Interpreting and Unifying Outlier Scores

Hans-Peter Kriegel Peer Kröger Erich Schubert Arthur Zimek

Institut für Informatik, Ludwig-Maximilians Universität München
http://www.dbs.ifi.lmu.de
{kriegel,kroegerp,schube,zimek}@dbs.ifi.lmu.de

Abstract

Outlier scores provided by different outlier models differ widely in their meaning, range, and contrast between different outlier models and, hence, are not easily comparable or interpretable. We propose a unification of outlier scores provided by various outlier models and a translation of the arbitrary "outlier factors" to values in the range $[0, 1]$ interpretable as values describing the probability of a data object of being an outlier. As an application, we show that this unification facilitates enhanced ensembles for outlier detection.

1 Introduction

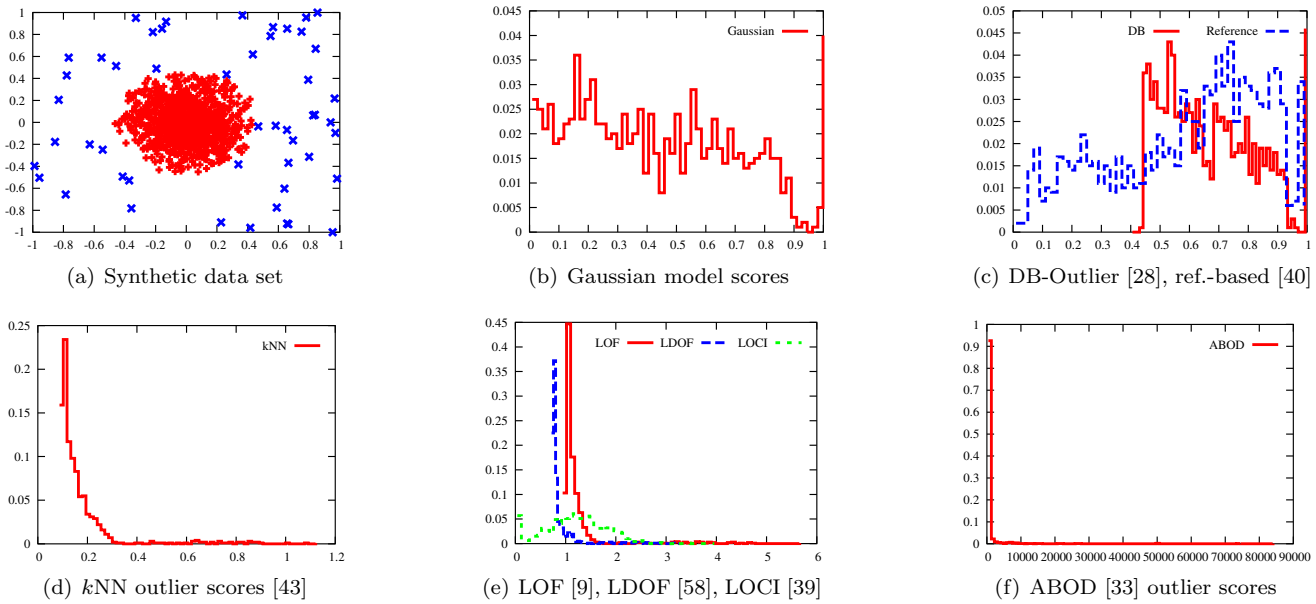
An outlier could be generally defined as being *"an observation (or subset of observations) which appears to be inconsistent with the remainder of that set of data"* [6]. How and when an observation qualifies as being "inconsistent" is, however, not as generally defined and differs from application to application as well as from algorithm to algorithm. From a systematic point of view, approaches can be roughly classified, e.g., as *global* versus *local* outlier models. This distinction refers to the scope of a database being considered when a method decides on the "outlierness" of a given object. Or we can distinguish *supervised* versus *unsupervised* approaches. In this paper, we focus on unsupervised outlier detection. One can also discern between *labeling* and *scoring* outlier detection methods. The former are leading to a binary decision of whether or not a given object is an outlier whereas the latter are rather assigning a degree of outlierness to each object which is closer to the original statistical intuition behind judging on the outlierness of observations. If a statistical test qualifies a certain threshold, a hard decision (or label) for a given observation as being or not being an outlier could be derived. As we will discuss in this study, however, the data mining approaches to outlier detection have moved far from the original statistical intuition during the last decade. Eventually, any outlier score provided by an outlier model should help the user to decide on the actual outlierness. For most approaches, however, the outlier score is not easily interpretable. The scores provided by varying methods differ widely in their scale, their range, and their meaning. While some methods provide probability estimates as outlier scores (e.g. [31]), for many methods, the scaling of occurring values of the outlier score even differs within the same method from data set to data set, i.e., outlier score x in one data set means, we have an outlier, in another data set it is not extraordinary at all. In many cases, even within one data set, the identical outlier score x for two different database objects can denote substantially different degrees of outlierness, depending on different local data distributions. Obviously, this makes the interpretation and comparison of different outlier detection models a very difficult task. Here, we propose scaling methods for a range of different outlier models including a normalization to become independent from the specific data distribution in a given data set as well as a statistically sound motivation for a mapping the scores into the range of $[0, 1]$, readily interpretable as the probability of a given database object for being an outlier. The unified scaling and better comparability of different methods could also facilitate a combined use to get the best of different worlds, e.g. by means of setting up an ensemble of outlier detection methods.

In the remainder, we review existing outlier detection methods in Section 2 and follow their development from the original statistical intuition to modern database applications losing the probabilistic interpretability of the provided scores step-by-step. In Section 3, we introduce types of transformations for selected prototypes of outlier models to a comparable, normalized value readily interpretable as a probability value. As an application scenario, we discuss ensembles for outlier detection based on outlier probability estimates in Section 4. In Section 5, we show how the proposed unification of outlier scores eases their interpretability as well as the comparison and evaluation of different outlier models on a range of different data sets. We will also demonstrate the gained improvement for outlier ensembles. Section 6 concludes the paper.

2 Survey on Outlier Models

2.1 Statistical Intuition of Outlierness We are primarily interested in the nature and meaning of outlier scores provided by unsupervised scoring methods. Scores are closer than labels to discussing probabilities whether or not a single data point is generated by a suspicious mechanism. Statistical approaches to outlier detection are based on presumed distributions of objects relating to statistical processes or generating mechanisms. The classical textbook of Barnett and Lewis [6] discusses numerous tests for different distributions depending on the expected number and location of outliers. A commonly used rule of thumb is that points deviating more than three times the standard deviation from the mean of a normal distribution are considered outliers [27]. Problems of these classical approaches are obviously the required assumption of a specific distribution in order to apply a specific test. There are tests for univariate as well as multivariate data distributions but all tests assume a single, known data distribution to determine an outlier. A classical approach is to fit a Gaussian distribution to a data set, or, equivalently, to use the Mahalanobis distance as a measure of outlierness. Sometimes, the data are assumed to consist of k Gaussian distributions and the means and standard deviations are computed data driven. However, mean and standard deviation are rather sensitive to outliers and the potential outliers are still considered for the computation step. There are proposals of more robust estimations of these parameters, e.g. [19,45], but problems still remain as the proposed concepts require a distance function to assess the closeness of points before the adapted distance measure is available. Related discussions consider the robust computation of PCA [14, 30]. Variants of the statistical modeling of outliers have been proposed in computer graphics ("depth based") [25, 46, 51] as well as databases ("deviation based") [4, 48].

