now in the presence of the additional object, their common labeling as outliers is more consistent as both objects can be seen as a clump, so the negative impact of the presence of the new object should be smaller.

Incorporating the Missing Intuitions: In order to capture both desired intuitions, we propose the use of classifiers with *soft margins*, as it is common practice both in the literature as well as in real-world applications when it comes to maximum margin classifiers, such as SVMs and Kernel Logistic Regression [36, 42]. By making use of a soft margin, these classifiers allow the misclassification of objects at the price of a penalty term P_t that is incorporated into the original objective of margin maximization. Such a term is typically in the form

$$P_t = C \sum_{j=1}^{N} \xi(\mathbf{x}_j),$$

where C is a constant and $\xi(\mathbf{x}_j)$ stands for the individual penalty component associated with object $\mathbf{x}_j$. The farther an object $\mathbf{x}_j$ is from the margin boundary on the wrong side, the greater the value of $\xi(\mathbf{x}_j)$.[1] The constant C controls the overall cost of penalties. In classification problems, this is a key parameter used to adjust the compromise between under- and overfitting. Here, since we are not interested in the performance of the classifiers for new data, this constant is not critical and should only be big enough so that the objects labeled as outliers in solution $\mathbf{S}$ can be discriminated from others by these classifiers when following the procedure to compute Equation (1). Indeed, as we will see in Section 4, results and conclusions drawn from such results are very stable across many data sets for values of C varying several orders of magnitude.

Soft margin classifiers allow the use of a generalized penalty component that can assign different costs to different objects, rather than a single, uniform cost C. Thus we can assign full cost C to the objects labeled as inliers yet only a fraction $\beta \in [0, 1]$ of C to the objects labeled as outliers, i.e.,

$$P_t = \sum_{j=1}^{N} C(\mathbf{x}_j)\xi(\mathbf{x}_j),$$

where $C(\mathbf{x}_j) = C$ or $\beta \cdot C$ depending on the label of $\mathbf{x}_j$ (*inlier* respectively *outlier*). For $\beta = 1$, the method reduces to the ordinary case where objects are treated equally no matter their labels. In the other extreme, $\beta = 0$, objects labeled as outliers can be misclassified for free (notice that, when evaluating the separability of a specific object, this is equivalent to removing all other objects labeled as outliers from the data set). The choice of β would therefore tune the influence of other objects depending on their assigned labels and, thus, address our **second desired intuition**.

In order to capture our **first intuition**, the modeling of possible clumps by defining a maximum clump size, m_{cl}, we can set the fraction of the penalty C as $\beta = 1/m_{\text{cl}}$. This way, we are left with m_{cl} as a single, optional control parameter in our evaluation method. It is optional because by setting $m_{\text{cl}} = 1$, the method reduces to the particular case where clumps are not modeled and the same, full penalty cost is assigned to all objects. As m_{cl} increases, objects labeled as outliers in a clump will individually affect less and less each other's measure of separability, and a larger number of nearby objects will be needed to get a certain negative impact. Notice that, by setting $\beta = 1/m_{\text{cl}}$, one needs m_{cl} objects labeled as outliers to get the same impact as a single inlier. Also, notice that for a top-n detection problem, it would be contradictory to set $m_{\text{cl}} > n$, as no more than n objects can be labeled as outliers. By considering this conceptual upper bound, one gets $1 \leq m_{\text{cl}} \leq n$. Except when $m_{\text{cl}} = 1$, the separability of each object in the general case depends on the labels of the other objects and, therefore, seeking a solution that maximizes the proposed index (rather than using it to assess a given solution) would hardly be computationally feasible: in principle, it would demand an exhaustive search in a space of size $\binom{N}{n}$, where typically $n \ll N$.

Summary and Algorithm: In brief, IREOS is summarized as follows: like the baseline index (Section 3.2), IREOS is also computed using Equation (1). However, we make use of classifiers with soft margins in order to compute the terms $p(\mathbf{x}_j, \gamma_l)$ in that equation, where the full penalty is assigned by the used classifier to those objects labeled as inliers and only a fraction $1/m_{\text{cl}}$ of the full penalty is assigned to those objects labeled as outliers.

A high level pseudo code for computing IREOS for a set $\mathbf{\Omega}$ of multiple top-n outlier detection solutions $\mathbf{S}$ (as, e.g., for model selection) is given in Algorithm 1.

As for the classifier to be used in practice, our method is not hooked on any specific soft margin classifier. Kernel Logistic Regression (KLR) [42], which we have used for all the experiments reported in this paper,[2] offers the following advantages: (i) it automatically provides $p(\mathbf{x}_j, \gamma_l)$ as the probability that object $\mathbf{x}_j$ belongs to the positive (outlier) class; (ii) these terms are not only provided directly as a byproduct of the classifier, but they are naturally normalized (as probabilities) within $[0, 1]$; and (iii) KLR is a classifier known to be robust even in the presence of imbalanced classes and small amounts of training data.

[1] For SVMs, $\xi(\mathbf{x}_j)$ is zero when $\mathbf{x}_j$ lies on the correct side of the margin boundary. For Kernel Logistic Regression, all objects (rather than only support vectors) can influence the decision boundary and $\xi(\mathbf{x}_j)$ can be non-null but tending to zero as $\mathbf{x}_j$ moves away from the margin boundary on the correct side.

[2] Our code is available upon request.

