Subsampling for Efficient and Effective Unsupervised Outlier Detection Ensembles

Arthur Zimek[*], Matthew Gaudet, Ricardo J. G. B. Campello[†], Jörg Sander
Department of Computing Science, University of Alberta, Edmonton, AB, Canada
{zimek,mgaudet,rcampell,jsander}@ualberta.ca

ABSTRACT

Outlier detection and ensemble learning are well established research directions in data mining yet the application of ensemble techniques to outlier detection has been rarely studied. Here, we propose and study subsampling as a technique to induce diversity among individual outlier detectors. We show analytically and experimentally that an outlier detector based on a subsample per se, besides inducing diversity, can, under certain conditions, already improve upon the results of the same outlier detector on the complete dataset. Building an ensemble on top of several subsamples is further improving the results. While in the literature so far the intuition that ensembles improve over single outlier detectors has just been transferred from the classification literature, here we also justify analytically why ensembles are also expected to work in the unsupervised area of outlier detection. As a side effect, running an ensemble of several outlier detectors on subsamples of the dataset is more efficient than ensembles based on other means of introducing diversity and, depending on the sample rate and the size of the ensemble, can be even more efficient than just the single outlier detector on the complete data.

Categories and Subject Descriptors

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

Keywords

outlier detection; ensemble

1. INTRODUCTION

An outlier is *"an observation (or subset of observations) which appears to be inconsistent with the remainder of that set of data"* [6]. Detecting outliers is an important task in many practical applications. Some applications of outlier detection, such as detecting measurement errors, are mostly concerned with removing the outliers from the data as a form of "noise". Other applications, such as credit card abuse detection, or the identification of unusual measurements in scientific data, are concerned with finding outliers because their deviating behavior from the rest of the data may require specific actions or provide opportunities for new insights.

Various approaches to outlier detection have been proposed, based on different notions of outliers, or targeted towards specific applications that require the identification of outliers. Here, we are interested in *unsupervised*, non-parametric outlier detection methods that assign a *score* to each data object and thus allow a ranking of objects according to their degree of outlierness.

Parametric, statistical approaches [6, 35] fit certain distributions to the data by estimating the parameters of these distributions from the given data. A problem with these approaches is that distribution parameters such as mean, standard deviation, and covariances are rather sensitive to the presence of outliers. Possible effects of outliers on the parameter estimation have been termed *"masking"* and *"swamping"*. Outliers can *mask* their own presence by influencing the values of the distribution parameters (resulting in false negatives), or *swamp* inliers to appear as outlying due to the influenced parameters (resulting in false positives) [6, 19].

Non-parametric approaches do not assume a specific distribution of the data, but estimate (explicitly or implicitly) certain aspects of the probability density. Non-parametric methods include the well-known "distance-based" and "density-based" methods. Both distance-based and density-based methods basically aim at providing a rather simple estimate of the density around points, which can be seen as an approximation of statistical kernel density estimates. Distance-based methods such as DB-outlier [25] and its variants are based on the k nearest neighbor (kNN) distances [5, 34], trying to find so-called *global* outliers as points that are, roughly speaking, far away from the rest of the data. Density-based methods such as LOF [10] and its variants try to find so-called *local* outliers as points that are, roughly speaking, located in an area of relative low density compared to their kNN (intended to indicate points that are outliers with respect to the nearest mode in the data distribution). The density around points in these methods is also estimated based on kNN distances. One problem with distance-based and density-based methods is that they can

[*] This work was done while the author was on leave of absence from Ludwig-Maximilians-Universität München, Germany.

[†] This work was done while the author was on sabbatical leave from University of São Paulo, São Carlos, Brazil.

KDD'13, August 11–14, 2013, Chicago, Illinois, USA.
Copyright 2013 ACM 978-1-4503-2174-7/13/08 ...$15.00.

also suffer from effects similar to *masking* and *swamping*, due to the simplicity of (and thus error in) the density estimates. Another problem is the typically high runtime of these approaches, due to the fact that their computation includes at least finding the kNN of each data point (resulting in an at least quadratic complexity w.r.t. the database size). In this paper, we address both problems of distance-based and density-based methods. We propose and study a general approach to improve both the quality and the performance of such outlier detection methods by combining into an ensemble results of a base method on subsamples of the data. Previous work on outlier ensembles is very limited and only shows empirically that ensembles of outlier detectors have the potential to improve the quality, compared to that of their base methods [30, 36], at an increased runtime cost.

