Mach Learn (2015) 100:509–531
DOI 10.1007/s10994-015-5507-y

A decomposition of the outlier detection problem into a set of supervised learning problems

Heiko Paulheim[1] · **Robert Meusel**[1]

Received: 4 February 2015 / Accepted: 22 May 2015 / Published online: 20 June 2015

Abstract Outlier detection methods automatically identify instances that deviate from the majority of the data. In this paper, we propose a novel approach for unsupervised outlier detection, which re-formulates the outlier detection problem in numerical data as a set of supervised regression learning problems. For each attribute, we learn a predictive model which predicts the values of that attribute from the values of all other attributes, and compute the deviations between the predictions and the actual values. From those deviations, we derive both a weight for each attribute, and a final outlier score using those weights. The weights help separating the relevant attributes from the irrelevant ones, and thus make the approach well suitable for discovering outliers otherwise masked in high-dimensional data. An empirical evaluation shows that our approach outperforms existing algorithms, and is particularly robust in datasets with many irrelevant attributes. Furthermore, we show that if a symbolic machine learning method is used to solve the individual learning problems, the approach is also capable of generating concise explanations for the detected outliers.

Keywords Outlier detection · Machine learning · Outlier explanations

1 Introduction

Outlier or *anomaly detection* methods are used to identify observations that "appear to deviate markedly from other members of the same sample", i.e., that "appear to be inconsistent with the remainder of the data" (Barnett and Lewis 1994). In other words, the majority of the

Editors: João Gama, Indre Žliobaite, Alípio M. Jorge, and Concha Bielza.

✉ Heiko Paulheim
heiko@informatik.uni-mannheim.de

Robert Meusel
robert@dwslab.de

[1] Data and Web Science Group, University of Mannheim, Mannheim, Germany

data is supposed to follow certain *patterns*, and outliers do not adhere to those patterns. Typical applications of outlier detection include fraud and network-intrusion detection, or error detection in data (Hodge and Austin 2004).

Many classic outlier detection methods use the notions of *density* and *proximity*, i.e., they mainly identify outliers as data points that occur in sparsely populated areas of the dataset, and far away from neighboring points. To do so, they rely on distance measures. Therefore, they struggle from the *curse of dimensionality*, which render many classic distance measures useless once the dataset is of higher dimensionality (Aggarwal et al. 2001). The underlying problem is that at high dimensionality, most distance measures collapse in a way that all pairs of instances have a similar distance, which makes distance-based data mining approaches fail at such datasets.

More concisely, such algorithms struggle with datasets containing a larger number of attributes that are *irrelevant* for the outlier detection, since they expose no (or only very weak) meaningful patterns. For example, in a dataset which contains the height and age of children, outliers would be, e.g., children that are unusually tall or short for their age. If the dataset contains a large number of other attributes, such as database identifiers, social security numbers of both the child and its parents, ZIP codes, phone numbers, etc., many distance-based algorithms may yield suboptimal results, given that those attributes are not identified and removed upfront.

In this paper, we propose the *attribute-wise learning for scoring outliers* (ALSO) approach, which, instead of exploiting density, directly searches for patterns in the data. Such patterns are expected to present themselves as dependencies between the different attributes. To detect those patterns, we decompose the outlier detection problem into a set of supervised learning problems. Learning algorithms solving those problems return both the *patterns* underlying the data, as well as estimators for the *strength* of those patterns in each attribute. These strengths can be turned into *attribute weights*, assigning low weights to attributes exposing no or only very weak patterns. Outliers are then identified as data points which deviate from the patterns found, taking the weights into account when quantifying the deviation.

We show that for numerical datasets, the approach can be used in conjunction with arbitrary regression learning algorithms, that it reliably yields good results using M5' (regression trees) or isotonic regression as base learners, and that its results are invariant to the adding of irrelevant noise attributes. Furthermore, we demonstrate that when using a *symbolic* learning algorithm, concise explanations for the outliers found can be generated.

The rest of this paper is structured as follows. Section 2 discusses related work, and shows how ALSO is novel with respect to existing approaches. We introduce the ALSO approach in Sect. 3, and present its evaluation in Sect. 4. Section 5 discusses the generation of explanations for outliers. We conclude with a summary and an outlook on future work.

2 Related work

In the past decades, an abundance of methods have been proposed for outlier detection (Aggarwal 2013; Chandola et al. 2009; Hodge and Austin 2004). Chandola et al. (2009) distinguish three types of approaches: *supervised* approaches are trained based on labeled examples for both outliers and normal examples, *semi-supervised* approaches are trained using labeled examples only for normal observations, and *unsupervised* approaches that are built using no labeled data at all.

In that classification, the ALSO approach discussed in this paper is an *unsupervised* one—while the outlier detection problem is decomposed into a set of *supervised* learning problems, the overall approach is still unsupervised.

Classic unsupervised approaches identify outliers based on their distance to the nearest neighbors and/or on the local density around instances. Outlier detection approaches using machine learning have already been proposed, but many of them rely on labeled examples, i.e., they are supervised or semi-supervised. In the following, we provide an overview of unsupervised, machine-learning based approaches.

Clustering-based approaches like CBLOF and LDCOF reformulate the problem of outlier detection as a clustering problem, to be solved with any clustering algorithm. They first identify clusters in the data. Those clusters are considered the model which underlies the data; consequently, data points that are not contained in any cluster (or only in a very small cluster) are considered as outliers. Measures used as outlier scores are, e.g., the distance of a data point to the next cluster centroid (Amer and Goldstein 2012; He et al. 2003).

One-class support vector machines try learn a boundary around a set of training examples, i.e., they can be applied to learn a model in a semi-supervised setting. *Robust* one-class support vector machines are capable of dealing with datasets that contain outliers, i.e., they learn the boundary of the region where *most* of the examples are located in, and mark the examples outside that area as outliers (Amer et al. 2013; Xu et al. 2006). Thus, they can be used for unsupervised outlier detection as well.

Abe et al. (2006) propose a method that solves the outlier detection problem by generating artificial instances as outliers, and thus turning the outlier detection problem into a problem of supervised classification, using a sample of the given instances as normal points. The authors propose the use of active learning to optimize the sampling of normal points. Similarly, the work described in Yamanishi and Takeuchi (2001) follows a two-step approach: it first aims at fitting a statistical distribution to the data in order to assign outlier scores. Then, those outlier scores are used to train a rule learning classifier telling outliers from non-outliers, using the data points with the highest outlier scores as positive examples.

Frequent itemset mining finds typical patterns that occur in (usually categorical) data. He et al. (2005) propose an approach that first mine frequent patterns, and then mark those instances as outliers that do not match any frequent pattern, or only very rare patterns. In Padmanabhan and Tuzhilin (2000), a direct approach for mining *rare* patterns is introduced, which are assumed to describe outliers.

Replicator Neural Networks (RNNs) (Hawkins et al. 2002) are an approach which is close to the one introduced in this paper. The authors propose training a neural network where the training vectors for input and output are identical (i.e., the neural network tries to *replicate* the training instances as good as possible), and use the prediction error of the neural network as an outlier score for each instance. The ALSO approach described in this paper can be seen as a *generalization* of that approach, which is capable of using any learning strategy (i.e., it is not limited to neural networks), and it inherently learn weights for each attribute, so that the computation of the prediction error focuses on the more relevant attributes, which increases the robustness of the approach, in particular in higher dimensional cases.

The concept of trying to predict an attribute from the other attributes is not new. In Teng (1999), a *noise removal* method based on such predictions has been proposed. While the authors' focus is on *noise removal* on attribute level, i.e., replacing single attribute values which are likely to be wrong, our focus is on *outlier detection*, i.e., identifying entire instances which deviate from the majority of the data.

The idea of identifying outliers by comparing actual and expected attribute values is also used in the *Correlation Outlier Probabilities* (*COP*) approach (Kriegel et al. 2012), which

tries to identify local correlations between attributes. Based on those correlations, deviations between the expected and the actual attribute values can be computed. Similarly, *DEMUD* computes a singular value decomposition of the data, and computes an outlier score from the per-attribute deviations from the actual data points and the values created from their SVD-based reconstruction (Wagstaff et al. 2013).

Isolation forests rely on a particular kind of decision trees, i.e., *isolation trees*, to directly learn a model for outliers (in contrast to most of the other approaches discussed above, which try to learn a model for the normal data points). Isolation trees create their splits in a way that each leaf node contains only one instance (or a set of instances of the exact same value). If the trees are cut at a certain height, all instances ending up in leaf nodes can be considered outliers. In Liu et al. (2012), the idea is combined with random forests, training a set of isolation trees on different attribute subsets.

For many outlier detection algorithms, especially distance and density based ones, a high dimensionality of the data can be a problem (Aggarwal and Yu 2001), since outliers in lower dimensional subspaces are likely to be obscured by accidental similarities in other attributes. One recently proposed approach to cope with that problem is to use ensemble methods, which are also popular in machine learning (Zimek et al. 2013). In that approach called *Ensemble Subsampling*, an ensemble of outlier detection methods is created, where the members of the ensemble are outlier detection methods trained on different subspaces of the original vector space.