Algorithm 1 IREOS

```
procedure IREOS($\mathbf{X}$, $\mathbf{\Omega}$, $m_{\text{cl}}$)
    $\gamma_{\text{max}}$ = value of $\gamma$ needed to separate from the other objects
    every object labeled as an outlier in all $\mathbf{S} \in \mathbf{\Omega}$
    setOfGammas = $[0, \gamma_{\text{max}}]$ discretized into $n_\gamma$ values
    for all ($\mathbf{S} \in \mathbf{\Omega}$) do
        for all ($\gamma \in$ setOfGammas) do
            for all ($\mathbf{x}_j \in \mathbf{S}$) do
                prob[$\mathbf{x}_j$] = Classifier($\mathbf{X}$, $\mathbf{x}_j$, $\mathbf{S}$, $m_{\text{cl}}$, $\gamma$)
            end for
            avgProb[$\gamma$] = Average(prob)
        end for
        ireos[$\mathbf{S}$] = NormAUC(avgProb, setOfGammas)
    end for
end procedure
```

3.4 Adjustment for Chance

The IREOS index as described above is ready to be used in practice if one is only interested in comparing in *relative terms* a set of different candidate solutions, e.g. for model selection. However, the interpretation of the index for individual solutions, e.g. for statistical validation, can be very misleading. The reason is that IREOS will provide a certain positive value even when evaluating purely random solutions. To make things worse, such a value is data dependent. In fact, note from Figure 1(d) that even inliers will exhibit a non null value for the AUC of separability. This prevents interpreting and assessing the statistical relevance of a given result in *absolute terms*, which requires the index to be adjusted for chance. Here, we follow the classic statistical framework for chance adjustment, i.e.,

$$I_{\text{adj}}(\mathbf{S}) = \frac{I(\mathbf{S}) - E\{I\}}{I_{\text{max}} - E\{I\}}, \tag{2}$$

where $I_{\text{adj}}(\mathbf{S})$ is the resulting (adjusted) index, $I(\mathbf{S})$ is the original index (Eq. 1), $I_{\text{max}} = 1$ is the maximum value that the index can take, and $E\{I\}$ is its expected value assuming that the n data objects labeled as outliers in a solution are chosen randomly. For random solutions, I_{adj} is expected to take values around zero. The maximum is still 1, but the index now can take negative values to indicate solutions even worse than what one would expect to obtain by chance.

Exact Computation: Term $E\{I\}$ in Equation (2) is given from Equation (1) as

$$E\{I\} = \frac{1}{n_\gamma} \sum_{l=1}^{n_\gamma} E\{\bar{p}(\gamma_l)\}, \tag{3}$$

where

$$\bar{p}(\gamma_l) = \frac{1}{n} \sum_{\mathbf{x}_j \in \mathbf{S}_R} p(\mathbf{x}_j, \gamma_l)$$

is a random variable associated with random solutions $\mathbf{S}_R$.

In the following we show that the expectation $E\{\bar{p}(\gamma_l)\}$ (and, accordingly, $E\{I\}$) can be computed in an exact way for the basic setup where $m_{\text{cl}} = 1$. In fact, recall that, when $m_{\text{cl}} = 1$, the classifiers just try to discriminate between each candidate outlier $\mathbf{x}_j$ and the other objects, no matter their labels. This means that $p(\mathbf{x}_j, \gamma_l)$ depends only on the data, not on any particular realization $\mathbf{S}_R$ of possible candidate outliers, and therefore it can be independently precomputed for each object $\mathbf{x}_j \in \mathbf{X}$ and γ_l ($l = 1, \cdots, n_\gamma$). Recalling that $|\mathbf{S}_R| = n$, it then follows that

$$\begin{aligned} E\{\bar{p}(\gamma_l)\} &= \frac{1}{n} \sum_{\mathbf{x}_j \in \mathbf{S}_R} E\{p(\mathbf{x}_j, \gamma_l)\} \\ &= E\{p(\mathbf{x}_j, \gamma_l)\}. \end{aligned}$$

This is an instance of the well-known result that the expected value for the mean of an i.i.d. sample of size n is the mean of the population. Here, for a given γ_l, our (finite) population consists of the N precomputed values $p(\mathbf{x}_j, \gamma_l)$ for all data objects $\mathbf{x}_j$ in the database $\mathbf{X}$. Taking their average gives the exact value for $E\{\bar{p}(\gamma_l)\}$, i.e.:

$$\begin{aligned} E\{\bar{p}(\gamma_l)\} &= E\{p(\mathbf{x}_j, \gamma_l)\} \\ &= \frac{1}{N} \sum_{\mathbf{x}_j \in \mathbf{X}} p(\mathbf{x}_j, \gamma_l). \end{aligned} \tag{4}$$

Statistical Validation: The variance is not needed for the adjustment for chance in Equation (2), but it can be useful for statistical validation when this type of validation is required. Since our index is given by a sum of random variables $\frac{1}{n_\gamma}\bar{p}(\gamma_l)$ over γ_l, we can compute the variance of the index as

$$\begin{aligned} \text{Var}\{I\} &= \text{Var}\left\{\tfrac{1}{n_\gamma} \textstyle\sum_{l=1}^{n_\gamma} \bar{p}(\gamma_l)\right\} \\ &= \tfrac{1}{n_\gamma^2} \textstyle\sum_{l_1,l_2=1}^{n_\gamma} \text{Cov}\left(\bar{p}(\gamma_{l_1}), \bar{p}(\gamma_{l_2})\right), \end{aligned} \tag{5}$$

which can also be rewritten equivalently as

$$\begin{aligned} \text{Var}\{I\} =& \frac{1}{n_\gamma^2} \sum_{l=1}^{n_\gamma} \text{Var}\{\bar{p}(\gamma_l)\} \\ &+ \frac{2}{n_\gamma^2} \sum_{l_1=1}^{l_2-1} \sum_{l_2=2}^{n_\gamma} \text{Cov}\left(\bar{p}(\gamma_{l_1}), \bar{p}(\gamma_{l_2})\right). \end{aligned}$$

This latter, equivalent form emphasizes the possible lack of independence of $\bar{p}(\gamma_l)$ over γ_l, specifically when the second term is not null. For the first term, it follows that

$$\begin{aligned} \text{Var}\{\bar{p}(\gamma_l)\} &= \frac{1}{n^2} \sum_{\mathbf{x}_j \in \mathbf{S}_R} \text{Var}\{p(\mathbf{x}_j, \gamma_l)\} \\ &= \frac{1}{n} \text{Var}\{p(\mathbf{x}_j, \gamma_l)\}, \end{aligned}$$

i.e., the variance of the sample mean is the variance of the population over the sample size. Analogously, for the covariance one has

$$\text{Cov}\left(\bar{p}(\gamma_{l_1}), \bar{p}(\gamma_{l_2})\right) = \frac{1}{n} \text{Cov}\left(p(\mathbf{x}_j, \gamma_{l_1}), p(\mathbf{x}_j, \gamma_{l_2})\right), \tag{6}$$

which can be exactly computed once we have precomputed the whole population $p(\cdot, \cdot)$.[3]

Provided that the sample size is not critically small, the Central Limit Theorem (CLT) ensures that, for each γ_l, the sample mean $\bar{p}(\gamma_l)$ follows at least approximately a Normal distribution, i.e.,

$$\bar{p}(\gamma) \sim \mathcal{N}\left(E\{\bar{p}(\gamma)\}, \text{Var}\{\bar{p}(\gamma)\}\right).$$

[3] Since the population is of a finite size (N), though, when considering sampling without replacement and sample sizes n significantly large w.r.t. N (e.g., more than 5%), a finite population correction factor $(N - n)/(N - 1)$ can be used to adjust the computed variance [38].