2.2 Prototypes and Variants The distance-based notion of outliers (DB-outlier) may be the origin of developing outlier detection algorithms in the context of databases but is in turn still closely related to the statistical intuition of outliers and unifies distribution-based approaches under certain assumptions [27, 28]. The model relies on two input parameters, D and p. In a database $\mathcal{D}$, an object $x \in \mathcal{D}$ is an outlier if at least a fraction p of all data objects in $\mathcal{D}$ has a distance above D from x. By definition, the DB-outlier model is a labeling approach. Adjusting the threshold can reflect the statistical $3 \cdot \sigma$-rule for normal distributions and similar rules for other distributions. Omitting the threshold p and reporting the fraction p_x of data objects $o \in \mathcal{D}$ where $\text{dist}(x, o) > D$ results in an outlier score for x in the range $[0, 1]$. The model is based on statistical reasoning but simplifies the approach to outlier detection considerably, motivated by the need for scalable methods for huge data sets. This, in turn, inspired many new outlier detection methods within the database and data mining community over the last decade. For most of these approaches, however, the connection to a statistical reasoning is not obvious any more. Scoring variants of the DB-outlier notion are proposed in [3, 8, 29, 40, 43] basically using the distances to the k-nearest neighbors (kNNs) or aggregates thereof. Yet none of these variants can as easily translate to the original statistical definition of an outlier as the original DB-outlier notion. Scorings reflect distances and can grow arbitrarily with no upper bound. The focus of these papers is not on reflecting the meaning of the presented score but on providing more efficient algorithms to compute the score, often combined with a top-n-approach, interested in ranking outlier scores and reporting the top-n outliers rather than interpreting given values. Density-based approaches introduce the notion of local outliers by considering ratios between the local density around an object and the local density around its neighboring objects. The basic local outlier factor (LOF) assigned to each object of the database $\mathcal{D}$ denotes a degree of outlierness [9]. The LOF compares the density of each object $o \in \mathcal{D}$ with the density of its kNNs. A LOF value of approximately 1 indicates that the corresponding object is located within a region of homogeneous density. The higher the LOF value of an object o is, the higher is the difference between the density around o and the density around its kNNs, i.e., the more distinctly is o considered an outlier. Several extensions of the basic LOF model and algorithms for efficient (top-n) computation have been proposed [22, 23, 49, 50] that basically use different notions of neighborhoods. The Local Outlier Integral (LOCI) bases on the concept of a multi-granularity deviation factor and ε-neighborhoods instead of kNNs [39]. The resolution-based outlier factor (ROF) [16] is a mix of the local and the global outlier paradigm based on the idea of a change of resolution, i.e., the number of objects representing the neighborhood. As opposed to the methods reflecting kNN distances, in truly local methods the resulting outlier score is adaptive to fluctuations in the local density and, thus, comparable over a data set with varying densities. The central contribution of LOF and related methods is to provide a normalization of outlier scores for a given data set. Recently, this notion of normalized local outlierness has been also merged with the distance-based notion of outliers, resulting in the local distance-based outlier detection approach LDOF [58]. Adaptations to high dimensional data are, e.g., [12, 32]. The outlier



(a) Synthetic data set (b) Gaussian model scores (c) DB-Outlier [28], ref.-based [40]

(d) kNN outlier scores [43] (e) LOF [9], LDOF [58], LOCI [39] (f) ABOD [33] outlier scores

Figure 1: Deteriorating interpretability of outlier scores for recent methods. For the data depicted in (a), the histograms (b)–(f) show bins of similar outlier scores of different algorithms along the x-axis and the relative number of objects in each bin along the y-axis.

score ABOD [33] is not primarily based on conventional distance measures. It assesses the variance in angles between an outlier candidate and all other pairs of points. Its relationship to other outlier models is unclear. Since the variance of angles is lower for outliers than for inliers, ABOD is an example where, as opposed to many other methods, the higher score reflects a lower outlierness.

2.3 Rise and Decline of Outlier Models While outlier detection algorithms became ever more sophisticated and efficient during the last decade of research e.g. in terms of providing the top-n outliers ever faster, less diligence has been invested in the meaning and comparability of the scores provided by the different methods. In Figure 1, we observe some interesting trends for several state-of-the-art methods of outlier detection. For a simple 2-d data set containing a large bivariate normal distribution (inliers) and a small but wider uniform distribution responsible for outliers (Fig. 1(a)), a simple statistical method is fitting a Gaussian model and deriving linearly inverted probability density values, sketched in Fig. 1(b) as a histogram. Many points show values between 0 and 0.9 corresponding to the distance from the mean. The relative gap between 0.9 and 1 reflects the spatial gap between the points truly generated by a Gaussian model and the outer rim of sparsely distributed points from the uniform distribution which attract an outlier probability of 1. While DB-Outlier scores [28] and reference-based outlier scores [40] are both normalized scorings in the range $[0, 1]$ (the latter assessing approximately k-NN distances and normalizing the obtained values for a data set), their distribution is rather different (Fig. 1(c)). A gap between an outlier score 0.9 and 1 is still visible for DB-Outlier but its size and distinctiveness is diminished considerably. Also, there are no true inlier probabilities for DB-Outlier. For the k-NN distances based outlier factor, the scale is not normalized anymore and in its scaling it is actually highly sensible to the data distribution and dimensionality at hand since it merely reflects occurring distances. There is no gap observable anymore, but scores for outliers are more and more increasing with no hint whatsoever which score is large enough to characterize an outlier. For LOF [9], LDOF [58], and LOCI [39] (Fig. 1(e)), the scale of outlier scores is also dependent on the given data distribution though the scores are locally adaptive and the authors proposed a typical inlier value (1, 0.5, and 0, for LOF, LDOF, and LOCI, respectively). While outliers generally attract higher scores, and the typical inlier values are abundantly occurring, there is once more no gap contrasting between still-inlier and already-outlier values. Finally, the range for ABOD [33] is 0 (corresponding to outliers) up to over $80,000$. Again, although the ranking is viable, no contrast between inlier scores and outlier

scores is visible and hence the user does not have any clue which score was small enough to label an outlier.

A reason that all these aspects have not gained much attention so far might be that outlier detection methods usually have been evaluated in a ranking setting ("give me the top-n outliers"). This, however, is an unrealistic or at least unsatisfactory scenario in many applications since it requires exact knowledge on the number of outliers in a data set in order to retrieve all and only outliers, especially if there is no distinguished gap between outlier scores and inlier scores.

2.4 Summary Outlier detection methods are based on quite different assumptions, intuitions, and models. They naturally also differ substantially in the scaling, range, and meaning of values, and they are differently well suited for different data sets. Their results, however, are in many cases very difficult to compare. Opposite decisions on the outlierness of single objects may be equally meaningful since there is no generally valid definition of what constitutes an outlier in the first place. For different applications, a different selection of outlier detection approaches may be meaningful. The development of methods within the database and data mining community, triggered by Knorr and Ng, was focussed on efficiency rather than on interpretability. As a consequence, these methods usually are not able to translate their scoring result into a label. There is usually no clear decision boundary which score is generally sufficient to designate any data point as an outlier (as opposed to the $3 \cdot \sigma$-rule-of-thumb in a statistical context). Here, we try to reestablish a genuine interpretability of outlier "scores" for some methods exemplary for the different families of outlier models.