In contrast to the approaches above, which usually adopt *one* machine learning method so that it can be used for outlier detection, the ALSO approach introduced in this paper can use *any* regression learning algorithm (linear regression, regression/model trees, neural networks, etc.) and exploit it for outlier detection. Furthermore, we do not assume any particular distribution of the data.

Another feature of ALSO is that the approach does not only identify outliers with an outlier score, but is also capable of delivering explanations, a task which is addressed by only a few outlier detection methods so far. One exception is the aforementioned COP, which measures deviations between actual and expected attribute values. Thus, those deviations may also be used for generating explanations for outliers.

ART-E is a clustering-based approach, which uses the distances of outlier points to clusters per attribute to identify those attribute values which make a data point an outlier (Mejía-Lavalle and Sánchez Vivar 2009), i.e., which attribute value contributes most to the distance measure, and provide an explanation based on those attributes. Knorr and Ng (1999) introduce the notion of *non-trivial outliers* being outliers in a feature space which are not identified as outliers in any subspace of that feature space. This notion helps isolating those attributes which actually contribute to identifying outliers. Similar to that idea, the *SOREX* tool looks for attributes in subspaces that do well identify outliers (Müller et al. 2010).

While ALSO also exploits deviations on single attributes for creating explanations, the approach is also capable of creating more concise explanations when combined with a symbolic learning algorithm, as we show in Sect. 5.

3 Approach

The core idea of the ALSO approach is that patterns within the underlying data and outliers are two sides of the same coin. More precisely, as outliers are deviations from patterns, the ALSO approach works in two steps: it first learns the patterns for each attribute, and

then measures the deviation of the actual data from the patterns found. For computing those deviations, a weighted distance metric is used, with the weights being learned together with the patterns.

3.1 Overall approach

ALSO identifies patterns in the data by learning one predictive model per attribute, using the other attributes as features and the actual attribute values as labels. By doing so, those models can be applied to all attributes of an instance i at hand, and, by making a prediction for each pattern, create a counterpart i'. If i and i' are far away from each other, this means that i is a point that does not follow the patterns observed in the data, which makes it an outlier, as shown in Fig. 1. Following this notion, we use the distance between i and i' as an outlier score.

Formally, we consider each instance i as an n-dimensional numerical feature vector $\langle i_1, i_2, \ldots, i_n \rangle$, with $i_k \in \mathbb{R}$ for $1 \leq k \leq n$. Then, the counterpart i' is defined as

$$\begin{aligned} i' := < & m_1(i_2, \ldots, i_n), \\ & m_2(i_1, i_3, \ldots, i_n), \\ & \ldots, \\ & m_n(i_1, i_2, \ldots, i_{n-1}) >, \end{aligned} \tag{1}$$

where m_k is a predictive model trained on the attribute set, using the k-th attribute as a target, and all other attributes as features.

In the following, we use the *Euclidean distance* to calculate the distance between i and i'. Thus, the first account of the outlier score is

$$o_{unweighted}(i) := \sqrt{\sum_{k=1}^{n} (i_k - i'_k)^2} \tag{2}$$

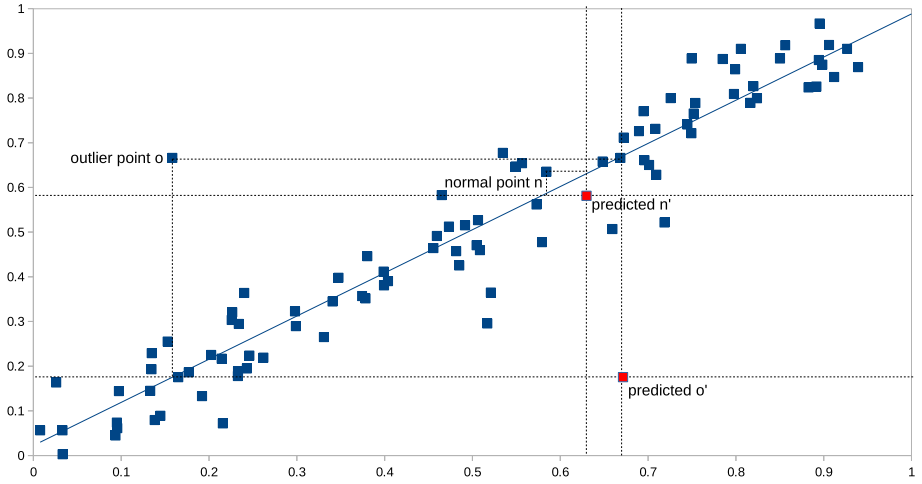


Fig. 1 Simplified example illustrating the idea of the ALSO approach. The example shows a two-dimensional problem with a linear regression model representing the global pattern in the data. The outlier point o has a larger distance to its predicted counterpart o' than the normal point n

All data is preprocessed using a standardization (i.e., z-transformation, normalizing the data so that the mean is 0 and the standard deviation is 1) in order to avoid skewing effects caused by different value ranges and distribution, and give deviations on different attribute scales equal influence on the final score.

3.2 Assigning weights to dimensions

To reduce the influence of attributes that do not expose strong patterns, we introduce weights w_i for all dimensions of the dataset, and refine the above definition to[1]:

$$o(i) := \sqrt{\frac{1}{\sum_{k=1}^{n} w_k} \sum_{k=1}^{n} w_k \cdot (i_k - i'_k)^2}, \tag{3}$$

The weights for dimensions are introduced in order to reduce the influence of meaningless attributes (such as IDs). Following the notion that we can only find outliers where we can observe patterns in the data, deviations on an attribute for which no patterns can be identified should not contribute to the outlier score.

Further, we want to emphasize deviations on attributes which expose very strong patterns, i.e., that are very well predictable from the others.

If we can find a pattern in an attribute, it means that we can predict the value of that attribute better than by mere guessing. For example, for database IDs, it is unlikely that we find a predictive model that works better than guessing,[2] while dependencies between attributes (e.g., age and height of children) lead to predictive models that are better than guessing.

Following that intuition, we compute the *root relative squared error* (RRSE) R_k for the k-th attribute, and use it to define a weight w_k for that attribute k:

$$w_k := 1 - min(1, R_k), \tag{4}$$

where the root relative squared error R_k on a dataset of m instances is defined as

$$R_k := \sqrt{\frac{\sum_{j=1}^{m} \left((i_j)_k - (i'_j)_k\right)^2}{\sum_{j=1}^{m} \left((i_j)_k - \overline{k}\right)^2}}. \tag{5}$$

In that equation, $(i_j)_k$ denotes the k-th feature of the j-th instance, i' is defined as in (1), and $\overline{k}$ is the average of the k-th feature.

Since a RRSE larger than 1 means that the predictive model is worse than guessing, we limit the values at 1 because it is irrelevant *how much worse* than guessing the learner is—in any case, the learner did not find any meaningful pattern for the attribute.

Given that a noise-tolerant learning algorithm is used for learning the individual models, meaningless attributes have no influence on the final outlier score. For a meaningless attribute, such as an ID or a ZIP code, a reasonable learning algorithm should not learn a model that is different from a default prediction (i.e., always predicting the mean value). Thus, in Eq. 5, all $(i'_j)_k$ would be equal to $\overline{k}$, thus, the RRSE of such a learner would be 1, which leads to a weight of zero. Since attributes with a weight of zero have no influence on the score

[1] In scenarios where only the *ordering* by scores is needed, the normalization by the inverse of the sum of weights as well as the square root can be omitted to simplify the computation.

[2] That is, unless the IDs show a correlation with other attributes, such as the IDs of orders may be correlated with the order time.

defined in (3), adding meaningless attributes to the dataset does not change the outlier score. In turn, this means that only attributes which expose meaningful patterns will be considered for computing the final computing outlier scores. That property of ALSO helps identifying outliers in high-dimensional datasets, which are otherwise obscured.

The use of such weights defines the a central property of the ALSO approach: if an attribute value deviates from its prediction given the learned model, it has a major impact on the final outlier score iff *both* the deviation as well as the attribute weight are high. This also gives way to a fundamental limitation of our approach: in datasets containing total redundancies, ALSO may fail to reliably identify outliers. As an extreme case, consider a dataset which contains temperature measurements made by a sensor, which are picked from the sensor and stored both in Celsius and Fahrenheit as two attributes. An extreme measurement made by the sensor is likely not to be revealed by ALSO since the two attributes are perfectly predictable from each other. Hence, no matter how far away from the majority of the values the extreme measurement is, it will not be identified as an outlier. While such constellations may happen in theory, we expect them to be rather rare in practical cases.

3.3 Nature of outliers found by ALSO

In our approach, we follow a different notion of *outliers* than, e.g., traditional distance or density based approaches. While in the latter, outliers are identified based on whether there are other data points close to them, we look at the underlying model instead, and identify outliers based on those models. This can lead to data points in sparse areas not identified as outliers, and vice versa. For example, in Fig. 1, a data point which is close to the regression line, but far away from the rest of the data (e.g., $\langle 100, 100 \rangle$) would not be recognized as an outlier.