Algorithm 2 Chance Adjustment — Exact Version ($m_{\text{cl}} = 1$)

```
procedure ChanceAdjustment($\mathbf{X}$, $I(\mathbf{S})$, $\gamma_{\text{max}}$, $n$)
    setOfGammas = $[0, \gamma_{\text{max}}]$ discretized into $n_\gamma$ values
    for all ($\gamma_l \in$ setOfGammas) do
        for all ($\mathbf{x}_j \in \mathbf{X}$) do
            $p(\mathbf{x}_j, \gamma_l)$ = Classifier($\mathbf{X}$, $\mathbf{x}_j$, $\gamma_l$)
        end for
        $E\{\bar{p}(\gamma_l)\}$ = AvgOverDB$(p(\cdot, \gamma_l))$                ▷ Eq. (4)
    end for
    $E\{I\}$ = AvgOverGamma$(E\{\bar{p}(\cdot)\})$                ▷ Eq. (3)
    $I_{adj}$ = IndexAdjustment$(I(\mathbf{S}), E\{I\})$                ▷ Eq. (2)
    for all ($\gamma_{l_1}, \gamma_{l_2} \in$ setOfGammas) do
        $\mathrm{Cov}(\bar{p}(\gamma_{l_1}), \bar{p}(\gamma_{l_2})) = \mathrm{Cov}(p(\cdot, \gamma_{l_1}), p(\cdot, \gamma_{l_2}))\ /n$
                                                                ▷ Eq. (6)
    end for
    $\mathrm{Var}\{I\}$ = AvgOverGammas$(\mathrm{Cov}(\bar{p}(\cdot), \bar{p}(\cdot)))$    ▷ Eq. (5)
    $p$Value = $z$-test($I(\mathbf{S})$, $E\{I\}$, $\mathrm{Var}\{I\}$)
end procedure
```

This means that, for random solutions $\mathbf{S}_R$, our index in Equation (1) is given by a sum of normally distributed variables $\frac{1}{n_\gamma}\bar{p}(\gamma_l)$ over γ_l. The sum of normally distributed random variables

$$X \sim \mathcal{N}(\mu_X, \sigma_X^2)$$

and

$$Y \sim \mathcal{N}(\mu_Y, \sigma_Y^2)$$

is also normally distributed, i.e.,

$$X + Y \ \sim \ \mathcal{N}(\mu_X + \mu_Y, \sigma_{X+Y}^2),$$

where [33]

$$\sigma_{X+Y}^2 = \sigma_X^2 + \sigma_Y^2 + 2\,\mathrm{Cov}(\sigma_X, \sigma_Y).$$

This leads to the very important result that our index IREOS, as a sum of sample means, will follow at least approximately a Normal distribution according to the CLT, i.e., $I \sim \mathcal{N}(E\{I\}, \mathrm{Var}\{I\})$, with mean $E\{I\}$ and variance $\mathrm{Var}\{I\}$ computed in an exact way as described above.

Since we know such a mean and variance for the population, we thus can go beyond the ordinary adjustment for chance (Eq. 2) and perform statistical validation as well. Particularly, if we are given a certain outlier detection solution, $\mathbf{S}$, and the corresponding value for our adjusted index, $I_{\text{adj}}(\mathbf{S})$, we can assess the statistical significance of $I_{\text{adj}}(\mathbf{S})$ by means of a z-test. In this case, a p-value can be trivially computed based on the Normal assumption by contrasting $I_{\text{adj}}(\mathbf{S})$ against the null hypothesis of a random solution [38]. The computation of the adjustment for chance of the IREOS index is summarized in Algorithm 2.

Approximate Computation via Monte Carlo: The exact computations described above presume $m_{\text{cl}} = 1$. For different evaluation setups, $p(\mathbf{x}_j, \gamma_l)$ can no longer be independently precomputed for each object $\mathbf{x}_j \in \mathbf{X}$, as the separability of a given object as assessed by the classifiers now depends also on the labels assigned to the other objects of the data set, for each possible random solution. This means that, for a given γ_l, the size of our finite population expands from N to $\binom{N}{n}$ and, as such, it can easily become intractable for exhaustive computations. Even when $m_{\text{cl}} = 1$, precomputing N terms $p(\mathbf{x}_j, \gamma_l)$ for each γ_l (i.e., $N \cdot n_\gamma$ in total) may be computationally prohibitive for large databases as well, as each term demands to train an independent classifier.

To make adjustment for chance and statistical validation feasible when $m_{\text{cl}} > 1$ or N is large, we can use Monte Carlo simulations in order to estimate statistics rather than trying to compute them in an exact and exhaustive way. The idea is to sample a number n_{MC} of random outlier detection solutions whereby the desired statistical moments can be estimated. In particular, the expected value in Equation (2) can be directly estimated from the sample.

When statistical validation is on the agenda as well, we also need to estimate the baseline distribution under the null hypothesis. There are different alternatives. If the normality assumption is evoked from the CLT, a parametric approach is possibly based on a t-student distribution with the sample estimates for the mean and variance (i.e., a t-test, which is known to be robust even when normality is not fully satisfied [6]). Alternatively, p-values can be directly derived from observed histograms in a non-parametric way [14].

The sample size, n_{MC}, clearly represents a trade-off between computational burden and accuracy. Larger (smaller) samples lead to more (less) accurate estimations yet from a larger (smaller) number of trained classifiers. Rather than setting the value for n_{MC} arbitrarily, one can also determine n_{MC} automatically, by specifying (i) a certain significance level as the probability that the sample mean will fall within, and (ii) a prespecified confidence interval around the population mean [38].