Our work is novel and advances the area of outlier detection in the following respects:

- We argue theoretically and demonstrate empirically that it is possible to construct ensemble members for outlier detection methods which perform individually already better than the base method, in general.
- Combining those outlier detectors into an ensemble renders the performance gain not only more robust but can improve the performance even further.
- At the same time, when using small sample sizes for the ensemble members, we can gain considerable speed-up in runtime compared to running a standard ensemble and, for small ensemble sizes, even compared to running the base method on the whole data set.
- The proposed principle is fundamental and flexible. It does not rely on specific data types. It can be combined with various conventional outlier detection techniques.

The rest of the paper is organized as follows: We discuss related work on outlier detection and ensembles for outlier detection (Section 2). We provide theoretical reasoning to support outlier detection ensembles in general and the claimed properties of our method in particular (Section 3). We provide experimental results to support our claims empirically (Section 4). We conclude the paper in Section 5.

2. RELATED WORK

The distance-based notion of outliers (DB-outlier) [25] was the first database-oriented approach in the area of unsupervised outlier detection, which initiated a new line of research on this topic in the data mining community. Variants of DB-outliers consider the distances to the k nearest neighbors of each object and use these distances to rank the objects [34], or, they use the sum of distances to all points within the set of kNN (called the "weight") as an outlier degree [5]. These methods are also called *global* methods in that the computed outlier scores represent global density scores for each point. The so-called *local* methods, e.g. LOF [10], consider instead local density scores, which are *ratios* between the density around an object and the density around its neighboring objects. Variants of the local outlier model include LoOP [27], and LOCI [33]. Also the distance-based method LDOF [44] is related in reasoning about local comparisons. It has been shown recently [37], however, that the differentiation between *global* and *local* methods is not strictly dichotomous but that there are degrees of *locality*.

Much research has aimed at improving the efficiency of unsupervised outlier detection by algorithmic techniques, for example based on approximations or improved pruning techniques for mining the top-n outliers [4, 7, 22, 23, 26, 42]. An analysis of such efficiency improving techniques for outlier detection algorithms has been provided by Orair et al. [32]. These techniques, however, do not aim at improving the approximations of the underlying statistical notion of outlierness. They only approximate a specific algorithmic model.

Ensemble techniques, on the other hand, have the potential to improve the performance of their components in terms of the quality of the detected outliers, rather than in terms of runtime (but we will show in this paper that it is even possible to gain performance improvements when constructing certain types of outlier ensembles). The first approach to improve outlier detection by ensemble techniques, based on "feature bagging", was proposed by Lazarevic and Kumar [30], combining different results of the same algorithm (namely LOF [10]) applied to different, randomly selected feature subsets. Feature bagging is a common procedure to induce diversity of ensemble members in ensemble classification [11] or ensemble clustering [8, 14, 40].

Subsequent research on outlier detection ensembles focused on the issue of comparability of scores for score combinations, using Sigmoid functions and mixture modeling to fit outlier scores, provided by different detectors, into comparable probability values [17], or scaling by standard deviation [31], or statistical reasoning about score distributions [28], enabling the combination of different outlier detection methods into one ensemble. Schubert et al. [36] proposed a similarity measure to appropriately compare different outlier rankings (based on scores) and to allow for the assessment of the diversity of different outlier detectors. As an application, they propose a greedy ensemble approach, demonstrating the importance of diversity for the performance of an ensemble. In all these papers, although outlier detection ensembles have been discussed and improved, no new method of inducing diversity has been pursued.

Except for feature bagging [30], all other existing ensemble methods for outlier detection [17, 28, 31, 36] are meta-methods and could be used on top of our sample-based method (or on top of feature bagging, as in [28,31,36]). They do not propose original means to induce diversity when using a selected base outlier detection method.

In general, while the motivation for ensemble methods for outlier detection is borrowed from the rich tradition in the literature on supervised ensemble learning [11,12,21,41], the theoretical foundation for ensemble learning in the unsupervised setting is far less mature. The same holds true not only for outlier detection ensembles but also for clustering ensembles despite the far more abundant literature on practical approaches in that area [18]. Although the problem setting is considerably different, let us finally note that sampling has been used in ensemble clustering to induce diversity. Different subsamples of the data set have been clustered and the resulting clusterings were combined into a consensus clustering [13, 16, 20, 39].