Figure 2 illustrates those different notions by comparing ALSO (with isotonic regression as a base learner) and a typical density-based approach, i.e., *LoOP* (Kriegel et al. 2009). The plots show a two-attribute version of the Auto MPG dataset,[3] which describes cars and their fuel consumption, reduced to the two attributes *weight* (x-axis) and *MPG* (y-axis).

LoOP finds identifies data points as outliers which have both a high weight and a high consumption[4] (bottom right corner) and cars that have a low weight (left side), since these regions are less dense in the dataset (i.e., the average distance to the nearest neighbors is large). In contrast, ALSO identifies those data points as outliers which are not inline with the underlying model (i.e., heavier cars have a higher consumption). This holds in particular for cars that have a high consumption at a low weight, and vice versa (data points to the lower left and upper right of the majority of the data points).

3.4 Algorithm

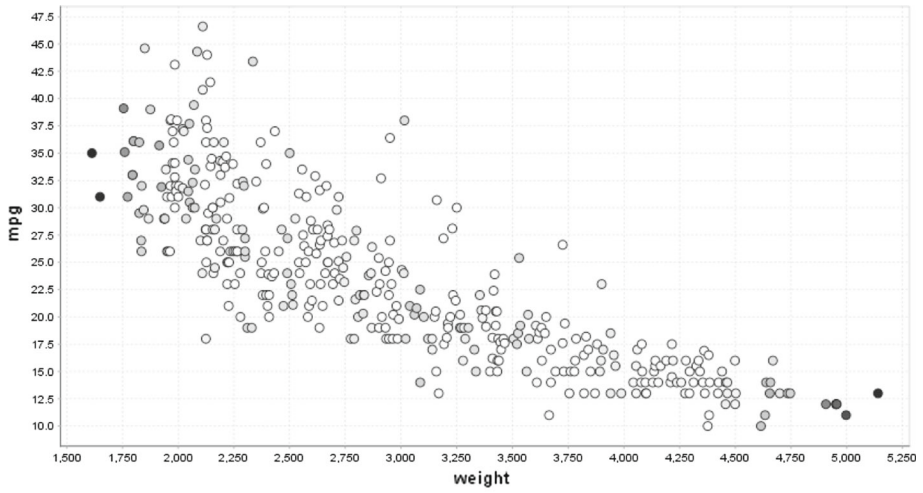
In order to avoid overfitting to outliers, we train the individual predictive models in different folds, using a cross-validation like scenario. The whole algorithm is shown in Algorithm 1.

The function $weight(error)$ computes the weights from the absolute squared errors by first normalizing them to a RRSE, and then applying Eq. 4. The weights for each attribute are learned on the fly while the algorithm proceeds, allowing it to adapt to the dataset at hand.[5]

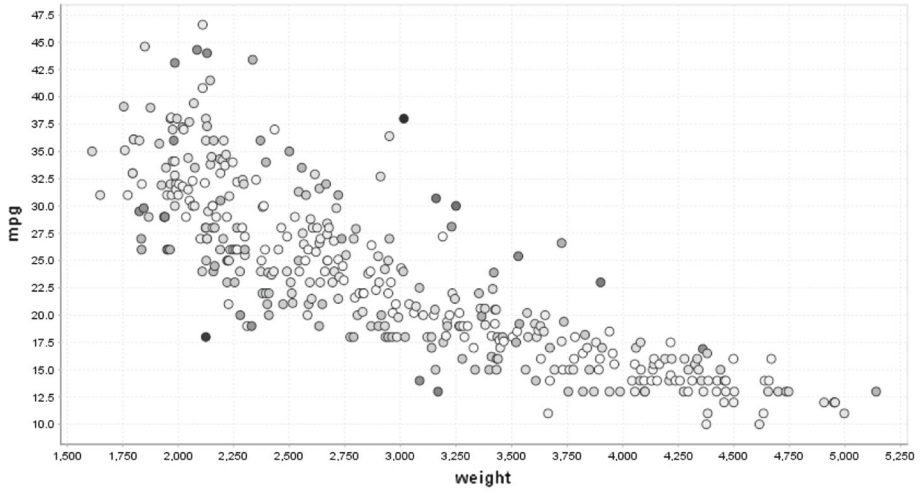
[3] http://archive.ics.uci.edu/ml/datasets/Auto+MPG.

[4] Note that a *high* consumption is reflected by a *low* MPG value.

[5] Note that for computing $\bar{k}$ as in (5), we take the average of the *overall* dataset, not the average per fold. The rationale is that for a dataset where a feature has many equal values, onefold may by coincidence only contain



(a)



(b)

Fig. 2 Comparison of outlier scores of LoOP and ALSO with isotonic regression on the UCI Auto MPG dataset, depicting *weight* on the x-axis and *MPG* on the y-axis. Each *circle* is an instance in the dataset; a *darker color* indicates a higher outlier score. **a** LoOP (k = 25), **b** ALSO with isotonic regression

Given that the algorithm is run on a dataset with n attributes and m instances, using f folds, $f \cdot n$ predictive models have to be trained and applied. With a learner that has a complexity $C_T(m, n)$ for training the model and $C_A(m, n)$ for applying the model on *one* instance, the

Footnote 5 continued
the same value in the test set, which would result in R_k being undefined. Moreover, if a feature contains *only* equal values in the whole dataset, we set its weight to 0, since it is of no use for determining the outlier score of an instance.

Algorithm 1: Pseudocode of the ALSO algorithm

Data: I: Set of m instances with n attributes
Result: An outlier score $score(i)$ for each instance $i \in I$

1 **foreach** $i \in I$ *and each attribute* k **do**
2 $\quad score(i,k) = 0$
3 **end**
4 **for** $k = 1$ *to number of attributes* **do**
5 $\quad error_k = 0$
6 $\quad$ **for** $f = 1$ *to number of folds* **do**
7 $\quad\quad S :=$ the f-th $1/f$ instances in I
8 $\quad\quad T := I \backslash S$
9 $\quad\quad$ set k-th attribute as learning target (i.e., label)
10 $\quad\quad$ train predictive model m_k on T
11 $\quad\quad$ **foreach** $i \in S$ **do**
12 $\quad\quad\quad score(i,k) = (i_k - m_k(i))^2$
13 $\quad\quad\quad error_k = error_k + (i_k - m_k(i))^2$
14 $\quad\quad$ **end**
15 $\quad$ **end**
16 $\quad$ compute R_k as $\sqrt{\frac{error_k}{\sum_{j=1}^{m} ((i_j)_k - \bar{k})^2}}$
17 $\quad$ compute weight w_k as $1 - min(1, R_k)$
18 **end**
19 **foreach** $i \in I$ **do**
20 $\quad score(i) = \sqrt{\frac{1}{\sum_{k=1}^{n} w_k} \sum_{k=1}^{n} w_k \cdot score(i,k)}$
21 **end**

overall complexity of ALSO is $O(n \cdot C_T(m,n) + n \cdot m \cdot C_A(m,n))$, since f is a constant factor.

It is, however, noteworthy that the loop in lines 4–17 can be executed in parallel, since there are no carry over variables. The same holds for the inner loop in lines 6–15, i.e., the training of the different predictive models (which will be the most costly operation), as well as their application to create the predicted counterparts for data points, can be performed in parallel. Given enough parallel computing resources, this can considerably reduce the overall runtime to $O(C_T(m,n) + m \cdot C_A(m,n))$. Since for most learners, $C_T(m,n) > C_A(m,n)$, the overall runtime can be reduced even to $O(C_T(m,n))$ in a massively parallel setting, i.e., the runtime of training one single predictive model for one attribute.

3.5 Interpretation of outlier scores

In a first step, the ALSO algorithm computes an individual outlier score for each attribute of each instance [shown as the value $score(i,k)$ in Algorithm 1]. These outlier scores depict the squared deviation of that attribute from what would be expected given the underlying model learned. As we use standardized data, the unit of this deviation is squared standard deviations, i.e., an outlier score of 4 for an attribute means that the value is 2 standard deviations away from the expected values.

The overall outlier score is a weighted average of those deviations, i.e., an outlier score of m means that the attribute values are on average m standard deviations away from the expected values, where deviations on attributes that are more relevant for the outlier detection obtain a higher weight. To illustrate that interpretation, consider an instance for which all attribute values are m standard deviations away from the predicted value, i.e., in standardized data,

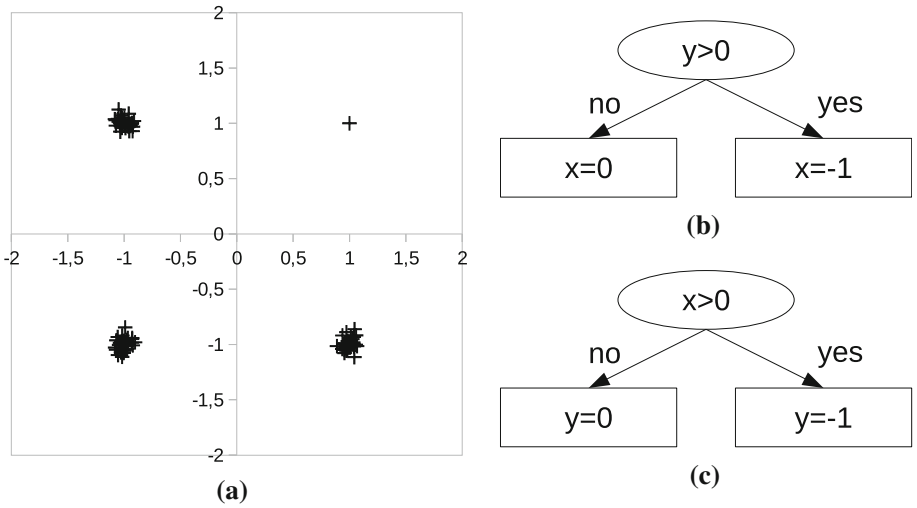


Fig. 3 Example with three clusters and one outlier. **a** Dataset, **b** regression tree for y, **c** regression tree for x

$i_k - i'_k = m$. This means that Eq. 3 collapses to

$$o(i) = \sqrt{\frac{1}{\sum_{k=1}^{n} w_k} \sum_{k=1}^{n} w_k \cdot (i_k - i'_k)^2} = \sqrt{\frac{1}{\sum_{k=1}^{n} w_k} \sum_{k=1}^{n} w_k \cdot m^2} = m \tag{6}$$

Thus, when using standardized data, it is possible to find outliers given user-defined boundaries, e.g., all data points deviating from their expected values by more than two standard deviations.