3.5 Complexity

The asymptotic computational complexity of the algorithm depends on the complexity of the classifier, $O(f(N, d))$, as a function of the database size N and dimensionality d. For each candidate solution $\mathbf{S}$, we need to compute IREOS (Equation 2), which demands training $n \cdot n_\gamma$ classifiers, thus resulting in an overall complexity of $O(n \cdot n_\gamma \cdot f(N, d))$. When the index is adjusted for chance, we need to evaluate n_{MC} different random solutions in Monte Carlo simulations in order to estimate the expected index, leading to a complexity of $O(n_{\text{MC}} \cdot n \cdot n_\gamma \cdot f(N, d))$. This is the complexity in the most general case. Using for instance the KLR classifier, which runs in $O(d \cdot N^3)$, we obtain the overall complexity of $O(n_{\text{MC}} \cdot n \cdot n_\gamma \cdot d \cdot N^3)$. While the question of how to reduce this complexity is an interesting direction for further investigations, we argue that the current complexity is not very critical, since every single classifier can be trained in a completely independent way. In the case where we have $O(n_{\text{MC}} \cdot n \cdot n_\gamma)$ computer cores for parallel processing, the complexity of IREOS reduces to that of the classifier used. In other words, IREOS is highly parallelizable and can be implemented in a straightforward way in distributed (e.g., cloud) environments using parallel computing frameworks such as MapReduce.

4. EVALUATION

4.1 Datasets

We combine different strategies to annotate datasets used for evaluation, following statistical considerations, following the semantical notion of unusual classes, or following common procedures and examples from the literature.

Synthetic Datasets: For experiments on synthetic data, we use collections of previously published benchmarking datasets. The first collection (60 datasets) has been designed and used to evaluate outlier detection methods [44, 45]. This dataset collection is split into two independent sets of 30 synthetic datasets each (batch1 and batch2). The datasets vary in the dimensionality $d \in [20, \dots, 40]$, in the number of clusters $c \in [2, \dots, 10]$, and for each cluster independently in the number of points $n_{c_i} \in [600, \dots, 1000]$. For each cluster, the points are generated following a Gaussian model with randomly selected parameters that are attribute-wise independent. The sampled cluster is randomly rotated and the covariance matrix is rotated accordingly. Based on the covariance matrix, the Mahalanobis distance between the mean of a cluster and each cluster point is computed. The distribution of the Mahalanobis distances follows a χ^2 distribution with d degrees of freedom. Those points that exhibit a distance to their cluster center larger than the theoretical 0.975 quantile were labeled as outliers, independently of the actually occurring Mahalanobis distances of the sampled points. This results in an expected amount of 2.5% outliers per dataset.

The second collection of synthetic data consists of the 40 2-dimensional datasets provided by Handl et al. [12], comprising examples with 4, 10, 20, and 40 clusters. The size of the clusters is uniformly chosen from $[50, 500]$ for the datasets with 4 and 10 clusters and from $[10, 100]$ for the datasets with 20 and 40 clusters. This dataset collection has been designed to evaluate clustering results. To label outliers, we followed the same procedure as Zimek et al. [45] (described above), however, for these datasets the original cluster covariance matrices were unknown and therefore calculated from the data.

Real World Datasets: In addition to the synthetic datasets, we use 11 publicly available real world datasets. All of them are available from the UCI repository [3]. For *Annthyroid*, *Diabetes* and *Ionosphere*, we use the version preprocessed for evaluation of outlier detection by Keller et al. [17]. For *Isolet*, *Multiple Features* and *Optical Digits*, we follow the same procedure as performed by Pham and Pagh [31] (for each of these three datasets independently), where all observations from some classes having common behaviors were labeled as inliers and observations of another class were labeled as outliers. In *Isolet*, the classes C, D, and E that share the 'e' sound were selected as inliers and 10 observations from class Y were selected as outliers. *Multiple Features* and *Optical Digits* consist of data representing handwritten numerals (0 - 9). Classes 6 and 9 of *Multiple Features*, and classes 3 and 9 of *Optical Digits* were selected as inliers because of the similarity in shape; and for both datasets 10 observations of class 0 were selected as outliers. In *Lymphography*, classes 1 and 4 are jointly considered as outliers, following the common use of this dataset in the literature [24, 27, 45]. For preprocessing *Shuttle*, we follow the procedure of Zhang et al. [41], using classes 1, 3, 4, 5, 6, and 7 as inliers and class 2 as outlier, and selecting 1000 inliers vs. 13 outliers. Following Micenková et al. [25], we adjust *Vowel* for outlier detection by choosing class 0 as inliers and selecting one instance from each of the remaining classes as outliers. The *Wisconsin Breast Cancer (WBC)* dataset distinguishes cancer types as benign (inliers) or malignant (outliers). Instances with missing values were removed. The outlier class was downsampled to 10 outliers, following the procedure of Schubert et al. [34]. The *Wisconsin Diagnostic Breast Cancer (WDBC)* dataset describes nuclear characteristics for breast cancer diagnosis, also distinguishing cancer types as benign (inliers) or malignant (outliers). We follow the preprocessing of Zhang et al. [41], downsampling the outlier class to keep only 10 outliers.

To make values of different attributes comparable, we also normalized the dataset (that did not already have normalized attribute values) by applying min-max normalization to each attribute independently, so all attribute values fall into the range of $[0, 1]$; we also removed duplicates from datasets that contained duplicate entries (Annthyroid, Multiple Features, and WBC).

4.2 Methods and Measures

In our experiments we evaluate results by contrasting the recommendations made by our index IREOS against the ground truth, i.e., against the labels as provided in the datasets (notice that these labels are not used by our index in any way). Therefore, we set n (the number of outliers in the solutions to be evaluated) to the number of outliers according to the ground truth. We then study the relationship between the quality assessments of the solutions with respect to the ground truth and the quality assessments of the solutions computed by IREOS. To assess the quality of a given solution with respect to the ground truth we compute *Precision at* n (prec@n).[4]

We perform two main types of experiments. The **first type** is a controlled experiment in which we produce, for a given dataset, a collection of candidate outlier detection solutions with prec@n varying from 1 to zero. We start from the perfect solution given by the ground truth and iteratively produce new solutions replacing one of the true outliers with a random inlier. This way, at each iteration prec@n is reduced by one unit and we get a diverse collection with n solutions to be evaluated. We then measure the goodness of fit between this ranking of solutions (with decreasing quality w.r.t. prec@n) and the ranking obtained by assessing the solutions in an unsupervised way using IREOS. The goodness of fit is measured by computing the Spearman correlation between these two rankings. We have performed this experiment for all datasets.