3. OUTLIER DETECTION ENSEMBLES BASED ON SUBAMPLING

In this section, we will discuss the potential benefits of using outlier detection ensembles based on subsampling.

Previous approaches using ensemble learning for outlier detection [17, 28, 30, 31, 36] transferred techniques without any theoretical foundation of why, what has a clear theoretical background in supervised learning, should also work in unsupervised outlier detection. Such a view can be loosely argued for when we consider outlier detection methods as "classifiers". When assuming that a threshold on outlier scores is used to distinguish between outliers and inliers, we can view the outlier method as classifying all objects into one of these two classes: outliers and inliers – even though, no labels are used in the "training" phase when the model (ranking) is built. If we succeed to construct diverse enough outlier detectors for the same data set, we can hope to improve the overall performance over the individual members by combining them into an ensemble. The "generic" argument given is that all the ensemble members are committing errors but on different cases, if the members are independent, i.e., diverse, or, in other words, if the errors are uncorrelated. While such a "generic" view may potentially explain some of the performance gains, we will show in the following subsections that there are more specific reasons for why (under some general assumptions) an ensemble of outlier detection methods can improve the performance over its individual members.

3.1 Benefits of Ensembles for Outlier Detection Based on Density Estimates

In this paper, we are focusing on distance-based and density-based outlier detection methods, which, as discussed in the introduction, compute outlier scores that are based, implicitly or explicitly, on some form of density estimates. One can view these methods as trying to identify the outliers in a given data set X with respect to an unknown probability density f, which represents the process that has "generated" the majority of the data set (at least the inliers). The data set X itself can be viewed as a sample drawn from the true, but unknown underlying density distribution, and the methods try to estimate the density $f(x)$ around points x using a more or less "rough" density estimate $\hat{f}_X(x)$ (in order to compute outlier scores in some way).

Assuming the correctness of the underlying outlier model of the methods, it is clear that the quality of a method's result depends on the quality of the density estimate $\hat{f}_X(x)$ and that the results will improve if the estimate can be improved. For this case, we can show formally that a diverse ensemble of such outlier detectors does in fact show an improved expected performance over the individual ensemble members, under some general conditions.

Given a true, smooth p.d.f. $f(x)$ and a data set X, we can express and estimate $\hat{f}_X(x)$ of $f(x)$ based on X as:

$$\hat{f}_X(x) = f(x) + v_X(x)$$

where $v_X(x)$ is a random variable describing the error of the estimate due to the finite sample.

The quality of the estimate $\hat{f}$ of f decides over success and failure of the outlier detection. However, the density estimates used by the considered outlier detection algorithms may not be reliable and stable in all regions of the data space, due to the natural intrinsic randomness associated with a single sample that the data set represents. If we are able to obtain multiple density estimates for each point x (e.g., as we propose via subsamples), we can obtain more reliable and stable density estimates by averaging the multiple density estimates for each point. The rationale for this is the following: The output of outlier methods is a ranking of all points x in terms of outlier scores that, in essence, depends on the ranking of the points according to $\hat{f}_X(x)$. Ideally, we want a ranking of the points x according to $f(x)$. If we have multiple density estimates for each point that we average, we can consider the estimate itself as a random variable and averaging[1] these estimates for each point gives us the expectation of this variable as:

$$\begin{aligned} E\{\hat{f}_X(x)\} &= E\{f(x)\} + E\{v_X(x)\} \\ &= f(x) + E\{v_X(x)\} \end{aligned}$$