3.6 Example

Consider the dataset depicted in Fig. 3a, consisting of three roughly equally sized clusters centered around $(-1, 1)$, $(-1, -1)$, and $(1, -1)$, and one outlier point in $(1, 1)$. If we use regression trees as a modeling approach (which have shown to work well with our approach, see below), we obtain two regression trees, depicted in Fig. 3b, c. In the following, we compute the outlier scores for one point from each of the clusters, using those trees. Since the dataset and the two models are symmetric, both models have the same accuracy, thus, we can omit the weights in our computation:

$$o(-1, 1) = \sqrt{(-1 - (-1))^2 + (-1 - 0)^2} = 1 \tag{7}$$

$$o(-1, -1) = \sqrt{(-1 - 0)^2 + (-1 - 0)^2} = 1.414 \tag{8}$$

$$o(1, -1) = \sqrt{(1 - 0)^2 + (-1 - (-1))2} = 1 \tag{9}$$

$$o(1, 1) = \sqrt{(1 - (-1))^2 + (1 - (-1))^2} = 2.828 \tag{10}$$

Thus, in this example, the outlier score for the outlier point is twice as large as the largest outlier score for a normal point. This example demonstrates two important properties of the ALSO approach:

1. It produces sensible results also in cases where there is no simple functional dependency between the attributes.

2. Even if the underlying regression models learned are far from perfect (note that the regression trees predict 0 instead of a correct 1 or −1 value in the majority of cases), they are good enough for our purpose of computing outlier scores.

In the following, we will analyze the behavior of ALSO on a number of real-world datasets.

4 Empirical evaluation

To evaluate our approach, we have created a set of datasets, which are derived from real world datasets for classification and regression. On those datasets, we compare ALSO with different base learner to a number of both well-established as well as recent outlier detection methods.

4.1 Datasets

Canonical datasets for evaluating outlier detection are rare. Hence, we follow an approximation as suggested in Emmott et al. (2013): starting from standard classification and regression datasets (which can be found in dataset collections quite frequently), we define a subset of the instances, typically one of the classes or unusually high and low regression target values, as outliers, and sample it to a smaller share.[6]

We selected twelve real-world datasets from UCI, four of which were also used by Amer et al. (2013). In particular, we use six classification dataests, i.e., *Shuttle*, *Satellite*, *Wisconsin Breast Cancer*, *Ionosphere*, *Glass*, and *Seismic Bumps*, as well as six regression datasets, i.e., *Concrete*, *Parkinsons Telemonitoring*, *White Wine Quality*, *Housing*, *CCPP*, and *Energy Efficiency*. The datasets were selected by looking for classification and regression datasets with only numerical attributes. Furthermore, we aimed at a selection of both smaller and larger datasets, with different numbers of attributes.

As proposed in Emmott et al. (2013), we use the class attribute of the classification datasets to divide the datasets into normal and outlier points, defining one class (usually the smallest) as outliers, and sample the outlier class to a smaller size. The original class attribute is removed and retained for evaluation. For the first three datasets, we used the already preprocessed data also used in Amer et al. (2013).[7]

For the regression datasets, Emmott et al. (2013) propose to split the dataset into two classes at the mean of the regression target, and treat the datasets like classification datasets. Here, we decided to use extreme values of the regression target as outliers instead, which comes closer to the intended semantics of an outlier. Hence, we use all data points with a regression target within one standard deviation from the mean as normal points, and all data points further than two standard deviations away as outliers. Again, the original regression attribute is removed and retained for evaluation.[8]

[6] A typical drawback with this approach is there can be also outliers within the set defined as normal data points, which manifest as unusual attribute values or combinations thereof, but with a majority class label or an average regression target. If an outlier detection algorithm correctly identifies those outliers, they count as false positives. However, since the problem equally exists for all approaches at hand, a fair comparison of approaches is still possible even in the presence of this drawback.

[7] Source: http://madm.dfki.de/downloads.

[8] For the Parkinsons Telemonitoring and the Energy Efficiency datasets, different regression targets exist. Here, all the original target variables were removed.

Table 1 Modified datasets used for evaluation, including final dataset size (in number of instances), percentage of sampled outliers, and the mean μ and standard deviation σ of each dataset

Dataset (DS)	Original # inst.	# Att.	Outlier class(es)	Resulting # inst.	Final sampl. outlier pct. (%)	μ of att. values	σ of att. values
Satellite	6435	36	2,4,5	5100	1.49	86.000	18.000
Shuttle	50,000	9	2,3,5,6	46,464	1.89	29.185	68.892
Breast cancer	469	30	M	367	2.72	870,423	$2.1E+07$
Ionosphere	351	32	b	233	3.40	0.292	0.520
Glass	214	9	5,6,7	170	4.11	11.265	22.124
Seismic bumps	2584	19	1	2584	6.57	5885.75	$3.5E+09$
Concrete	1030	9	CCS	711	5.63	298.6	$1.2E+05$
Parkinsons tele.	5875	26	Total_UPDRS	4170	7.53	9.55	763.75
Wine quality	4898	12	Quality	3847	4.99	18.43	1726.01
Housing	506	14	MEDV	334	5.09	75.82	22,396.09
CCPP	9569	4	PE	5974	2.54	290.13	$1.7E+05$
Energy efficiency	768	8	Y1	492	1.44	167.98	56,871.52

Table 1 shows the final size and outlier characteristics of the used datasets after preprocessing.[9]

Since we want to investigate the influence of irrelevant attributes on the outlier detection performance, we furthermore created three additional versions of each dataset by adding a certain number (10, 50, and 100 % of the original number of attributes) of random noise attributes. The attribute values of each random noise attribute are drawn from a normal distribution, with its mean μ and standard deviation σ equal to the mean and standard deviation of the original attribute values of the corresponding dataset. Thus, our evaluation was eventually carried out on 48 datasets.

4.2 Setup

For evaluating the ALSO approach, we use three different base learners, i.e., *Linear Regression*, *Isotonic Regression* (Barlow et al. 1972), and *M5'* (Quinlan et al. 1992) as base learners.[10] For all learners, we use the implementation in *Weka*.[11] For M5', we learn pruned *regression* trees with a minimum leaf size of 4; linear and isotonic regression were run with their respective standard settings.

The choice for those algorithms is to have a larger variety in the types of models that can be learned. Linear regression learns only linear functions, isotonic regression learns an arbitrary monotonically increasing or decreasing function, and M5' can also cover more complex functions.

[9] Note that, although downloaded from the web page by the authors of Amer et al. (2013), the percentage of outliers is different than reported in their papers. Furthermore, the number of attributes in the Ionosphere dataset differs from the paper.

[10] We intentionally did not consider support vector machine (SVM) regression, since SVMs require careful parameter tuning, which is difficult to achieve in our case since learning the model for an attribute represents a new learning problem for each attribute, for which the SVM parameters would have to be tuned individually.

[11] http://www.cs.waikato.ac.nz/ml/weka/.

The number of folds for learning and evaluating the models for each attribute was fixed to 10.

We compare the ALSO approach to the following classic outlier detection methods:

1. The *k-NN global anomaly score* (*GAS*) is the average distance to the k nearest neighbors (Angiulli and Pizzuti 2002), following the intuition that outliers are located in rather sparsely populated areas of the vector space. We use GAS with $k = 10$, $k = 25$, and $k = 50$.
2. The *Local Outlier Factor* (*LOF*) is computed from the density of data points around the point under inspection, which in turn is computed from the distances to the k nearest neighbors (Breunig et al. 2000). Similar to our setup for GAS, we use $k_{min} = 10$ and $k_{max} = 50$.
3. The *Local Outlier Probability* (*LoOP*) follows a similar idea as LOF, but maps the outlier scores to probabilities in a [0; 1] interval (the scores assigned by other methods are usually unbound) (Kriegel et al. 2009). Like for GAS, we compute LoOP with $k = 10$, $k = 25$, and $k = 50$.