In addition, for the real datasets we have also performed a **second type** of experiment involving model selection. For each dataset, we produced a diverse collection of candidate solutions by running the well-known algorithms LOF [7], as a representative of *local* outlier detection methods, and kNN Outlier [32], as a representative of *global* outlier detection methods. LOF has the parameter $minPts$ for the neighborhood size used in the algorithm, and kNN Outlier has the parameter k for the number of nearest neighbors considered when computing the kNN distances. We vary both $minPts$ as well as k from 2 to 50 in steps of 3. As the set of candidate solutions, we then take the top-n best scored objects for each value of $minPts$ (from the solutions produced by LOF), as well as the top-n best scored objects for each value of k (from the solutions produced by kNN Outlier). IREOS is then applied to this set of candidate solutions, and the best solution according to the IREOS score is selected. This solution, selected by IREOS, is then compared in terms of prec@n against the best prec@n, the worst prec@n, and the expected (average) prec@n that can be obtained in the set of candidate solutions.

We evaluate IREOS with m_{cl} set to both extremes of the valid interval (see Section 3), in order to represent cases with ($m_{\text{cl}} = n$) and without ($m_{\text{cl}} = 1$) the optional mechanism for modeling clumps. The number of discrete γ values for the practical computation of the index was $n_\gamma = 100$ in all experiments. As previously discussed, the penalty cost for soft margin violations, C, only needs

[4] Note that it is not meaningful to use ROC curves in the realm of top-n outlier detection as we are not evaluating rankings or scorings, but rather binary solutions whose quality is rated considering solely a subset of n objects labeled as outliers.

Table 1: Spearman correlation between IREOS and prec@n for varied soft margin costs (Handl's data collection [12]).

Cost C	$m_{\text{cl}} = 1$	$m_{\text{cl}} = n$
100	0.996 ± 0.009	0.997 ± 0.008
1000	0.998 ± 0.004	0.994 ± 0.02
20000	0.998 ± 0.001	0.995 ± 0.01
800000	0.997 ± 0.003	0.993 ± 0.018

Table 2: Spearman correlation between IREOS and prec@n: synthetic data collections (top) and real datasets (bottom).

Dataset	$m_{\text{cl}} = 1$	$m_{\text{cl}} = n$
Zimek et al. [45]	0.995 ± 0.011	0.996 ± 0.012
Handl et al. [12]	0.998 ± 0.004	0.994 ± 0.02
Annthyroid	0.999	0.999
Diabetes	0.997	0.64
Ionosphere	0.998	0.948
Isolet	1	1
Lymphography	1	1
Multiple Features	0.981	0.99
Optical Digits	1	1
Shuttle	0.52	0.995
Vowel	1	1
WBC	1	0.99
WDBC	1	1

to be big enough. We have experimented with values varying orders of magnitude. The results are very similar and the conclusions do not change. An example — involving a controlled experiment of the first type for one of the collections of synthetic datasets — is shown in Table 1. Notice that the results (correlations) are very similar across values of C from 100 to 800000. For this reason, in the following we show results for $C = 1000$ only.

4.3 Results

The results for the controlled experiments of the 1st type are summarized in Table 2 for all data sets. For the real datasets, which are individual datasets, the entries denote the value of the Spearman correlation between IREOS scores and prec@n for the set of candidate solutions; for the synthetic datasets, which represent collections of datasets, the entries denote the average and the standard deviation of the Spearman correlation between IREOS scores and prec@n. Over a total of 111 datasets, IREOS correlates in almost all cases extremely high with the ground truth (prec@n), for both configurations of m_{cl}. An individual, typical example is illustrated in detail in Figure 3, corresponding to one of the datasets from Handl et al. [12] (2d-10-no2) and IREOS with $m_{\text{cl}} = 1$. The average separability curves that are used to compute IREOS are displayed for the whole collection of candidate solutions (solid lines), which are colored according to prec@n. We can clearly see that the temperature of the color highly correlates with the area under the curve. In fact, we can see by comparing the values of IREOS and prec@n (shown on the right) that there is a perfect correlation in this case. We also display in the figure the curves of values that are expected by chance (dashed) $\pm\sigma$ (dotted). These values have been computed in an exact way as described in Section 3.4 and were used to adjust the index for chance. Based on these statistics, we have also computed and displayed p-values for each candidate solution. As expected, notice that solutions with values of Adjusted IREOS around zero are in fact those composed mostly of randomly selected objects, and are also those with the highest p-values.

Table 3: Monte Carlo simulations (30 runs for varied sample sizes n_{MC} that lead to different percentages of the number of classifiers required for the exact computations).

	$E\{I\}$	Estimated $E\{I\}$	Worst Abs. Difference
1%	0.941	0.940 ± 0.023	0.068
2%	0.941	0.941 ± 0.014	0.044
5%	0.941	0.942 ± 0.007	0.018
10%	0.941	0.941 ± 0.006	0.012
20%	0.941	0.940 ± 0.004	0.01

For the same example dataset, we have also evaluated the trade-off between computational cost and accuracy controlled by the sample size n_{MC} when the adjustment for chance is performed approximately by Monte Carlo simulations. We compared the results between the exact and approximate values for the expected index $E\{I\}$ with sample sizes n_{MC} corresponding to 1%, 2%, 5%, 10%, and 20% of the number of classifiers required for the exact computations. The results are shown in Table 3. Across 30 independent Monte Carlo simulations for each sample size, the *worst case* absolute difference between the exact and approximate values of $E\{I\}$ were 0.068, 0.044, 0.018, 0.012, and 0.010, respectively. As expected, the estimate becomes more accurate as n_{MC} increases, and is, on average, very close to the exact value already for small sample sizes.