3 Unifying Outlier Scores

The fundamental motivation for a re-scaling of outlier scores is to establish sufficient contrast between outlier scores and inlier scores, inspired by Hawkins [20]: *"a sample containing outliers would show up such characteristics as large gaps between 'outlying' and 'inlying' observations and the deviation between outliers and the group of inliers, as measured on some suitably standardized scale"*. As shown above, existing outlier scores often lack this contrast between outliers and inliers, i.e., there is usually no clear gap between the scores of outliers and inliers. Beside a poor semantic of the provided scores it becomes hard for the user to differentiate between objects from both classes. The above typification of outlier models and the respective meaning of resulting outlier scores (cf. Section 2) facilitates types of functions to transform an outlier score to a comparable, normalized value or even to a probability value. For example, the scorings of LOF and LDOF share a similar meaning and, thus, can be transformed by similar functions. The DB-Outlier labeling can be translated into a scoring approach directly resulting in a score in the range $[0, 1]$. Scores reaching infinity (either as maximal outlier score or as maximal inlier score) require more complex considerations for transformation. For example, the normalization procedure applied to k-NN score approximations in the reference points approach [40] does not result in good contrast of outlier scores vs. inlier scores (see Fig. 1(c)). Instead, it is just a scaling onto the range $[0, 1]$. For scores with extremely low numeric contrast we therefore consider functions to stretch interesting ranges while shrinking irrelevant regions.

3.1 A General Procedure for Normalizing Outlier Scores Based on the above considerations, we will first derive a general framework for transforming outlier scores to a comparable, normalized value or even to a probability value. After that, we sketch how this framework can be implemented for existing outlier scores. The anticipated minimum requirements of the resulting score are *regularity* and *normality*. An outlier score S is called *regular* if $S(o) \geq 0$ for any object o, $S(o) \approx 0$ if o is an inlier, and $S(o) \gg 0$ if o is an outlier. An outlier score S is called *normal* if S is regular and the values are restricted by $S(o) \in [0, 1]$. Many outlier scores are already regular and/or normal. For example, probability based outlier scores obviously are already normal. Others will require slight adjustment or larger transformation for regularity. A transformation that yields a regular (normal) score is called *regular (normal)*. An important property of transformations is that they should not change the ordering obtained by the original score. Formally we require that for any $o_1, o_2 : S(o_1) \leq S(o_2) \Rightarrow T_S(o_1) \leq T_S(o_2)$. Alternatively, for inverted scores, it is also admissible to have for any $o_1, o_2 : S(o_1) \leq S(o_2) \Rightarrow T_S(o_1) \geq T_S(o_2)$. A transformation that either fulfills the first or the second equation is called *ranking-stable*. Note that this does permit $S(o_1) < S(o_2) \wedge T_S(o_1) = T_S(o_2)$. This is intentional since we do accept some loss in information for inliers (where the ranking information is not interesting) if that helps increasing the contrast for outliers.

We consider formally two steps in the process of unification of scores, where either step may be optional (depending on the score S): (i) A *regularization* Reg basically transforms a score S onto the interval $[0, \infty)$ such that $Reg_S(o) \approx 0$ for inliers and $Reg_S(o) \gg 0$ for outliers. We propose different regularization procedures for different types of scores in Section 3.2 depending on their original domains. (ii) A *normalization* basically transforms a score into the interval $[0, 1]$. We propose

different normalization procedures in Section 3.3. Both, regularizations and normalizations, can be used to enhance the contrast between inliers and outliers. Some normalizations will require a regularized score to work with, while others do not have this requirement but may still perform better if the data was regularized beforehand. Hence, it can even be beneficial to apply a normalization to an already normal score, if the normalization stretches interesting regions and shrinks irrelevant intervals.

3.2 Regularization Different scores need different transformations to become regular. In the following, we outline regularization procedures for different classes of outlier scores

3.2.1 Baseline Regularization The local outlier scores LOF, LDOF, and their variants are not yet regular, since the expected value for non-outliers is not 0. In case of LOF and its variants, the expected inlier value is $base_{\text{LOF}} = 1$. For LDOF the expected inlier score is $base_{\text{LDOF}} = \frac{1}{2}$. The expected outlier score is $\gg base$ in both cases. These scores can however be regularized with a very simple transformation. Let $base_S$ be the baseline (expected inlier value) of the outlier score S. The idea for a regular transformation is to take the difference of the observed value $S(o)$ of an object o and the baseline value $base_S$. This transforms any interval $[base, \infty)$ to the interval $[0, \infty)$. Since the considered scores may also produce scores that are smaller than $base_S$ indicating also inliers, we need some adjustment not to get negative scores after transformation:

$$Reg_S^{base_S}(o) := \max\{0, S(o) - base_S\}.$$

This regularization is ranking-stable:

$$\begin{aligned} &S(o_1) \leq S(o_2) \Leftrightarrow S(o_1) - base_S \leq S(o_2) - base_S \\ &\quad \Rightarrow \quad \max\{0, S(o_1) - base_S\} \leq \max\{0, S(o_2) - base_S\} \\ &\quad \Leftrightarrow \quad Reg_S^{base_S}(o_1) \leq Reg_S^{base_S}(o_2). \end{aligned}$$

In other words, if o_1 has a lower score than o_2 which means o_1 is less an outlier than o_2 for LOF, its variants and LDOF, then it cannot have a higher score after a baseline regularization. Note that for $S(o_1) < S(o_2) < base_S$ we lose information, since $Reg_S^{base_S}(o_1) = 0 = Reg_S^{base_S}(o_2)$, but this is intentional. It is also easy to see that this regularization does not enhance the contrast between inlier and outlier scores.

3.2.2 Linear Inversion In some outlier models, high scores are inliers. For example, when using the density function of a Gaussian model, high density identifies inliers, while a density close to 0 indicates outliers. To regularize such models, we need to invert them. For that purpose, we simply take the difference between the observed score $S(o)$ and the maximum possible (or observed) score $S_{\max}$.

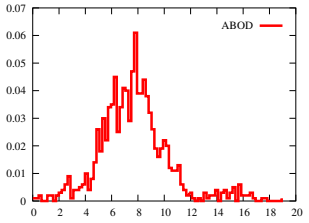


Figure 2: ABOD scores, after logarithmic inversion

$$Reg_S^{\text{lininv}}(o) := S_{\max} - S(o).$$

Since $S_{\max} \geq S(o)$, this transformation is regular. Ranking-stability for inverted scores can also easily be shown: $S(o_1) \leq S(o_2) \Leftrightarrow -S(o_1) \geq -S(o_2) \Leftrightarrow Reg_S^{\text{lininv}}(o_1) \geq Reg_S^{\text{lininv}}(o_2)$. It is also easy to see that this regularization does not enhance the contrast between inlier and outlier scoring.

3.2.3 Logarithmic Inversion A simple linear inversion as mentioned above is not appropriate for algorithms with very low contrast such as ABOD. A more useful regularization for ABOD that also addresses the enhancement of contrast between inliers and outliers uses the logarithm function:

$$Reg_S^{\text{loginv}}(o) := -\log(S(o)/S_{\max}).$$

Note that this regularization is only defined if $S(o) > 0$ for all objects o and $S_{\max}$ finite. ABOD is such a score that produces only positive values greater than zero; there is no upper bound so we have to choose $S_{\max}$ from the occurring scores. Since the logarithm is a monotone function, the logarithmic inversion is ranking-stable. As it can be seen in Fig. 2, this regularization can significantly increase the contrast (compare to Fig. 1(f)).