In this formulation, one can clearly see that the ranking of objects w.r.t. $E\{\hat{f}_X(x)\}$ is the same as the ranking w.r.t. the true density $f(x)$ (the "ideal ranking"), if just the *expectation* of the error $v_X(x)$ in the individual estimates is the same for every point x. This is obviously the case when the random variable that describes the error would not depend on x, in which case $E\{v_X(x)\} = E\{v_X\} = \mu_{v_X}$, but one would also obtain the "ideal" ranking when the error is *not* independent on x; for instance, when the error would vary between points but the *expectation* is the same for each point, we would also have the same ranking. We can even obtain the same ranking as the "ideal" ranking if the expectations $E\{v_X(x_1)\}$ and $E\{v_X(x_2)\}$ differ for two points x_1 and x_2, as long as the difference does not cause an inversion between the actual ranks $E\{\hat{f}_X(x_1)\}$ and $E\{\hat{f}_X(x_2)\}$, respectively. Furthermore, if we consider that for successful outlier detection, the methods only have to *distinguish* between outliers and inliers, we can even allow inversions between ranks, as long as rank inversions occur only within outliers or within inliers. Only a rank inversion between an outlier and an inlier would be problematic. In the next subsection, we will argue that for the proposed ensemble technique using subsamples, the expectation of the error in the density estimate $E\{v_X(x)\}$ does depend on the location x and its surrounding density, but that the method has the desirable property that it can increase the "gap" in ranks between the outliers and the inliers, making inversions in rank between these groups of points even less likely.

3.2 Additional Benefits of Subsampling

Subsampling is theoretically well suited to introduce diversity into an ensemble of otherwise identical distance-based or density-based outlier detection methods. Every member of the ensemble will determine the outlier score of every object in the database, but only using a small subset of the data to estimate the density around points. Learning density estimates for outlier detection on smaller samples can actually improve the detection rate of outliers, compared to learning these estimates on the whole data set that conceptually represents just a somewhat larger sample of an unknown distribution f. We will see in the empirical evaluation that in practice, surprisingly small sample sizes (such as 20% or in many cases even just 10%) are typically not leading to a deteriorated but to a considerably improved quality of the outlier detection for a sample-based ensemble of outlier detectors. One reason for the improved performance of an ensemble is, as expected, just the combination of the results of multiple outlier detectors. Compared to using the dataset as the only sample drawn from f, drawing multiple subsamples X from this sample can minimize the effect of the randomness associated with a single sample.

[1] Note that averaging the scores to build an ensemble has been, heuristically, common practice [17, 28, 30, 31, 36], but now it finds also a theoretical justification.

Another, more interesting reason for the improved performance is that the base method applied to a smaller subsample of a given data often shows an improved outlier detection rate, compared to the same method applied to the whole data set. As we will argue formally in the following, this is due to the fact that distance-based and density-based methods are essentially using simple (not volume normalized) k nearest neighbor distances to estimate density.

To understand the effects of sample based k nearest neighbor distances, consider a sphere of radius r in a d-dimensional Euclidean space, containing n data points uniformly distributed within the sphere. The expected Euclidean distance from a point to its k nearest neighbour (kNN) is given by [9]:

$$E\{d_k\} = r\left(\frac{k}{n}\right)^{\frac{1}{d}} \tag{1}$$

For a given data set, let r be a constant value small enough so that, for two spheres having the same radius r but lying on different positions of the data space, the data points within both spheres are approximately uniformly distributed. Now, suppose that the number of data points within each of these spheres is different, given by n_1 and n_2 ($n_1 \neq n_2$), which means that the densities of the data in the respective regions of the space are different (as their volumes are the same). For example, one sphere might be located inside a dense cluster, whereas the other one might lie on a sparse area containing background noise. Then, it follows from (1) that the expected kNN distances in the corresponding regions of the space are given by:

$$E\{d_k^{'}\} = r\left(\frac{k}{n_1}\right)^{\frac{1}{d}}; \qquad E\{d_k^{''}\} = r\left(\frac{k}{n_2}\right)^{\frac{1}{d}} \tag{2}$$

If one randomly removes a fraction $1 - m$ of the data objects with equal probability, the expected number of remaining objects within those two spheres are given by $n_1 m$ and $n_2 m$, respectively. In this case, the expected kNN distances become:

$$E\{d_k^{'}\} = r\left(\frac{k}{n_1 m}\right)^{\frac{1}{d}}; \qquad E\{d_k^{''}\} = r\left(\frac{k}{n_2 m}\right)^{\frac{1}{d}} \tag{3}$$