The results for the controlled experiments of the 2nd type (model selection for varied candidate LOF and kNN Outlier solutions) are summarized in Table 4, showing prec@n for the worst, the expected (average), and the best cases among the candidates, along with prec@n for the solution selected as best according to IREOS; for each selected solution, the table also indicates which algorithm, LOF or kNN Outlier (or both), produced this solution (in some cases the top solution according to IREOS scores was produced by both algorithms).

By using IREOS with $m_{\text{cl}} = n$ one would select the most accurate solution according to the ground truth in 7 out of the 11 datasets. IREOS with $m_{\text{cl}} = 1$ makes the best choice for 3 out of the 11. In the other cases, the choice is better or competitive when compared with the expected value, often much better and close the quality of the best possible selection according to the ground truth. In all cases the worst solutions are avoided by a large margin, which could not be guaranteed without any validation index. To get a better sense where the selected solutions are located within the distribution of the prec@n values for all candidate solutions obtained by LOF and kNN Outlier, we also show box plots of the distributions for each dataset in Figure 4. The position of the solutions selected by IREOS are indicated by special symbols in the plots.

Comparisons of performances of IREOS for different values of m_{cl} must be taken with a grain of salt, though. The ground truth (i.e., labels based on some sort of semantic) can be seen as a particular perspective of what outliers should be in each dataset. According to this particular perspective, modeling clumps ($m_{\text{cl}} = n$) may be better than not modeling ($m_{\text{cl}} = 1$) or vice versa, but the result could be the reverse if the perspective was different. The Vowel dataset might in fact be an example: not modeling clumps, i.e., $m_{\text{cl}} = 1$, leads to the selection of the best possible solution while

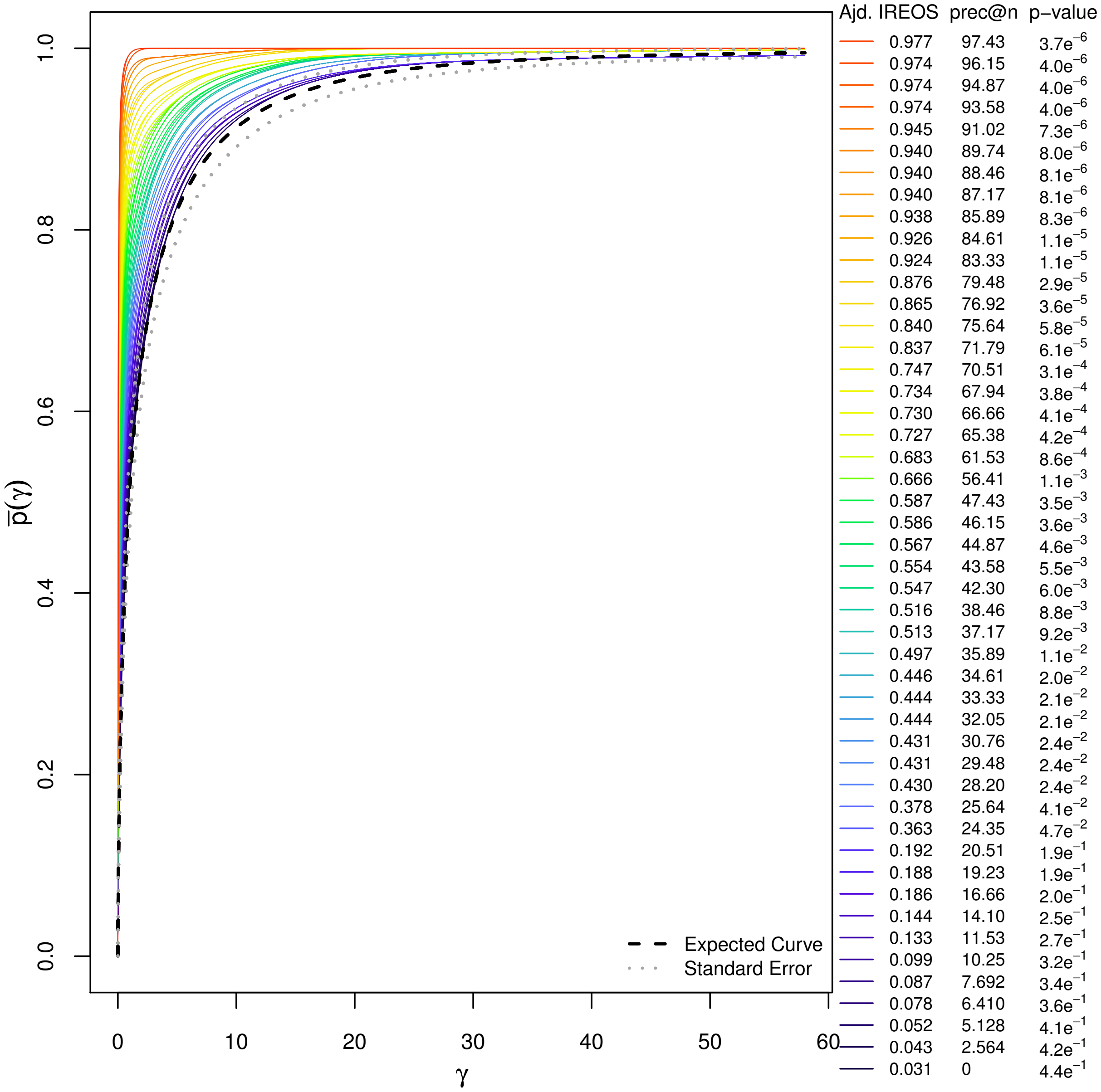


Figure 3: IREOS separability curves for controlled solutions: colors reflect the ground truth (hotter for higher prec@n).

Table 4: Prec@n for LOF and kNN Outlier solutions with varied parameters ($minPts$ and k).