3.3 Normalization The simplest way of bringing outlier scores onto a normalized scale is to apply a linear transformation such that the minimum (maximum) occurring score is mapped to 0 (1). A simple linear normalization can be obtained by

$$Norm_S^{\text{linear}}(o) := \frac{S(o) - S_{\min}}{S_{\max} - S_{\min}}.$$

Note that, if S is a regular or regularized score, $S_{\min} = 0$ can be used to simplify the formula. Obviously,

this normalization does not add any contrast to the distribution of scores. Thus, here we study possibilities to enhance the contrast between inliers and outliers based on some exemplary statistical scaling methods and, hence, motivating a probabilistic interpretability of the unified scores.

3.3.1 Statistical Scaling of Outlier Scores Outlier scores usually are not equally distributed in their value range but follow a much more complex distribution which is hard to grasp analytically, and many assumptions have to be made to get a reliable result. In particular, when one has to assume that the data set is composed out of multiple clusters and noise distributions, a purely analytical analysis is infeasible. Even a single multidimensional normal distribution already poses the challenge, that there is no known closed form solution for its cumulative distribution function. The nature of many outlier scores depending on kNNs and even "kNNs of kNNs" like in the LOF model makes even data based on strong assumptions intractable. Instead it becomes much easier to analyze the actual score distribution on the data set without assuming anything on the distribution of the data. Note that the linear scaling proposed above can be interpreted as assuming a uniform distribution of the outlier scores. However, in order to avoid overfitting it seems advisable to choose a primitive distribution function with limited degrees of freedom that is fit to the observed outlier scores. Here we propose several normalization functions that are based on such distribution functions. It should be again pointed out that we do not draw any assumptions on the distribution of the data but on the distribution of the derived outlier scores. Let us note that any distribution function can be used for this purpose analogously, depending on how well it is fitting to the distribution of the derived outlier scores. The intuition of this approach is the following: we do not have a direct way to interpret the score, we instead evaluate how "unusual" the score of a point is, using the algorithms output score as a one-dimensional feature projection of the data.

We chose Gaussian and Gamma distributions as examples. However, the same method can be applied to any other distribution function. For ratio-based algorithms such as LOF and LDOF, ratio distributions such as the Cauchy-distribution and F-distribution are good candidates. ABOD scores, after the logarithmic inversion, on the other hand appear to be normally distributed. For example, Figure 2 shows the regularized but not yet normalized scores of ABOD on the data set from Figure 1. The optimal choice of distribution depends on the algorithm, and maybe also on the data set. As such we will not give optimal distributions for any algorithm here, but we will show in the experiments that the choice of an arbitrary distribution already offers significantly improved performance.

3.3.2 Gaussian Scaling According to the central limit theorem, the most general distribution for values derived from a large number of similarly distributed values is the normal distribution. Having just two degrees of freedom (the mean μ and the standard deviation σ) it is not susceptible to overfitting. Using the known estimator functions $\hat{\mu} = E(S)$ and $\hat{\sigma}^2 = E(S^2) - E(S)^2$, the parameters can be fit to the data efficiently in a single pass. A more sophisticated fit can be achieved by using the Levenberg-Marquardt method. Given the mean μ_S and the standard deviation σ_S of the set of derived values using the outlier score S, we can use its cumulative distribution function and the "Gaussian error function" *erf*() to transform the outlier score into a probability value:

$$Norm_S^{\text{gauss}}(o) := \max\left\{0, erf\left(\frac{S(o)-\mu_S}{\sigma_S \cdot \sqrt{2}}\right)\right\}.$$

Note that the same value can be obtained by linear regularization of the cumulative distribution function:

$$cdf_S^{\text{gauss}}(o) := \frac{1}{2}\left(1 + erf\left(\frac{S(o)-\mu_S}{\sigma_S \cdot \sqrt{2}}\right)\right).$$

Using the mean $\mu_{\text{cdf}} = cdf_S^{\text{gauss}}(\mu_S) = \frac{1}{2}$ and $\max_{\text{cdf}} = 1$ it follows that

$$\begin{aligned}&\max\left\{0, \frac{cdf_S^{\text{gauss}}(o) - \mu_{\text{cdf}}}{\max_{\text{cdf}} - \mu_{\text{cdf}}}\right\}\\ =\ &\max\{0, 2 \cdot cdf_S^{\text{gauss}}(o) - 1\} = Norm_S^{\text{gauss}}(o).\end{aligned}$$

Since the Error Function *erf*() is monotone it is ranking-stable.

If the algorithm the normalization is applied to has known characteristics, we can also use other estimators for μ or σ. Comparable to the baseline approach above, let $\mu_S = base_S$ and $\sigma_S^2 = \frac{1}{|S|}\sum(S(o) - base_S)^2$. We call such an approach *customized* Gaussian scaling. Note that the use of such a different base essentially assumes that the scores should be distributed with the given parameters. For LOF, LDOF, or LOCI, assuming a Gaussian distribution around the corresponding values of *base* does show nearly the same performance as a not customized Gaussian scaling. These results suggest that the values of *base* given by the authors of LOF, LDOF, or LOCI are actually rather appropriate.

3.3.3 Gamma Scaling The assumption of a Gaussian distribution works quite well in particular in

high dimensionality. However, the histograms of low-dimensional kNN and LOF scores rather resemble a Gamma distribution. Notice that the χ^2 distribution and exponential distributions are specializations of the Gamma distribution: $\chi^2(\nu) \sim \Gamma(k = \nu/2, \theta = 2)$ and $Exp(\lambda) \sim \Gamma(k = 1, \theta = 1/\lambda)$.

We can estimate the parameters of the Gamma distribution $\Gamma(k, \theta)$ with shape k and Gamma mean θ using the estimators $\hat{\mu}$ and $\hat{\sigma}$ as above and then using the moment estimators $\hat{k} := \frac{\hat{\mu}^2}{\hat{\sigma}^2}$ and $\hat{\theta} := \frac{\hat{\sigma}}{\hat{\mu}^2}$. Note that we need $\mu \neq 0$ for this to be well-defined which is a reasonable assumption in most cases. The cumulative density function is given by

$$cdf_S^{\text{gamma}}(o) := \frac{\gamma(k, S(o)/\theta)}{\Gamma(k)} = P(k, S(o)/\theta),$$

where P is the regularized Gamma function.

Similar to the Gaussian normalization, we use

$$Norm_S^{\text{gamma}}(o) := \max\left\{0, \frac{cdf_S^{\text{gamma}}(o) - \mu_{\text{cdf}}}{1 - \mu_{\text{cdf}}}\right\},$$

where $\mu_{\text{cdf}} = cdf_S^{\text{gamma}}(\mu_S)$.

Having transformed the scores of quite different outlier models to a range of $[0, 1]$ does not yield a unified meaning of these scores. If interpreted as probability of being an outlier, each model still yields this probability value under certain assumptions that constitute the specific outlier model. Nevertheless, the level of unification yielded by the above transformations improves the comparability of the decisions of different approaches.