The difference in the expected distances are therefore:

$$\Delta_1 = r\left(\frac{k}{n_1 m}\right)^{\frac{1}{d}} - r\left(\frac{k}{n_1}\right)^{\frac{1}{d}} = r\left(\frac{k}{n_1}\right)^{\frac{1}{d}}\left(\frac{1 - m^{\frac{1}{d}}}{m^{\frac{1}{d}}}\right) \tag{4}$$

$$\Delta_2 = r\left(\frac{k}{n_2 m}\right)^{\frac{1}{d}} - r\left(\frac{k}{n_2}\right)^{\frac{1}{d}} = r\left(\frac{k}{n_2}\right)^{\frac{1}{d}}\left(\frac{1 - m^{\frac{1}{d}}}{m^{\frac{1}{d}}}\right) \tag{5}$$

In *relative terms*, if we divide Δ_1 and Δ_2 by the original expected distances (for the full dataset, i.e., before the subsampling), we get:

$$\frac{\Delta_1}{r\left(\frac{k}{n_1}\right)^{\frac{1}{d}}} = \frac{\Delta_2}{r\left(\frac{k}{n_2}\right)^{\frac{1}{d}}} = \left(\frac{1 - m^{\frac{1}{d}}}{m^{\frac{1}{d}}}\right) \tag{6}$$

The result in (6) says that the expected kNN distances within the spheres increase proportionally as a function of the subsampling rate m. This result reflects the intuition that, in relative terms, the contrast between the densities of the spheres is kept constant, which justifies the use of a subsampling procedure with even sampling probabilities. In an ensemble setting, for instance, this means that one can get multiple (sub)samples that exhibit variability (diversity) in terms of their observations, but keep the same expected density profile as the full dataset.

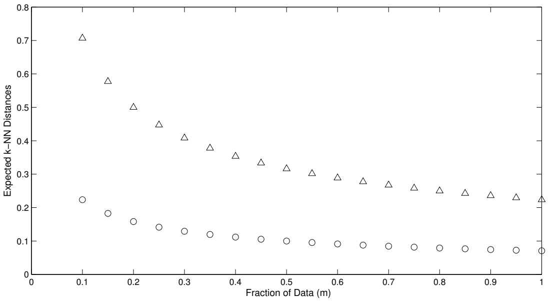


Figure 1: Behaviour of the expected 5-NN distances for two spheres with radius $r = 1$, in a 2D Euclidean space, containing $1000m$ (circles) and $100m$ (triangles) objects uniformly distributed (m is a fraction of the data).

The above result is important but it does not explain all implications of subsampling when using unnormalized k nearest neighbor distances. In *absolute terms*, Equations (4) and (5) tell us that the expected difference in the kNN distances will be greater for a less dense sphere, i.e., $\Delta_1 > \Delta_2$ if $n_1 < n_2$. This means that the expected kNN distances "diverge" in absolute terms when the data are downsampled to a fraction m of their original size. In other words, the absolute differences between the expected kNN distances in areas of different densities tend to increase as a function of the subsampling rate. This effect is illustrated in Figure 1 for $r = 1$, $d = 2$, $k = 5$, $n_1 = 100$, $n_2 = 1000$, and m ranging from 0.1 to 1.

Such an effect can be beneficial for outlier detection, since it can make it easier to distinguish between outliers and inliers. Particularly when also using an ensemble as discussed above, the gap in the ranks between outliers and inliers can increase, making inversion of ranks between these two groups less likely.

3.3 Method and Complexity

Note that the implementation of our proposal is not as simple as to take subsamples and then run the outlier detection algorithms on these subsamples. This way we would very likely completely miss information on the outlierness of many objects that are not contained in any subsample, and many objects would get scores only from some of the subsamples. Instead, for each ensemble member, we draw a subsample from the database and compute the neighborhood of each object in the database based on the subsample. This way, using subsample-based ensembles can also lead to a considerable speed-up, compared to other types of ensembles and, for small subsamples and ensemble sizes, even compared to running the base method on the whole data set. We will demonstrate in the experimental evaluation that sample sizes small enough to achieve substantial runtime improvements are good choices in practice, leading to good outlier detection rates. In this subsection, we show the expected runtime improvements by studying the theoretical complexities.