	Dataset	Min	Max	Avg	IREOS ($m_{\text{cl}} = 1$)		IREOS ($m_{\text{cl}} = n$)	
1	Ann_thyroid	0.0538	0.3118	0.241	0.2473	KNN	0.2473	KNN
2	Diabetes	0.3881	0.5597	0.4966	0.5522	KNN	0.5522	KNN
3	Ionosphere	0.6349	0.8492	0.7386	0.8492	KNN	0.8492	KNN
4	Isolet	0	1	0.8353	0.8	LOF	1	KNN
5	Lymphography	0.1667	0.6667	0.4606	0.6111	KNN	0.6667	KNN
6	Multiple Features	0.1	0.5	0.3882	0.4	KNN	0.5	LOF
7	Optical Digits	0	0.8	0.5765	0.2	KNN	0.8	KNN
8	Shuttle	0.0769	0.3846	0.1855	0.3077	LOF|KNN	0.3846	LOF
9	Vowel	0.1	0.9	0.5324	0.9	KNN	0.4	LOF|KNN
10	WBC	0	0.9	0.5265	0.8	KNN	0.8	KNN
11	WDBC	0.1	0.9	0.8324	0.9	LOF|KNN	0.9	LOF|KNN

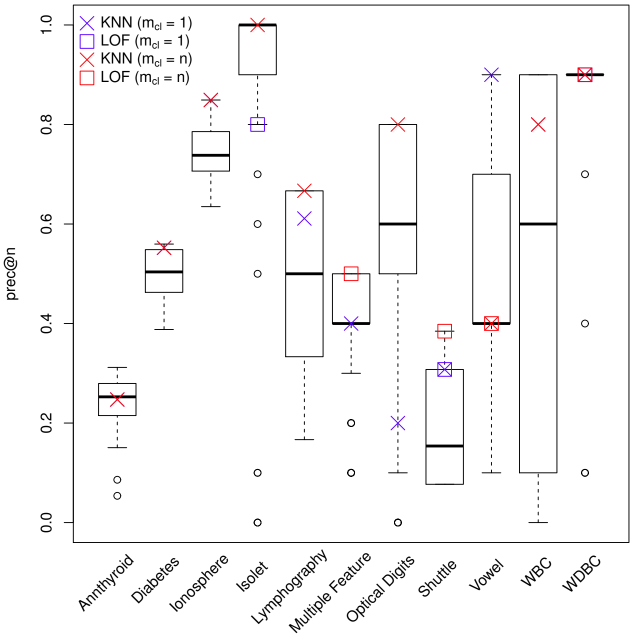


Figure 4: Distribution of the prec@n values for all candidate solutions obtained by LOF and kNN Outlier for the real data sets. The position of the solutions selected by IREOS for different $m_{\text{cl}} = 1$ and $m_{\text{cl}} = n$ are indicated by symbols of different shapes (encoding the method that generated the solution — possibly both) and different colors (encoding the two values of m_{cl}). Some symbols are superposed.

($m_{\text{cl}} = n$) result in a much inferior selection. As discussed in Section 3.3, the choice of modeling clumps is application or user dependent.

5. CONCLUSIONS

We tackled in this paper the long-term open problem [43] of internal and relative evaluation of outlier detection results, that is, the assessment of the quality of results of unsupervised outlier detection methods without refering to external information (such as class labels). In the typical application scenario of outlier detection (other than evaluating new algorithms in the literature), such external information is not available and results need to be assessed by domain experts. IREOS is the first measure to allow such quality assessment of solutions automatically and, as a consequence, to select better solutions (models, parametrizations) for a given problem. We discussed the properties of IREOS, including derived statistics, p-values, and adjustment for chance. Experiments with synthetic and real data, with controlled rankings, and with results of outlier detection algorithms (LOF [7] and kNN Outlier [32]), show the high correlation of IREOS with the true quality of results.

For this first approach to internal evaluation of outlier detection results, we chose the setting as a top-n outlier problem. An interesting research question for future work will be, how to automatically determine n or how to internally evaluate the ranking of outliers independent of a choice of n. Both questions represent long-term open problems too, and we are already working on possible alternatives to tackel these more general problems.

6. ACKNOWLEDGMENTS

This project was partially funded by FAPESP (Brazil — grant #2013/18698-4), CNPq (Brazil — grants #304137/2013-8 and #400772/2014-0), and NSERC (Canada).

7. REFERENCES

[1] F. Angiulli and F. Fassetti. DOLPHIN: an efficient algorithm for mining distance-based outliers in very large datasets. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 3(1):4:1–57, 2009.

[2] F. Angiulli and C. Pizzuti. Outlier mining in large high-dimensional data sets. *IEEE Transactions on Knowledge and Data Engineering*, 17(2):203–215, 2005.

[3] K. Bache and M. Lichman. UCI machine learning repository, 2013.

[4] V. Barnett and T. Lewis. *Outliers in Statistical Data*. John Wiley&Sons, 3rd edition, 1994.

[5] S. D. Bay and M. Schwabacher. Mining distance-based outliers in near linear time with randomization and a simple pruning rule. In *Proceedings of the 9th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD), Washington, DC*, pages 29–38, 2003.

[6] A. Boneau. The effects of violations of assumptions underlying the t test. *Psychological Bulletin*, 57(1):49–64, 1960.

[7] M. M. Breunig, H.-P. Kriegel, R. Ng, and J. Sander. LOF: Identifying density-based local outliers. In *Proceedings of the ACM International Conference on Management of Data (SIGMOD), Dallas, TX*, pages 93–104, 2000.

[8] N. Craswell. Precision at n. In L. Liu and M. T. Özsu, editors, *Encyclopedia of Database Systems*, pages 2127–2128. Springer, 2009.

[9] A. Ghoting, S. Parthasarathy, and M. E. Otey. Fast mining of distance-based outliers in high-dimensional datasets. *Data Mining and Knowledge Discovery*, 16(3):349–364, 2008.

[10] M. Halkidi, Y. Batistakis, and M. Vazirgiannis. On clustering validation techniques. *Journal of Intelligent Information Systems*, 17(2-3):107–145, 2001.

[11] J. Han, M. Kamber, and J. Pei. *Data Mining: Concepts and Techniques*. Morgan Kaufmann, 3rd edition, 2011.

[12] J. Handl, J. Knowles, and D. B. Kell. Computational cluster validation in post-genomic data analysis. *Bioinformatics*, 21(15):3201–3212, 2005.

[13] D. Hawkins. *Identification of Outliers*. Chapman and Hall, 1980.

[14] A. K. Jain and R. C. Dubes. *Algorithms for Clustering Data*. Prentice Hall, Englewood Cliffs, 1988.

[15] W. Jin, A. Tung, and J. Han. Mining top-n local outliers in large databases. In *Proceedings of the 7th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD), San Francisco, CA*, pages 293–298, 2001.