4 Application Scenario: Outlier Ensembles

4.1 Outlier Ensemble Approaches Building ensembles of outlier detection methods has been proposed occasionally [17, 34, 38], i.e., building different instances of outlier detection algorithms (called "detectors") for example by different parametrization, using different subspaces, or actually using different algorithms and combining the outlier scores or ranks provided by the different detectors somehow. The first approach [34] proposes to combine detectors used on different subspaces of a data set. Two combination methods have been discussed, namely breadth-first and cumulative sum. The breadth-first method is purely based on the ranks of the data objects provided by the different detectors. A combined list sorted according to their ranks in different detectors is created by merging the rank lists provided by all detectors. The cumulative sum approach provides the sum of all outlier scores provided by detectors for a data object as the object's new outlier score and re-ranks the data objects according to this new score. Although the paper pretends to provide a framework for the combination of actually different algorithms, this is questionable for the cumulative sum approach. Indeed, the paper presents results based on combinations of different instances of LOF only. While the breadth-first method does not actually compare different outlier scores, the cumulative sum would not be suitable to combine scores of largely different scales (let alone inverted scores). An improvement has been proposed by [17], utilizing calibration approaches (sigmoid functions or mixture modeling) to fit outlier scores provided by different detectors into probability values. The combination of probability estimates instead of mere ranks is demonstrated in [17] to slightly improve on the methods of [34]. Notably, although their method should theoretically be applicable to combine different outlier detection algorithms, their experiments demonstrate the combination of different values for k of the kNN distance as an outlier score only. Note that the sigmoid learning and mixture model fitting approaches proposed for calibration in [17] are based on the generalized EM algorithm and are rather unstable. In addition, they favor extreme values (0 and 1), which is not favorable for combination, but degenerates to counting and boolean combinators. The recent approach proposed in [38] eventually addresses the combination of actually different methods more explicitly. The scores provided by a specific algorithm are centered around their mean and scaled by their standard deviation, hence not interpretable as probability estimates and possibly still different in scale for different methods. As combination method, [38] propose weighting schemes trained in a semi-supervised manner similar to the training procedure described in [1]. But for an unsupervised scenario, an unweighted combination should still be admissible. To induce diversity among different detectors, [38] follow the feature bagging approach of [34].

4.2 Combining Different Outlier Methods Our unification approach to outlier scores combines the advantages of the previous methods while avoiding their pitfalls. We transform outlier scores of arbitrary methods into probability estimates (as we will demonstrate below, improving on the approaches proposed in [17]) and hence allow for a more stable and also more meaningful combination of arbitrary outlier detection methods in a completely unsupervised manner (improving on [34, 38]). As a simple demonstration of the applicability of our unification of outlier scores and the gained improvement in outlier ranking quality, we propose to construct an ensemble out of instances of arbitrary, different outlier detection algorithms and use the probability estimate produced by our method out of the outlier

score directly as a weight in the combination. As a result, we predict the probability estimate averaged over all ensemble members, i.e., for a data object o, a selection OD of instances of outlier detection algorithms and for each element $OD_i \in OD$ some function $prob_{OD_i}$, mapping an outlier score S_i provided by OD_i to a probability value using a normalization suitable for OD_i, as discussed in Section 3, we obtain as ensemble outlier score:

$$P(o) = \frac{1}{|OD|} \sum_{OD_i \in OD} prob_{OD_i}(S_i(o))$$

This is a rather simple, baseline scheme. Obviously, there are many other combinations possible that make more use of the probabilistic interpretation.

5 Evaluation

A bigger advance than "yet another outlier model" is a way to the unification of the variety of existing outlier models since this opens up possibilities of comparison or even combination of different methods. In Section 3, we presented some first steps towards this goal. Here, we are however not interested in evaluating the different methods for outlier detection or ensemble construction (which would be far beyond the scope of this study) but in evaluating the increase in usability of different methods that can be gained by applying our unification approach. The evaluation of this unification irrespective of the underlying methods is not straightforward. Displaying ROC curves or ROC AUC values would not make sense since the transformations generally do not affect the ranking. We will discuss how to evaluate the unification of scores (motivated as probability estimations) meaningfully (Section 5.1), and subsequently provide a broad range of corresponding experiments (Section 5.2). Finally, we will also evaluate the application scenario sketched in Section 4 incorporating the unified scores into simple ensemble approaches to outlier detection (Section 5.3), as a proof of principle. All proposed methods and the competitors have been implemented in the framework ELKI [2].

5.1 Evaluating Probability Estimates The concept of calibration has been used to assess the reliability of probability estimates or confidence values in classification. An optimally calibrated prediction algorithm is expected to be correct in a portion x of all predictions that are subject to a confidence value of x [10]. Obtaining more or less well calibrated probability estimates has gained a lot of attention in the context of classification (e.g. [41, 42, 56, 57]). Here, in the unsupervised context of outlier detection, we propose the first approach to reconvert plain outlier scores roughly into probability estimates as a motivation for the unification and interpretation of scores. In an unsupervised setting, however, previous approaches cannot be pursued since one cannot learn or fit a mapping but has to assume certain properties for a certain outlier algorithm. We can however assess typical distributions of outlier scores (as surveyed above) and propose suitable scaling methods to convert outlier scores into probability estimates though not optimized and thus not perfectly calibrated for each data set. In general, the pure concept of calibration is questionable anyway [13, 37, 55]. Here, however, our goal is not perfect calibration but improved interpretability of scores by an improved contrast between outlier scores (i.e., high outlier probability) and inlier scores (i.e., low outlier probability). This relates to the concept of refinement or sharpness, i.e., the forecast probabilities should be concentrated near 1 and near 0 (still being correlated with the actual occurrence of the forecast event) [13, 47]. It should be noted, though, that there may be cases of intermediate outlier probabilities. If these are indeed borderline cases requiring closer inspection, to assign an intermediate probability estimate to these data objects may be desirable. In the classification area, abstaining classifiers are adequate in such cases (see e.g. [41]). In the outlier detection area, this problem has not found much attention in recent methods but is the genuine merit of the original statistical approaches [6, 20].

An important difference between the probability estimation of outliers and of classes is the inherently imbalanced nature of the outlier detection problem. Since the data are largely dominated by the class of inliers and only a minimal number of data objects are truly outliers, assessing the root mean squared error in reliability as deviation of the probability estimates vs. the observed frequency (as e.g. in [37], for different classical evaluation measures see [36]) is not directly applicable to the scenario of outlier probability estimates. Any outlier detection procedure always estimating a zero (or a very small) outlier probability would already be almost perfectly calibrated. Optimal calibration can be achieved by merely estimating the proportion of outliers in a data set instead of assessing the outlierness of single data objects. In the supervised context of classification, much effort has been spent on methods of sanitizing imbalanced class prior probabilities for training as well as for evaluation of classifiers [7, 24, 44, 53]. Closely related to the problem of imbalanced classes are cost sensitive learning [11, 54] and boosting [26]. Applying a cost-model to a prediction task means that errors are differently penalized for different classes during the learning procedure. Different costs for different types of error are a quite realistic scenario. What truly counts is not

the optimally calibrated probability estimation but the minimized costs of a decision (in the decision-theoretic sense of [52], see also [35]), where the decision, of course, is based on the estimated probability. Instead of costs, the expected utility could also be modeled, or both, utility and costs. While in supervised scenarios classifiers can be optimized w.r.t. a certain cost model [15, 56], in the unsupervised scenario of outlier detection, the assumed cost-model cannot be used to fit or train the algorithm but only to evaluate its results. It should be noted, though, that while calibration and purely calibration related scores in itself are not a sufficient evaluation measure, a useful cost-model-based evaluation of decisions will also encourage calibration [55]. In the context of imbalanced classes, it is customary to either sample the small class up, to sample the large class down, or to alter the relative costs according to the class sizes. In [21], the latter has been shown to be generally more effective than the alternatives. Hence we adopt this procedure here, though, since we tackle outlier detection as an unsupervised task, not for adapting a training procedure to differently weighted misclassification costs but merely to evaluate the impact of a probabilistic scaling and regularization of outlier scores. Aside from a quantitative improvement, the major motivation for such a probabilistic scaling is to revert the more and more deteriorated interpretability of modern outlier detection methods into a statistically interpretable context.