In addition to those methods, we also evaluate several approaches already discussed in Sect. 2:

4. The *Cluster-based Local Outlier Factor* (*CBLOF*) was used in conjunction with use the X-means algorithm, which restarts k-means with different values for k, in order to find an optimal one (Pelleg et al. 2000), using $k_{min} = 2$ and $k_{max} = 60$.
5. The *Local Density Cluster-based Outlier Factor* (*LDCOF*) was used with X-means in the same configuration as above.
6. We use *one-Class Support Vector Machines* (see Sect. 2 with three different kernels: the standard kernel (1-classSVM$_1$) as well as the robust kernel (1-classSVM$_r$) and eta kernel (1-classSVM$_e$) defined particularly for outlier detection (Amer et al. 2013).
7. For *Replicator Neural Networks* (*RNN*), we follow the setup in Hawkins et al. (2002), using three hidden layers (size 35, 3, and 35), and 1000 iterations.
8. For *COP*, which requires the identification of nearest neighbors, we use $k = 3 * d$, where d is the number of dimensions.
9. For *ensemble subsampling*, we follow the setup in Zimek et al. (2013), using LOF as a base outlier method, with 20 ensemble members, and averaging the outlier scores.
10. As reported in Liu et al. (2012), *isolation forests* provide stable results if at least 30 trees are learned, and the best results are achieved with a height limit of 1, so we use those values.

For the first six approaches, we use the implementation available in the RapidMiner Anomaly Detection extension (Goldstein 2014).[12] COP and the ensemble method were executed using the ELKI data mining framework (Achtert et al. 2008).[13] The standard settings of those algorithms in the respective tools have been used unless specified otherwise. For isolation forests, we use an implementation in R.[14] The RNN approach was built in Java using the Encog library.[15] Our own approach was built in Java as a RapidMiner operator, which allows for nesting arbitrary base learners, accessing the Weka base learners via RapidMiner's Weka extension.[16] Our implementation in RapidMiner supports full parallelization of the individual learning problems, as discussed above (Fig. 4).

[12] http://madm.dfki.de/rapidminer/anomalydetection.

[13] http://elki.dbs.ifi.lmu.de/.

[14] http://sourceforge.net/projects/iforest/.

[15] http://www.heatonresearch.com/encog.

[16] http://marketplace.rapid-i.com/UpdateServer/faces/product_details.xhtml?productId=rmx_weka.

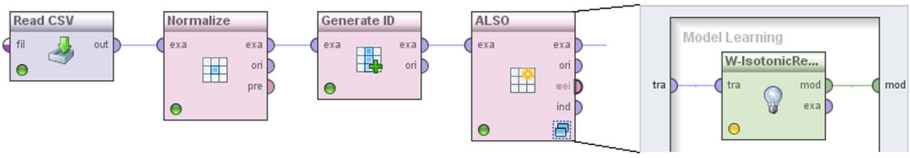


Fig. 4 Implementation of our approach in RapidMiner. The ALSO operator allows for using an arbitrary learning algorithm in a nested process

For all approaches, we compare the area under the ROC curve (AUC). To that end, outlier scores are computed for each instance, and the instances are ordered by those scores. The ROC curve is then drawn using the outliers as the positive class, plotting the true positive rate (y-axis) against the false positive rate (x-axis).

Since different tools were used for conducting the experiments (as there is no single tool implementing all the approaches at hand), we omit a comparison of runtimes, since such a comparison would be skewed, e.g., by the different programming languages and data storage strategies of the individual tools. However, if any of the approaches was not able to process a dataset within 12 h, we canceled the run. For being fair on this policy, we ran ALSO in single thread mode. All processes and datasets used for the evaluation, as well as the full set of individual experimental results, can be found online.[17]

4.3 Results

Table 2 depicts the average results on the twelve datasets, with different amounts of irrelevant attributes added. We show the results for 16 compared approaches from the literature, as well as for ALSO with three different base learners. Two observations can be made from these results:

- ALSO yields the best results when using M5' as a base learner, significantly outperforming all of the compared approaches. The results using Isotonic Regression are slightly worse, while using Linear Regression as a base learner is clearly inferior.
- Most of the compared approaches yield significantly worse results when adding different amounts of irrelevant attributes. 1-class SVM with a robust kernel, COP, and Ensemble Subsampling are the only approaches that do not show a significant decrease for any of the modified datasets (the increase for COP is not statistically significant). For ALSO, when inspecting the weights learned for random noise attributes, those are always close to 0.

In our experiments, the *Shuttle* dataset poses scalability problems to many approaches, with COP as well as all three 1-class SVM variants not being able to compute them within 12 h.

For ALSO, Linear regression performs particularly worse than the other base learners on the *Shuttle* dataset. The reason for this is that this dataset has some attributes which have very strong linear trends, but the outlier points are well in line with those linear trends as well, as shown in Fig. 5a. In those cases, these attributes get a high weight (according to their strong trend), which leads to outliers not being properly recognized, and false positives being marked as outliers, as the figure shows. In contrast, the *Ionosphere* dataset, on which linear regression works very well, has attributes with linear trends, where the outlier points are far away from the trend line, as shown in Fig. 5b. Other base learners are capable of identifying

[17] http://dws.informatik.uni-mannheim.de/en/research/attribute-wise-learning-for-scoring-outliers.

Table 2 Results reporting the average AUC across all twelve datasets. 0.0 denotes the results on the unmodified datasets; 0.1, 0.5, and 1.0 are the results on the datasets with different amounts of added noise attributes

Algorithm	0.0	0.1		0.5		1.0	
GAS (k = 10)	.789*	.669**	(−15.21 %*)	.613**	(−22.31 %**)	.611**	(−22.56 %**)
GAS (k = 25)	.792*	.651**	(−17.80 %**)	.617**	(−22.10 %*)	.652**	(−17.68 %**)
GAS (k = 50)	.777*	.638**	(−17.89 %*)	.642**	(−17.37 %*)	.613**	(−21.11 %*)
LOF	.777*	.648**	(−16.60 %*)	.627**	(−19.31 %*)	.629**	(−19.05 %*)
LoOP (k = 10)	.678**	.580**	(−14.45 %)	.597**	(−11.95 %*)	.574**	(−15.34 %)
LoOP (k = 25)	.730**	.586**	(−19.73 %*)	.586**	(−19.73 %*)	.597**	(−18.22 %)
LoOP (k = 50)	.727**	.608**	(−16.37 %**)	.591**	(−18.71 %*)	.585**	(−19.53 %)
CBLOF	.654**	.567**	(−13.33 %)	.548**	(−16.21 %**)	.517**	(−20.9 %*)
LDCOF	.749**	.642**	(−14.29 %*)	.579**	(−22.70 %*)	.605**	(−19.23 %**)
1-ClassSVM$_1$	.740*	.642**	(−13.24 %)	.548**	(−25.95 %*)	.588**	(−20.54 %*)
1-ClassSVM$_r$	.636**	.579**	(−8.96 %)	.563**	(−11.48 %)	.603**	(−5.19 %)
1-ClassSVM$_e$	.743*	.630**	(−15.21 %)	.560**	(−24.63 %**)	.604**	(−18.71 %*)
RNN	.731*	.669**	(−8.48 %*)	.651**	(−10.94 %)	.652**	(−10.81 %)
COP	.704**	.746**	(+5.97 %)	.754**	(+7.10 %)	.728**	(+3.41 %)
Ensemble	.731**	.643**	(−12.04 %)	.626**	(−14.36 %)	.629**	(−13.95 %)
iForest	.781*	.765*	(−2.05 %)	.749**	(−4.10 %*)	.732**	(−6.27 %*)
ALSO (M5')	.854	.854	(±0.00 %)	.853	(−0.12 %)	.852	(−0.23 %)
ALSO (Iso)	.848	.848	(±0.00 %)	.849	(+0.12 %)	.836	(−1.42 %)
ALSO (LR)	.749	.744	(−0.67 %)	.745	(−0.53 %)	.747	(−0.27 %)

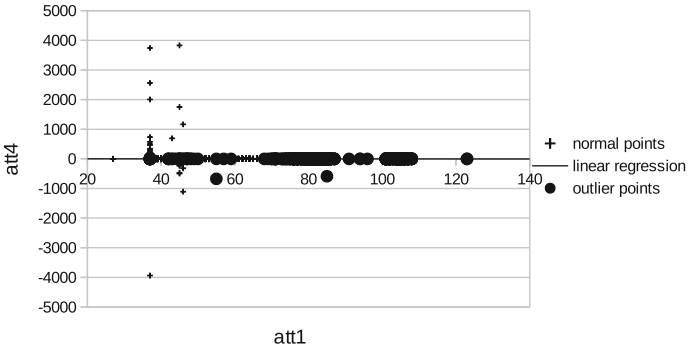
Statistical significance of the deviation between a result and the best performing setup of ALSO (with M5') are marked with * (<0.05) and ** (<0.01), determined using a one-sided paired *t* test. The numbers in parentheses denote the loss of result quality with respect to the unmodified dataset, also marked with statistical significance (i.e., whether the results with added noise attributes differ significantly from those achieved on the unmodified datasets)

patterns in attributes which are not linear, and which allow for better separation of normal and outlier points.