[16] W. Jin, A. K. H. Tung, J. Han, and W. Wang. Ranking outliers using symmetric neighborhood relationship. In *Proceedings of the 10th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD), Singapore*, pages 577–593, 2006.

[17] F. Keller, E. Müller, and K. Böhm. HiCS: high contrast subspaces for density-based outlier ranking. In *Proceedings of the 28th International Conference on Data Engineering (ICDE), Washington, DC*, 2012.

[18] E. M. Knorr and R. T. Ng. Algorithms for mining

distance-based outliers in large datasets. In *Proceedings of the 24th International Conference on Very Large Data Bases (VLDB), New York City, NY*, pages 392–403, 1998.

[19] E. M. Knorr, R. T. Ng, and V. Tucanov. Distance-based outliers: Algorithms and applications. *The VLDB Journal*, 8(3–4):237–253, 2000.

[20] G. Kollios, D. Gunopulos, N. Koudas, and S. Berchthold. Efficient biased sampling for approximate clustering and outlier detection in large datasets. *IEEE Transactions on Knowledge and Data Engineering*, 15(5):1170–1187, 2003.

[21] H.-P. Kriegel, P. Kröger, E. Schubert, and A. Zimek. LoOP: local outlier probabilities. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management (CIKM), Hong Kong, China*, pages 1649–1652, 2009.

[22] H.-P. Kriegel, P. Kröger, E. Schubert, and A. Zimek. Interpreting and unifying outlier scores. In *Proceedings of the 11th SIAM International Conference on Data Mining (SDM), Mesa, AZ*, pages 13–24, 2011.

[23] H.-P. Kriegel, M. Schubert, and A. Zimek. 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, NV*, pages 444–452, 2008.

[24] A. Lazarevic and V. Kumar. Feature bagging for outlier detection. In *Proceedings of the 11th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD), Chicago, IL*, pages 157–166, 2005.

[25] B. Micenková, R. T. Ng, X. H. Dang, and I. Assent. Explaining outliers by subspace separability. In *Proceedings of the 13th IEEE International Conference on Data Mining (ICDM), Dallas, TX*, pages 518–527, 2013.

[26] G. W. Milligan and M. C. Cooper. An examination of procedures for determining the number of clusters in a data set. *Psychometrika*, 50(2):159–179, 1985.

[27] H. V. Nguyen, H. H. Ang, and V. Gopalkrishnan. 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, Japan*, pages 368–383, 2010.

[28] H. V. Nguyen and V. Gopalkrishnan. Efficient pruning schemes for distance-based outlier detection. In *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases (ECML PKDD), Bled, Slovenia*, pages 160–175, 2009.

[29] G. H. Orair, C. Teixeira, Y. Wang, W. Meira Jr., and S. Parthasarathy. Distance-based outlier detection: Consolidation and renewed bearing. *Proceedings of the VLDB Endowment*, 3(2):1469–1480, 2010.

[30] S. Papadimitriou, H. Kitagawa, P. Gibbons, and C. Faloutsos. LOCI: Fast outlier detection using the local correlation integral. In *Proceedings of the 19th International Conference on Data Engineering (ICDE), Bangalore, India*, pages 315–326, 2003.

[31] N. Pham and R. Pagh. A near-linear time approximation algorithm for angle-based outlier detection in high-dimensional data. In *Proceedings of the 18th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD), Beijing, China*, 2012.

[32] S. Ramaswamy, R. Rastogi, and K. Shim. Efficient algorithms for mining outliers from large data sets. In *Proceedings of the ACM International Conference on Management of Data (SIGMOD), Dallas, TX*, pages 427–438, 2000.

[33] S. Ross. *Introduction to Probability Models*. Academic Press, 10 edition, 2009.

[34] E. Schubert, R. Wojdanowski, A. Zimek, and H.-P. Kriegel. On evaluation of outlier rankings and outlier scores. In *Proceedings of the 12th SIAM International Conference on Data Mining (SDM), Anaheim, CA*, pages 1047–1058, 2012.

[35] E. Schubert, A. Zimek, and H.-P. Kriegel. Local outlier detection reconsidered: a generalized view on locality with applications to spatial, video, and network outlier detection. *Data Mining and Knowledge Discovery*, 28(1):190–237, 2014.

[36] B. Schölkopf and A. J. Smola. *Learning with Kernels. Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, 2002.

[37] J. Tang, Z. Chen, A. W.-C. Fu, and D. W. Cheung. 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, Taiwan*, pages 535–548, 2002.

[38] M. Triola. *Elementary Statistics*. Pearson, 10 edition, 2007.

[39] L. Vendramin, R. J. G. B. Campello, and E. R. Hruschka. Relative clustering validity criteria: A comparative overview. *Statistical Analysis and Data Mining*, 3(4):209–235, 2010.

[40] L. Vendramin, P. A. Jaskowiak, and R. J. G. B. Campello. On the combination of relative clustering validity criteria. In *Proceedings of the 25th International Conference on Scientific and Statistical Database Management (SSDBM), Baltimore, MD*, pages 4:1–12, 2013.

[41] K. Zhang, M. Hutter, and H. Jin. 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, Thailand*, pages 813–822, 2009.

[42] J. Zhu and T. Hastie. Kernel logistic regression and the import vector machine. *Journal of Computational and Graphical Statistics*, 2005.

[43] A. Zimek, R. J. G. B. Campello, and J. Sander. Ensembles for unsupervised outlier detection: Challenges and research questions. *ACM SIGKDD Explorations*, 15(1):11–22, 2013.

[44] A. Zimek, R. J. G. B. Campello, and J. Sander. Data perturbation for outlier detection ensembles. In *Proceedings of the 26th International Conference on Scientific and Statistical Database Management (SSDBM), Aalborg, Denmark*, pages 13:1–12, 2014.

[45] A. Zimek, M. Gaudet, R. J. G. B. Campello, and J. Sander. 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, IL*, pages 428–436, 2013.

[46] A. Zimek, E. Schubert, and H.-P. Kriegel. A survey on unsupervised outlier detection in high-dimensional numerical data. *Statistical Analysis and Data Mining*, 5(5):363–387, 2012.