In summary, we transfer the experiences with (i) probability estimates, (ii) imbalanced classification problems, and (iii) cost-sensitive learning reported in the context of supervised learning to the context of unsupervised outlier detection. Hence, we assess the impact of our transformation methods for outlier scores w.r.t. the reduction of error-costs, taking into account the different cardinality of the class of inliers I and the class of outliers O. At the same time, to simultaneously account for calibration, we do not assess binary decisions but multiply assigned probability estimates with the corresponding costs. Since the costs for the correct decisions are always 0, only errors account for the reported values. The corresponding accuracy values would be symmetric since the assigned probability estimates would just be the complementary probabilities of the ones accounting for error costs. Formally, the reported costs are for each dichotomous problem consisting of classes I and O:

$$\frac{1}{2}\sum_{x\in I} P(O|x)\cdot\frac{1}{|I|}+\frac{1}{2}\sum_{x\in O} P(I|x)\cdot\frac{1}{|O|}$$

based on probability estimates $P(C|x)$ for an object x to belong to class C as provided by the (unified) outlier score.

Table 1: Normal data, uniform noise

	Lin	**LinM**	**Gau**	**GauC**	Γ	Rank	SigM	MixM
Gaussian	31.6	**18.8**	25.1	36.6	26.6	25.1	28.8	51.5
kNN [43]	20.4	19.4	**3.3**	25.6	4.5	25.0	9.2	15.1
agg. kNN [3]	21.1	20.2	**2.9**	24.2	4.4	25.0	7.9	16.1
LDOF [58]	22.6	21.9	**4.2**	27.8	4.5	25.1	9.0	21.1
LOCI [39]	29.1	25.8	**15.0**	33.2	16.0	33.2	23.5	19.9
LOF [9]	21.6	21.4	**2.3**	6.0	2.9	25.0	5.6	11.8
Refer. [40]	33.8	**14.4**	19.5	39.4	21.0	25.8	26.9	48.4
DB-Out. [28]	10.1	10.1	**2.4**	10.8	4.8	25.2	3.4	18.5
ABOD [33]	24.0	n/a	**7.4**	n/a	8.5	25.0	n/a	n/a

Table 2: Uniform data, uniform noise

	Lin	**LinM**	**Gau**	**GauC**	Γ	Rank	SigM	MixM
Gaussian	37.2	**23.8**	29.6	40.7	30.8	26.5	32.1	52.1
kNN [43]	26.5	24.7	**8.9**	37.5	9.4	26.3	14.8	28.6
agg. kNN [3]	26.9	25.2	**8.1**	36.1	8.7	26.1	14.5	27.5
LDOF [58]	28.9	27.7	**8.9**	34.5	9.2	26.0	12.8	27.8
LOCI [39]	37.1	29.9	**24.0**	41.2	25.1	33.4	33.1	37.2
LOF [9]	28.0	27.7	**8.2**	10.6	8.4	26.4	10.7	41.9
Refer. [40]	36.4	**22.6**	24.4	41.1	25.3	30.4	29.2	44.0
DB-Out. [28]	31.2	**19.3**	17.5	40.6	18.4	26.6	25.7	50.0
ABOD [33]	34.5	n/a	19.0	n/a	**17.2**	27.8	n/a	n/a

5.2 Reduction of Error-Costs Artificial data sets offer well-defined outliers, and are thus much easier to evaluate. At the same time, there is a risk of overfitting the algorithms (or in our setup, the normalizations) to the idealized setting of artificial data. Real world data on the other hand are hard to evaluate since it is often impossible to decide whether or not a point is an outlier. Real world data sets can contain points with the same attribute values where one point is marked as outlier and the other is not. This can be due to measurement errors or incomplete data or even misclassification. As such, it is impossible to achieve "perfect" results on real data. It is best to think of real data as containing outliers, some of which are known and marked as interesting. An algorithm is working well as long as it is reporting the interesting outliers and not too many of the others.

5.2.1 Artificial Data The data sets we generated contained uniformly or normally clustered data, along with uniform background noise. We generated data sets of 1, 2, 3, 5, 10 and 100 dimensions, and averaged the results. The results were surprisingly stable except for LOCI and DB-Outlier detection, which can be attributed to their distance parameterization. We chose $k = 50$ for all kNN-based algorithms, and a radius dependant on the dimensionality for radius parameters across algorithms. Since this choice was not optimized for the individual algorithms, one should not compare the values of different algorithms but rather one should compare just the different normalizations of the same algorithm. The results (in percent) for the normally distributed data are shown in Table 1. Table 2 depicts

Table 3: PenDigits data set

	Lin	**LinM**	**Gau**	**GauC**	Γ	Rank	SigM	MixM
Gaussian	48.0	**30.3**	41.8	48.9	41.8	34.7	36.1	50.0
kNN [43]	30.9	30.0	**12.0**	34.5	13.0	26.7	18.7	24.3
agg. kNN [3]	30.4	29.3	**11.0**	33.7	11.6	25.9	18.4	18.6
LDOF [58]	46.3	47.0	38.8	45.7	38.9	39.6	**35.0**	46.2
LOCI [39]	40.9	39.2	**22.8**	40.0	24.3	30.0	26.2	51.0
LOF [9]	33.8	34.0	**13.3**	17.7	13.5	27.5	16.0	13.5
Refer. [40]	53.7	57.9	58.0	52.4	57.9	58.0	61.7	48.9
DB-Out. [28]	34.0	**15.8**	21.4	48.8	21.6	26.2	25.6	50.0
ABOD [33]	35.1	n/a	**18.5**	n/a	20.1	28.6	n/a	n/a

Table 4: Wisconsin Breast Cancer data set

	Lin	**LinM**	**Gau**	**GauC**	Γ	Rank	SigM	MixM
Gaussian	48.3	35.0	46.5	49.0	46.4	**29.1**	40.5	65.6
kNN [43]	30.5	29.9	**14.4**	35.4	15.1	27.8	17.3	15.5
agg. kNN [3]	30.7	30.1	**14.6**	35.1	15.3	27.8	17.0	15.8
LDOF [58]	31.6	30.7	**14.8**	35.7	15.2	27.8	18.7	20.2
LOCI [39]	37.9	38.0	15.9	26.8	17.7	27.8	**14.5**	21.3
LOF [9]	29.8	29.8	14.1	18.7	14.5	27.7	16.4	**13.9**
Refer. [40]	39.3	**23.9**	27.4	43.2	28.4	30.7	30.1	52.0
DB-Out. [28]	19.0	14.3	**13.9**	28.7	15.7	27.8	17.9	26.5
ABOD [33]	28.2	n/a	**14.0**	n/a	15.8	27.3	n/a	n/a

the results for the uniformly distributed data sets.

We compare different transformations, starting with a simple linear scaling to the interval $[0, 1]$ ("Lin"), a linear baseline transformation with $\mu_S = base_S$ ("LinM"), a Gaussian scaling ("Gau"), a customized Gaussian scaling adjusting $\mu_S = base_S$ ("GauC") and a Gamma scaling ("Γ"). As comparison scalings we include a naive transformation that discards the actual scores and just uses the resulting ranking (in $[0, 1]$ and denoted by "Rank") and the normalizations introduced by [17], namely Sigmoid fitting ("SigM") and mixture model fitting ("MixM"). In case of ABOD, we used the log-regularization in combination with linear, Gaussian, and Gamma scaling. As we do not want to compare the different methods but only the transformations, one should only compare different values from the same row. It can be seen, that for all methods, most scaling methods yield considerably better results (in terms of decreased error costs) compared to the original score ("Lin"). While the methods of [17] sometimes work quite well, they have shown to be unreliable due to the use of an EM-style model estimation. In particular the mixture model ("MixM") approach often converges to a "no outliers" or "all outliers" model. The overall results also suggest that a Gaussian scaling is — although not always the winner — a good choice. This demonstrates that the application of our proposed unification can clearly increase the usability of outlier scores.