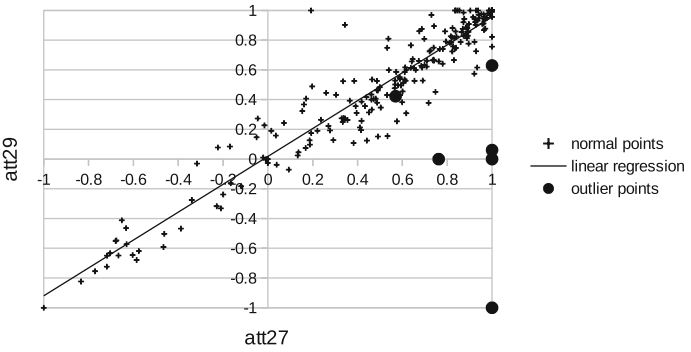
The results above show that ALSO delivers very good results, and that the result quality is stable across different datasets, even in the presence of many random noise attributes. This finding is underlined by the depiction of ranges shown Fig. 6, which shows that all approaches but ALSO with M5' and Isotonic Regression yield an AUC below 0.5 (i.e., a result worse than guessing) for at least one dataset.

Moreover, our experiments have shown that base learners capable of learning non-linear regression functions outperform linear regression, and that M5' slightly outperforms Isotonic Regression (i.e., an approach restricted to learning *monotonic* functions). This indicates that learning algorithms which can learn more complex models are usually better suited for ALSO. In practice, a learning algorithm for ALSO should be chosen that (a) is capable of handling irrelevant features, (b) is capable of learning non-linear functions, (c) does not require extensive parameter tuning to work well for a given problem, and (d) is reasonably performant for the amount of models to be learned.

In addition to comparing the AUC values achieved by the different approaches, we also compared the *ranks* of the approaches across the different datasets. Following Demšar (2006), we first use a Skillings–Mack test (Skillings and Mack 1981), a variant of the Friedman



(a)



(b)

Fig. 5 Examples for linear correlations in two of the real-world datasets, showing both normal and outlier points. **a** Linear correlation of two example attributes in the shuttle dataset, **b** linear correlation of two example attributes in the ionosphere dataset

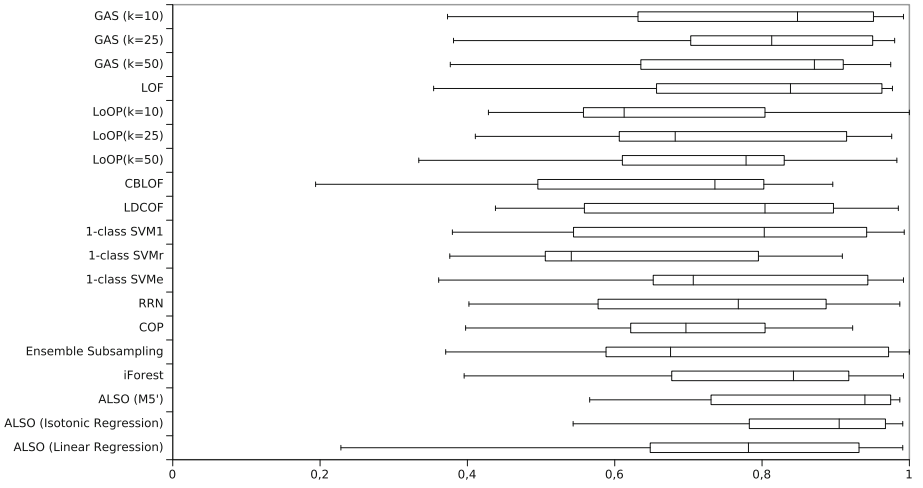


Fig. 6 Box plot of the different approaches' results (AUC values) on the real-world datasets, including different amounts of random attributes

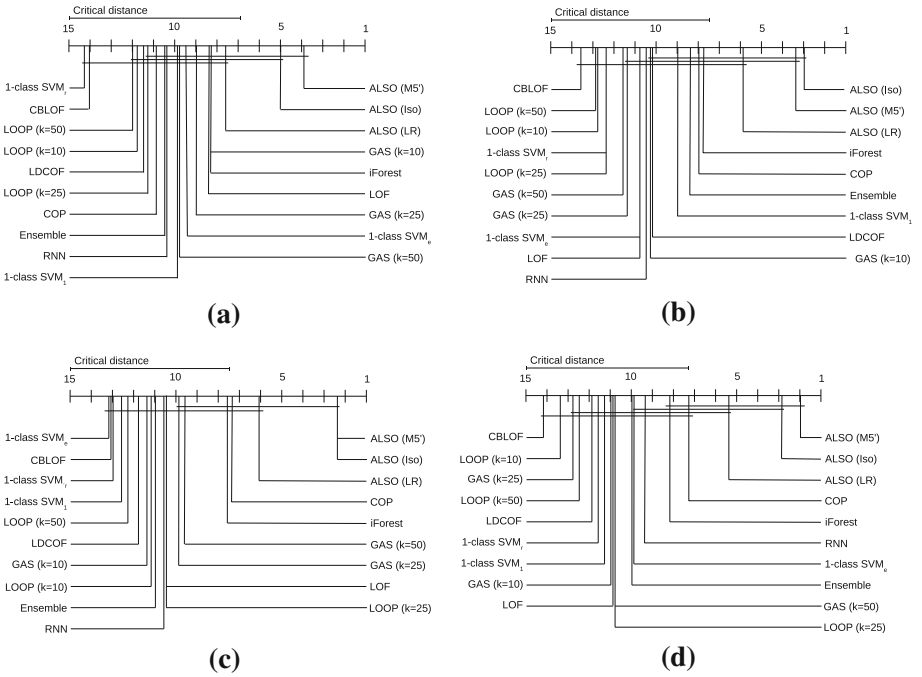


Fig. 7 Critical distance diagrams using a Nemenyi post hoc test at $p = 0.10$. The diagrams depict the average ranks of the compared approaches. Approaches whose ranks do not differ significantly according to the Nemenyi test are connected with a bar. **a** Unmodified datasets, **b** 10 % irrelevant attributes added, **c** 50 % irrelevant attributes added, **d** 100 % irrelevant attributes added

test which can also be applied to data with missing observations[18], to confirm that there are significant differences in the approaches' ranks. The significance of the individual differences has then been determined using a Nemenyi post-hoc test (Nemenyi 1962).

The results of the rank comparison are summarized in the critical distance diagrams in Fig. 7. Following Demšar (2006), we use a significance level of $p = 0.10$ for the Nemenyi test. It can be observed that especially for the datasets with more irrelevant attributes added, ALSO—especially with M5' and Isotonic Regression as base learners—are ranked significantly better than most of the state of the art approaches.

4.4 Experiments on higher dimensional datasets

To show that ALSO also provides valid results in higher dimensional data, we chose four datasets with more than 100 attributes from UCI. We preprocessed them in the same manner described in Sect. 4.1, but no variants with additional random attributes were generated. The datasets were selected as to have many attributes, but only a moderate number of instances, in order to keep the experiments tractable. Table 3 shows the datasets used.

Table 4 depicts the results achieved on the higher dimensional datasets. For ALSO, we restrict ourselves to using M5' as a base learner, which has been shown to work best in the experiments described above. It can be observed that ALSO outperforms existing approaches

[18] Since not each approach was able to finish on every dataset, missing observations occur in our setting.

Table 3 Modified high dimensional datasets used for evaluation, including final dataset size (in number of instances), percentage of sampled outliers, and the mean μ and standard deviation σ of each dataset

Dataset (DS)	Original # inst.	# Att.	Outlier class(es)	Resulting # inst.	Final sampl. outlier pct. (%)	μ of att. values	σ of att. values
Comm. and crime	1994	128	>2000	297	6.06	$1.0E+04$	$6.7E+09$
Internet adv.	3279	1558	*Ad*	2107	6.12	0.138	24.269
Multiple features	2000	649	1–9	218	8.25	114.257	$7.5E+04$
Urban land cover	508	148	*Shadow*	508	5.90	346.920	$2.2E+06$

Table 4 Results on the four high dimensional datasets: communities and crime (CC), Internet Advertisements (IA), multiple features (MF), and urban land cover (ULC). The approach marked with "–" means did not finish on the respective dataset within 24 h. For the Internet Advertisements dataset, no valid configuration of COP was possible (marked "X"): values for k smaller than d lead to an error, while smaller ones were not accepted as valid values

The best performing approach for each dataset are marked in bold.

Approach	CC	IA	MF	ULC	Avg.
GAS (k = 50)	0.569	0.528	0.981	0.392	0.618
LOF	0.568	0.466	0.980	0.377	0.598
LoOP (k = 10)	0.611	0.566	0.981	0.633	0.698
LoOP (k = 25)	0.649	0.690	0.681	0.735	0.689
LoOP (k = 50)	0.596	0.629	0.974	0.716	0.729
CBLOF	0.621	0.566	0.983	0.668	0.710
LDCOF	0.429	0.355	0.166	0.694	0.411
1-Class SVM_1	0.579	0.374	0.558	0.653	0.541
1-Class SVM_r	0.594	0.139	0.980	0.508	0.555
1-Class SVM_e	0.545	0.781	0.977	0.503	0.702
RNN	0.489	**0.856**	0.958	0.236	0.635
COP	0.715	X	–	0.641	0.678
Ensemble	0.614	0.591	0.986	0.656	0.712
iForest	0.742	0.649	0.911	0.774	0.769
ALSO (M5')	**0.761**	0.707	**0.995**	**0.856**	**0.830**

on average as well as on most of the datasets, except for the Internet Advertisements dataset, on which RNN and the 1-class$_e$ SVM perform better.

The problem with the latter dataset is that it is mostly a sparse dataset. The majority of the dataset are binary variables, most of which are very sparse, i.e., they contain mostly 0s. That, in turn, means that it is likely that a model for one of those attribute is trained on a training set with mostly 0s as labels, and, hence, only a default model is learned. That, in turn, assigns the weight of 0 to most of the attributes. This means that the attribute set is implicitly reduced to the non-sparse attributes, i.e., outliers are only identified based on anomalies of the non-sparse attributes.[19]

In summary, however, ALSO performs well also on datasets with several hundred attributes, and it is again noteworthy that the result quality is rather stable compared to other approaches. In contrast, RNN and 1-class SVM_e, which outperform ALSO on the

[19] More precisely: as M5' is used in its standard configuration, a minimum of four instances are required per leaf node. Thus, to form at least one leaf node with a majority of 1s (i.e., three out of four) in each of the folds, an attribute has to contain a minimum 30 instances with a 1 value in the optimistic case. This, however, is only the case for 130 of the 1558 attributes of the dataset.

Table 5 Outlier scores on the zoo dataset, using ALSO with M5' as a base learner, and largest addend according to Eq. (3)

Instance	Eggs	Backbone	Milk	Outlier score	Largest summands
Scorpion	0	0	0	1.498	Eggs (2.033), Backbone (2.346)
Platypus	1	1	1	1.377	Milk (2.114)
Seasnake	0	1	0	1.135	Eggs (2.033)

We depict all attributes that have a corresponding addend of at least 2, as well as the original attribute values of the respective attributes

Internet Advertisements dataset, achieve a performance of an AUC of only slightly above 0.5, or even below, on others.

5 Generating explanations of outliers

For computing the overall outlier score, ALSO first determines the weighted deviation from the expected value on each attribute. These deviations indicate which attributes contribute most to an instance's outlier score. If the dataset has been normalized using a standardization, an addend value larger than n means that the actual attribute value is more than $\sqrt{n}$ standard deviations away from the value that was predicted by the underlying model. Comparing the predicted to the actual values for those attributes, a first understanding why an instance has been marked as an outlier can be gained. Identifying the highest-scoring attributes for an outlier instance is a first step towards explaining outliers, like it is also done, e.g., by COP (Kriegel et al. 2012).

If a symbolic learning approach is used as a base learner, i.e., a learner creating human-interpretable models, the learned model for the respective attribute can be used to create an even more concise explanation why a certain instance has been marked as an outlier.

Since the datasets used for the quantitative evaluation in Sect. 4 have non-speaking attribute names and/or require extensive domain knowledge for understanding, we demonstrate the generation of explanations on the well-known *Zoo* dataset[20], which describes animals according to different features. Like in the experiments above, we normalize the features using standardization, and we use M5' as a base learner (learning regression trees). Using that approach, we found three instances that have an anomaly score larger than 1 (see Table 5), and identify the attributes with the largest summands for those instances.

Figure 8 depicts the corresponding M5' regression trees for the involved attributes. By following the paths for the three instances in the trees (marked in bold), explanations for the outliers can be derived as follows:

- The *scorpion* is an outlier because animals not giving milk are expected to lay eggs, and since non-toothed animals with a tail are expected to have a backbone (cf. Fig. 8a, b).
- The *platypus* is an outlier because animals laying eggs are not expected to give milk (cf. Fig. 8c).
- The *seasnake* is an outlier because animals not giving milk are expected to lay eggs (cf. Fig. 8a).

[20] https://archive.ics.uci.edu/ml/datasets/Zoo.

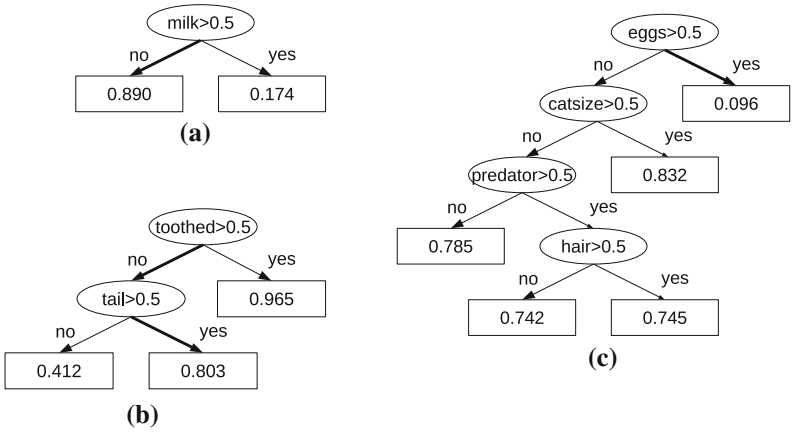


Fig. 8 Regression trees for the three attributes *eggs*, *backbone*, and *milk* in the zoo dataset, with the paths in the trees for the three outlier instances marked in bold. For illustrative purposes, the values used in the trees are in the original scale, not the standardized one. **a** Decision tree for the *eggs* attribute. The path in the tree for the *scorpion* and *seasnake* instance is marked in *bold*, **b** decision tree for the *backbone* attribute. The path in the tree for the *scorpion* instance is marked in *bold*, **c** decision tree for the *milk* attribute. The path in the tree for the *platypus* instance is marked in *bold*

Such explanations go beyond pointing out those attributes which contribute most to the outlier score, and are more concise and useful for analyzing the data. In fact, the explanations describe rare particularities about the animals identified as outliers (scorpions and seasnakes being among the few species giving live birth, but not feeding their offspring with milk, etc.).

6 Conclusion and outlook

In this paper, we have introduced a novel method of unsupervised outlier detection, which reformulates the problem of unsupervised outlier detection as a set of supervised learning problems. Our approach foresees the training of a predictive model for each attribute in the dataset, using the other attributes as features. In order to compute outlier scores, the algorithm compares the predicted attribute values to the original ones, using attribute weights that are learned on the fly based on the quality of the respective predictive models. These weights allow dealing with irrelevant attributes without losing the power to identify outliers. The ALSO approach can use any learning algorithm to build the predictive models and does not make any assumptions about the distribution of attributes within the dataset.

We have made experiments with different datasets to validate our approach. We have shown that ALSO yields good results, compared to a number of established approaches. The best results have been achieved with using regression tree learning (M5') and isotonic regression as base learners.

In addition to the quantitative results, ALSO is also capable of delivering *interpretations* for outliers. When using symbolic base learners, such as tree learners, those interpretations go beyond pointing out the attributes that have the largest contribution on the outlier score, which is the current state of the art for most outlier explanation approaches.

On the downside, ALSO requires training a number of single models for each attributes, which can become time-consuming, depending on the number of attributes. However, the approach is highly parallelizable by design (i.e., each of the predictive models can be trained

independently from the the others). Our implementation in RapidMiner is capable to parallelize the model training on a multi-core machine.

Other approaches for improving the runtime are also possible. For example, the weights of the attributes could be estimated based on a small dataset, including only those attributes with high weights in the computation of the outlier scores, which would reduce the number of predictive models to learn on the whole dataset. Furthermore, our approach in principle allows for treating the learning problem as a multi-target regression problem (Aho et al. 2009), which could help improving both the quality and the performance approach.

So far, we have used only datasets with numeric attributes (and regression learning for building the predictive models), but our approach is not limited to that. While a straight forward approach would be converting categorical attributes to numeric ones in a preprocessing step, that approach might not yield the best results. It might be more beneficial to use learning algorithms that are tailored to categorical attributes. However, for mixed datasets containing both categorical and numeric attributes, a suitable definition for a weighted distance function has to be defined first. As a direction for future work, we aim at exploring the possibilities of ALSO for such mixed datasets.

One limitation we have observed, in particular for high dimensional datasets, is the application of ALSO to sparse datasets. Here, the base learners often fail to learn a useful model, if the vast majority of the examples has 0 as a regression target. Future work will examine this limitation and ways to circumvent it, e.g., by preprocessing the data at hand, or using base learners tailored to that type of learning problem.

Another limitation of ALSO are datasets that contain a lot of missing values. While various strategies exist for dealing with missing values (ignoring instances with missing values, filling in attributes with a special code, or average values), analyzing how those strategies affect the performance of the ALSO approach will be an issue of future research.

While the experiments in this paper only consider batch outlier detection, approaches for online outlier detection (Pokrajac et al. 2007) and outlier detection in data streams (Elahi et al. 2008) have been proposed as well. The ALSO approach may also be extended to those classes of problems, e.g., by using incremental learning algorithms as base learners and/or applying windowing techniques. We aim at analyzing those capabilities in more detail in the future.