5.2.2 Real Data To produce real-world data sets for use in outlier detection, we chose two well-known data sets from the UCI machine learning repository [5], known as "Pen Digits" and "Wisconsin Breast Cancer".

Table 5: Ensemble results

KDDCup1999 data set:

Ensemble construction	ROC AUC	Combination method
Feature Bagging LOF	0.7201	unscaled mean [34]
10 rounds,	0.7257	sigmoid mean [17]
dim $\in [d/2 : d-1]$,	0.7300	mixture model mean [17]
$k = 45$	0.7312	HeDES scaled mean [38]
	0.7327	maximum rank [34]
	0.7447	mean **unified score**
LOF [9]	0.6693	mixture model mean [17]
$k = 20, 40, 80, 120, 160$	0.7078	unscaled mean [34]
	0.7369	sigmoid mean [17]
	0.7391	HeDES scaled mean [38]
	0.7483	maximum rank [34]
	0.7484	mean **unified score**
Combination of	0.5180	mixture model mean [17]
different methods:	0.9046	maximum rank [34]
LOF [9], LDOF [58],	0.9104	unscaled mean [34]
kNN [43], agg.kNN [3]	0.9236	sigmoid mean [17]
	0.9386	HeDES scaled mean [38]
	0.9413	mean **unified score**

ALOI images with outliers:

Ensemble construction	ROC AUC	Combination method
Feature Bagging LOF	0.7812	mixture model mean [17]
10 rounds,	0.7832	sigmoid mean [17]
dim $\in [d/2 : d-1]$,	0.7867	maximum rank [34]
$k = 45$	0.7990	unscaled mean [34]
	0.7996	HeDES scaled mean [38]
	0.8000	mean **unified score**
LOF [9]	0.7364	mixture model mean [17]
$k = 10, 20, 40$	0.7793	maximum rank [34]
	0.7805	sigmoid mean [17]
	0.7895	HeDES scaled mean [38]
	0.7898	unscaled mean [34]
	0.7902	mean **unified score**
Combination of	0.7541	mixture model mean [17]
different methods:	0.7546	maximum rank [34]
LOF [9], LDOF [58],	0.7709	unscaled mean [34]
kNN [43], agg.kNN [3]	0.7821	sigmoid mean [17]
	0.7997	mean **unified score**
	0.8054	HeDES scaled mean [38]

To obtain outliers, one of the classes (corresponding to the digit 4 and to malignant tumors) was down sampled to 20 (Pen Digits) and 10 (Wisconsin) instances. This approach was adopted from [58]. To obtain more statistic reliability, we produced 20 random samples each, and averaged the results on these data sets. For control purposes, the standard deviation was recorded and is insignificant. Given that each of the classes in the data can well contain their own outliers, and it is just the down sampled class that we consider as interesting outliers, it is not surprising that the best result still incurred error costs of 10 to 15 percent. The results on the real world data sets are depicted in Table 3 (PenDigits) and Table 4 (Wisconsin). Again it can be observed that applying our transformations can considerably reduce the error costs which indicates an improved usability for discerning between outliers and inliers. In summary, it can be observed on synthetic as well as real-world data that our transformations work well. It is worth noting that none of the distributions (uniform, Gaussian, Gamma) assumed to model the values of the outlier scores is the winner in all experiments. However, it

is not the scope of this study to evaluate which distribution best fits which outlier score on which data settings. Rather, we wanted to show that using any distribution-based transformation can yield a significant boost in accuracy compared to the original methods.

5.3 Improvement in Outlier Ensembles We constructed ensembles for the KDDCup 1999 dataset as well as an image data set derived from the Amsterdam Library of Object Images [18] (50.000 images, 1508 of rare objects with $1-10$ instances) using three strategies to obtain diversity: "feature bagging", where a subset of dimensions is selected for each instance; LOF with different values of k on the full-dimensional dataset; and combination of four different methods (LOF, LDOF, kNN, and aggregate kNN). We did not include ABOD, since its inverted score would be unfair to use without regularization. We evaluated different methods for normalizing and combining the scores into a single ranking, and computed the ROC area under curve (AUC) value for each combination. For comparison, we implemented the methods of [17, 34, 38] (see Section 4).

Table 5 gives an overview of the results. Despite the trivial mean operator used, the use of unified scores produced excellent and stable results, in particular when combining the results of different algorithms, and even when the kNN methods did not work well on the ALOI image data.

6 Conclusion

State-of-the-art outlier scores are not standardized and often hard to interpret. Scores of objects from different data sets and even scores of objects from the same data set cannot be compared. In this paper, we proposed a path to the unification of outlier models by regularization and normalization aiming (i) at an increased contrast between outlier scores vs. inlier scores and (ii) at deriving a rough probability value for being an outlier or not. Although we took the first step on this path only, we demonstrated the impact of such a procedure in reducing error-costs. We envision possible applications of this unification in a *competition scenario*, i.e., comparing different methods in order to select the most appropriate one for a given application, as well as a *combination scenario*, i.e., using different methods complementary in order to combine their strengths and alleviate their weaknesses. In the first scenario, the improved possibilities of analyzing strengths and weaknesses of different existing methods may facilitate better direction of efforts in design and development of new and better approaches to outlier detection. For the latter scenario, we demonstrate the beneficial impact of our approach in comparison to previous attempts of building outlier ensembles. Future work could focus on deriving comparable probability values for a broader selection of methods, a better calibration, or on further tuning the transformation functions towards optimal contrast between outlier- and inlier-scores.

References

[1] N. Abe, B. Zadrozny, and J. Langford. Outlier detection by active learning. In *Proc. KDD*, 2006.

[2] E. Achtert, H.-P. Kriegel, L. Reichert, E. Schubert, R. Wojdanowski, and A. Zimek. Visual evaluation of outlier detection models. In *Proc. DASFAA*, 2010.

[3] F. Angiulli and C. Pizzuti. Fast outlier detection in high dimensional spaces. In *Proc. PKDD*, 2002.

[4] A. Arning, R. Agrawal, and P. Raghavan. A linear method for deviation detection in large databases. In *Proc. KDD*, 1996.

[5] A. Asuncion and D. J. Newman. UCI Machine Learning Repository, 2007. `http://www.ics.uci.edu/~mlearn/MLRepository.html`.

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

[7] G. E. A. P. A. Batista, R. C. Prati, and M. C. Monard. A study of the behavior of several methods for balancing machine learning training data. *SIGKDD Explor.*, 6(1):20–29, 2004.

[8] S. D. Bay and M. Schwabacher. Mining distance-based outliers in near linear time with randomization and a simple pruning rule. In *Proc. KDD*, 2003.

[9] M. M. Breunig, H.-P. Kriegel, R. Ng, and J. Sander. LOF: Identifying density-based local outliers. In *Proc. SIGMOD*, 2000.

[10] G. W. Brier. Verification of forecasts expressed in terms of probability. *Monthly Weather Review*, 78(1):1–3, 1950.