In summary, ALSO is a method of decomposing the outlier detection problem into a number of supervised learning problems. Given a suitable base learner, it has been shown to be a robust and flexible outlier detection method, which is tolerant to irrelevant attributes and reliable in yielding good results on a large variety of datasets. Furthermore, when used with symbolic learning algorithms, it can deliver concise explanations for outliers.

Acknowledgments The work presented in this paper was supported by RapidMiner in the course of the RapidMiner Academia program. The authors would like to thank all the anonymous reviewers, as well as Frederik Janssen from the Knowledge Engineering Group at TU Darmstadt, for their valuable feedback on previous versions of this paper. Moreover, the authors would like to thank Petar Ristoski on his advice and assistance in performing the statistical computations presented in this paper.

References

Abe, N., Zadrozny, B., & Langford, J. (2006). Outlier detection by active learning. In *Proceedings of the 12th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 504–509). ACM.

Achtert, E., Kriegel, H. P., & Zimek, A. (2008). ELKI: A software system for evaluation of subspace clustering algorithms. In *Scientific and statistical database management*. Lecture notes in computer science (Vol. 5069, pp. 580–585). Berlin, Heidelberg: Springer. doi:10.1007/978-3-540-69497-7_41.

Aggarwal, C. C. (2013). *Outlier analysis*. Berlin: Springer.

Aggarwal, C. C., Hinneburg, A., & Keim, D. A. (2001). On the surprising behavior of distance metrics in high dimensional space. In J. Van den Bussche & V. Vianu (Eds.), *Database theory—ICDT 2001*. Lecture notes in computer science (Vol. 1973, pp. 420–434). Berlin, Heidelberg: Springer.

Aggarwal, C. C., & Yu, P. S. (2001). Outlier detection for high dimensional data. *SIGMOD Record, 30*(2), 37–46. doi:10.1145/376284.375668.

Aho, T., Zenko, B., & Dzeroski, S. (2009). Rule ensembles for multi-target regression. In *ICDM* (pp. 21–30).

Amer, M., & Goldstein, M. (2012). Nearest-neighbor and clustering based anomaly detection algorithms for rapidminer. In *Proceedings of the 3rd RapidMiner community meeting and conference (RCOMM 2012)* (pp. 1–12).

Amer, M., Goldstein, M., & Abdennadher, S. (2013). Enhancing one-class support vector machines for unsupervised anomaly detection. In *Proceedings of the ACM SIGKDD workshop on outlier detection and description (ODD)* (pp. 8–15).

Angiulli, F., & Pizzuti, C. (2002). Fast outlier detection in high dimensional spaces. In T. Elomaa, H. Mannila, & H. Toivonen (Eds.), *Principles of data mining and knowledge discovery* (Vol. 2431, pp. 15–27). Berlin, Heidelberg: Springer. doi:10.1007/3-540-45681-3_2.

Barlow, R. E., Bartholomew, D. J., Bremner, J., & Brunk, H. D. (1972). *Statistical inference under order restrictions: The theory and application of isotonic regression*. New York: Wiley.

Barnett, V., & Lewis, T. (1994). *Outliers in statistical data*. New York: Wiley.

Breunig, M. M., Kriegel, H. P., Ng, R. T., & Sander, J. (2000). Lof: Identifying density-based local outliers. *ACM Sigmod Record*, *29*(2), 93–104.

Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. *ACM Computing Surveys (CSUR), 41*(3), 15.

Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research*, *7*, 1–30.

Elahi, M., Li, K., Nisar, W., Lv, X., & Wang, H. (2008). Efficient clustering-based outlier detection algorithm for dynamic data stream. In *Fifth international conference on fuzzy systems and knowledge discovery, FSKD '08* (Vol. 5, pp. 298–304). doi:10.1109/FSKD.2008.374.

Emmott, A. F., Das, S., Dietterich, T., Fern, A., & Wong, W. K. (2013). Systematic construction of anomaly detection benchmarks from real data. In *ACM SIGKDD workshop on outlier detection and description* (pp. 16–21).

Goldstein, M. (2014). Anomaly detection. In M. Hofmann & R. Klinkenberg (Eds.), *RapidMiner—Data mining use cases and business analytics applications* (pp. 409–436). CRC Press.

Hawkins, S., He, H., Williams, G., & Baxter, R. (2002). Outlier detection using replicator neural networks. In Y. Kambayashi, W. Winiwarter, & M. Arikawa (Eds.), *Data warehousing and knowledge discovery*. Lecture notes in computer science (pp. 170–180). Berlin, Heidelberg: Springer. doi:10.1007/3-540-46145-0_17.

He, Z., Xu, X., & Deng, S. (2003). Discovering cluster-based local outliers. *Pattern Recognition Letters*, *24*(9), 1641–1650.

He, Z., Xu, X., Huang, Z. J., & Deng, S. (2005). Fp-outlier: Frequent pattern based outlier detection. *Computer Science and Information Systems/ComSIS*, *2*(1), 103–118.

Hodge, V. J., & Austin, J. (2004). A survey of outlier detection methodologies. *Artificial Intelligence Review*, *22*(2), 85–126.

Knorr, E. M., & Ng, R. T. (1999). Finding intensional knowledge of distance-based outliers. In *Proceedings of the 25th international conference on very large data bases, VLDB '99* (Vol. 99, pp. 211–222). San Francisco, CA: Morgan Kaufmann Publishers Inc.

Kriegel, H. P., Kröger, P., Schubert, E., & Zimek, A. (2009). Loop: Local outlier probabilities. In *Proceedings of the 18th ACM conference on information and knowledge management* (pp. 1649–1652). ACM.

Kriegel, H. P., Kröger, P., Schubert, E., & Zimek, A. (2012). Outlier detection in arbitrarily oriented subspaces. In *2012 IEEE 12th international conference on data mining, ICDM '12* (pp. 379–388). IEEE.

Liu, F. T., Ting, K. M., & Zhou, Z. H. (2012). Isolation-based anomaly detection. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, *6*(1), 3.

Mejía-Lavalle, M., & Sánchez Vivar, A. (2009). Outlier detection with explanation facility. In *Machine learning and data mining in pattern recognition*. Lecture notes in computer science (Vol. 5632, pp. 454–464). Berlin, Heidelberg: Springer. doi:10.1007/978-3-642-03070-3_34.

Müller, E., Schiffer, M., Gerwert, P., Hannen, M., Jansen, T., & Seidl, T. (2010). Sorex: Subspace outlier ranking exploration toolkit. In J. Balcázar, F. Bonchi, A. Gionis, & M. Sebag (Eds.), *Machine learning and*

knowledge discovery in databases. Lecture notes in computer science (pp. 607–610). Berlin, Heidelberg: Springer. doi:10.1007/978-3-642-15939-8_44.

Nemenyi, P. (1962). Distribution-free multiple comparisons. *Biometrics*, *18*(2), 263.

Padmanabhan, B., & Tuzhilin, A. (2000). Small is beautiful: Discovering the minimal set of unexpected patterns. In *Proceedings of the sixth ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 54–63). ACM.

Pelleg, D., Moore, A. W. (2000). X-means: Extending K-means with efficient estimation of the number of clusters. In *Proceedings of the seventeenth international conference on machine learning, ICML '00* (pp. 727–734). San Francisco, CA: Morgan Kaufmann Publishers Inc.

Pokrajac, D., Lazarevic, A., & Latecki, L. J. (2007). Incremental local outlier detection for data streams. In *IEEE symposium on computational intelligence and data mining, CIDM '07* (pp. 504–515). IEEE.

Quinlan, J. R., et al. (1992). Learning with continuous classes. In *Proceedings of the 5th Australian joint conference on artificial intelligence* (Vol. 92, pp. 343–348).

Skillings, J. H., & Mack, G. A. (1981). On the use of a friedman-type statistic in balanced and unbalanced block designs. *Technometrics*, *23*(2), 171–177.

Teng, C. M. (1999). Correcting noisy data. In *Proceedings of the sixteenth international conference on machine learning, ICML '99* (pp. 239–248). San Francisco, CA: Morgan Kaufmann Publishers Inc.

Wagstaff, K. L., Lanza, N. L., Thompson, D. R., Dietterich, T. G., & Gilmore, M. S. (2013). Guiding scientific discovery with explanations using demud. In *AAAI conference on artificial intelligence*. http://www.aaai.org/ocs/index.php/AAAI/AAAI13/paper/view/6171.

Xu, L., Crammer, K., & Schuurmans, D. (2006). Robust support vector machine training via convex outlier ablation. In *Proceedings of the 21st national conference on artificial intelligence, AAAI '06* (Vol. 1, pp. 536–542). Boston, MA: AAAI Press. http://dl.acm.org/citation.cfm?id=1597538.1597625.

Yamanishi, K., & Takeuchi, J.i. (2001). Discovering outlier filtering rules from unlabeled data: Combining a supervised learner with an unsupervised learner. In *7th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 389–394). ACM.

Zimek, A., Gaudet, M., Campello, R. J., & Sander, J. (2013). Subsampling for efficient and effective unsupervised outlier detection ensembles. In *19th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 428–436). ACM.