[11] N. V. Chawla, D. A. Cieslak, L. O. Hall, and A. Joshi. Automatically countering imbalance and its empirical relationship to cost. *Data Min. Knowl. Disc.*, 17(2):225–252, 2008.

[12] T. de Vries, S. Chawla, and M. E. Houle. Finding local anomalies in very high dimensional space. In *Proc. ICDM*, 2010.

[13] M. DeGroot and S. E. Fienberg. The comparison and evaluation of forecasters. *The Statistician*, 32(1/2):12–22, 1983.

[14] N. Delannay, C. Archambeau, and M. Verleysen. Improving the robustness to outliers of mixtures of probabilistic PCAs. In *Proc. PAKDD*, 2008.

[15] P. Domingos. MetaCost: a general method for making classifiers cost-sensitive. In *Proc. KDD*, 1999.

[16] H. Fan, O. R. Zaïane, A. Foss, and J. Wu. A nonparametric outlier detection for efficiently discovering top-N outliers from engineering data. In *Proc. PAKDD*, 2006.

[17] J. Gao and P.-N. Tan. Converting output scores from outlier detection algorithms into probability estimates. In *Proc. ICDM*, 2006.

[18] J. M. Geusebroek, G. J. Burghouts, and A. Smeulders. The Amsterdam Library of Object Images. *Int. J. Computer Vision*, 61(1):103–112, 2005.

[19] J. Hardin and D. M. Rocke. Outlier detection in the multiple cluster setting using the minimum covariance determinant estimator. *Comput. Stat. and Data Anal.*, 44(4):625–638, 2004.

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

[21] N. Japkowicz and S. Stephen. The class imbalance problem: A systematic study. *Intelligent Data Analysis*, 6:429–449, 2002.

[22] W. Jin, A. Tung, and J. Han. Mining top-n local outliers in large databases. In *Proc. KDD*, 2001.

[23] W. Jin, A. K. H. Tung, J. Han, and W. Wang. Ranking outliers using symmetric neighborhood relationship. In *Proc. PAKDD*, 2006.

[24] T. Jo and N. Japkowicz. Class imbalances versus small disjuncts. *SIGKDD Explor.*, 6(1):40–49, 2004.

[25] T. Johnson, I. Kwok, and R. Ng. Fast computation of 2-dimensional depth contours. In *Proc. KDD*, 1998.

[26] M. V. Joshi, V. Kumar, and R. Agarwal. Evaluating boosting algorithms to classify rare classes: Comparison and improvements. In *Proc. ICDM*, 2001.

[27] E. M. Knorr and R. T. Ng. A unified notion of outliers: Properties and computation. In *Proc. KDD*, 1997.

[28] E. M. Knorr and R. T. Ng. Algorithms for mining distance-based outliers in large datasets. In *Proc. VLDB*, 1998.

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

[30] H.-P. Kriegel, P. Kröger, E. Schubert, and A. Zimek. A general framework for increasing the robustness of PCA-based correlation clustering algorithms. In *Proc. SSDBM*, 2008.

[31] H.-P. Kriegel, P. Kröger, E. Schubert, and A. Zimek. LoOP: local outlier probabilities. In *Proc. CIKM*, 2009.

[32] H.-P. Kriegel, P. Kröger, E. Schubert, and A. Zimek. Outlier detection in axis-parallel subspaces of high dimensional data. In *Proc. PAKDD*, 2009.

[33] H.-P. Kriegel, M. Schubert, and A. Zimek. Angle-based outlier detection in high-dimensional data. In *Proc. KDD*, 2008.

[34] A. Lazarevic and V. Kumar. Feature bagging for outlier detection. In *Proc. KDD*, 2005.

[35] A. H. Murphy. A note on the utility of probabilistic predictions and the probability score in the cost-loss ratio decision situation. *J. Appl. Meteor.*, 5(4):534–537, 1966.

[36] A. H. Murphy and E. S. Epstein. Verification of probabilistic predictions: A brief review. *J. Appl. Meteor.*, 6(5):748–755, 1967.

[37] A. H. Murphy and R. L. Winkler. Reliability of subjective probability forecasts of precipitation and temperature. *Appl. Statist.*, 26(1):41–47, 1977.

[38] H. V. Nguyen, H. H. Ang, and V. Gopalkrishnan. Mining outliers with ensemble of heterogeneous detectors on random subspaces. In *Proc. DASFAA*, 2010.

[39] S. Papadimitriou, H. Kitagawa, P. Gibbons, and C. Faloutsos. LOCI: Fast outlier detection using the local correlation integral. In *Proc. ICDE*, 2003.

[40] Y. Pei, O. Zaïane, and Y. Gao. An efficient reference-based approach to outlier detection in large datasets. In *Proc. ICDM*, 2006.

[41] T. Pietraszek. Optimizing abstaining classifiers using ROC analysis. In *Proc. ICML*, 2005.

[42] J. C. Platt. Probabilities for SV machines. In A. Smola, P. Bartlett, B. Schölkopf, and D. Schuurmans, editors, *Advances in Large Margin Classifiers*, pages 61–74. MIT Press, 2000.

[43] S. Ramaswamy, R. Rastogi, and K. Shim. Efficient algorithms for mining outliers from large data sets. In *Proc. SIGMOD*, 2000.

[44] B. Raskutti and A. Kowalczyk. Extreme re-balancing for SVMs: a case study. *SIGKDD Explor.*, 6(1):60–69, 2004.

[45] P. Rousseeuw and K. Van Driessen. A fast algorithm for the minimum covariance determinant estimator. *Technometrics*, 41:212–223, 1999.

[46] I. Ruts and P. J. Rousseeuw. Computing depth contours of bivariate point clouds. *Comput. Stat. and Data Anal.*, 23:153–168, 1996.

[47] F. Sanders. The verification of probability forecasts. *J. Appl. Meteor.*, 6(5):756–761, 1967.

[48] S. Sarawagi, R. Agrawal, and N. Megiddo. Discovery-driven exploration of OLAP data cubes. In *Proc. EDBT*, 1998.

[49] P. Sun and S. Chawla. On local spatial outliers. In *Proc. ICDM*, 2004.

[50] J. Tang, Z. Chen, A. W.-C. Fu, and D. W. Cheung. Enhancing effectiveness of outlier detections for low density patterns. In *Proc. PAKDD*, 2002.

[51] J. Tukey. *Exploratory Data Analysis*. Addison-Wesley, 1977.

[52] J. von Neumann and O. Morgenstern. *Theory of Games and Economic Behavior*. Princeton University Press, 3rd edition, 1953.

[53] G. M. Weiss. Mining with rarity: A unifying framework. *SIGKDD Explor.*, 6(1):7–19, 2004.

[54] G. M. Weiss, K. McCarthy, and B. Zabar. Cost-sensitive learning vs. sampling: Which is best for handling unbalanced classes with unequal error costs? In *Proc. DMIN*, 2007.

[55] R. L. Winkler. Scoring rules and the evaluation of probabilities. *TEST*, 5(1):1–60, 1996.

[56] B. Zadrozny and C. Elkan. Learning and making decisions when costs and probabilities are both unknown. In *Proc. KDD*, 2001.

[57] B. Zadrozny and C. Elkan. Transforming classifier scores into accurate multiclass probability estimates. In *Proc. KDD*, 2002.

[58] K. Zhang, M. Hutter, and H. Jin. A new local distance-based outlier detection approach for scattered real-world data. In *Proc. PAKDD*, 2009.