While other ensemble methods require a multiple of the computing time compared to the base learner, the theoretical behaviour of a subsample based ensemble is faster (and requires less resources) than other types of ensembles. The typical complexity of a base method is $\mathcal{O}(n^2)$, due to the required kNN queries over a database of n objects. The runtime of a "standard ensemble" such as feature bagging is essentially s times the runtime of the base method, where s is a factor that is determined by the number of base learners used in the ensemble (i.e., the size of the ensemble). This factor is reduced in the case of feature bagging. Using only a subset of the dimensions makes individual distance computations faster by some constant factor.

For sample based ensembles, on the other hand, the complete ensemble can even be faster than the base method on the complete dataset, because of the quadratic runtime in n of the base method. While the base method requires kNN queries for each object on the complete database (hence $\mathcal{O}(n^2)$), using a subsample of size $m \cdot n$, $0 < m < 1$, reduces this to $\mathcal{O}(n^2 \cdot m)$. The runtime of a sample based ensemble is essentially s times the runtime of the base method, using a much smaller data set for the neighborhood computation.

For an ensemble size of 10 base learners and sample size of 10%, the sample-based ensemble would require roughly the same runtime than a single base method on the full dataset but 10 times less time than an ensemble with the same number s of ensemble members based on other means of diversity. For larger ensembles, the ensemble requires only a small multiple of the base method but still only 10% (or the equivalent of the sample size m) of a standard ensemble. For example, if we use 25 ensemble members and sample size 10%, the ensemble will require roughly 2.5 times the runtime of the base method.

4. EVALUATION

4.1 Methods and Parameters

For the reasons discussed in Section 2, the canonical competitor is feature bagging (FB) [30]. As base methods we use LOF [10], LDOF [44], and LoOP [27].

For the setup of experiments, we have to consider various parameters. For both ensemble methods (feature bagging and subsampling), we choose a fixed number of 25 ensemble members. We follow the original setup of the feature bagging method, combining the scores of the ensemble members by computing the average. For the subsampling, we consider various sample sizes. Each of the base methods requires a size k of the neighborhood. Hence we will show experimental results (i) with a fixed choice of k and varying sample size; (ii) with a fixed sample size, varying k; and (iii) with fixed choices of k and sample size, comparing different base methods. When we fix k, we choose a value that gives a reasonable result quality (i.e., better than random) for the *base method* and compare that to the ensemble variants. Finally (iv), for the synthetic dataset collections, where the individual datasets follow the same general characteristics, we show an average behaviour over all datasets of the collection.

We report the area under the receiver operating characteristic curve (ROC AUC), which plots the true positive rate vs. the false positive rate, a common measure for evaluation of outlier detection methods [17, 28, 30, 31, 36]. The experiments are performed using ELKI [2, 3].

4.2 Datasets

For a statistical assessment, we generate two independent sets of 30 synthetic datasets (batch1 and batch2). For each dataset, we choose randomly values for the following parameters in the given range: dimensionality $d \in [20, \ldots, 40]$, number of clusters $c \in [2, \ldots, 10]$, for each cluster independently the number of points $n_{c_i} \in [600, \ldots, 1000]$. For each cluster, the points are generated following a Gaussian model as follows: For each cluster c_i, and each attribute a, we choose a mean $\mu_{c_i,a}$ from a uniform distribution in $[-10, 10]$ and a standard deviation $\sigma_{c_i,a}$ from a uniform distribution in $[0.1, 1]$. Then for the cluster c_i, n_{c_i} cluster objects (points) are generated attribute-wise by the Gaussians $\mathcal{N}(\mu_{c_i,a}, \sigma_{c_i,a})$. The resulting cluster is rotated by a series of random rotations and the covariance matrix Σ corresponding to the theoretical model is computed by the corresponding matrix operations [38]. Then, we compute for each point the Mahalanobis distance to its corresponding cluster center, using the covariance matrix Σ of the cluster. For a dataset dimensionality d, the Mahalanobis distances for each cluster follow a χ^2 distribution with d degrees of freedom. We label as outliers those points that exhibit a distance to their cluster center larger than the theoretical 0.975 quantile, independently of the actually occurring Mahalanobis distances of the sampled points. This results in an expected amount of 2.5% outliers per dataset.

As real datasets we use the datasets Satimage, Lymphography, and Segment (used also by Lazarevic and Kumar [30]). Additionally, we chose from the UCI machine learning repository [15]: Wisconsin breast cancer (WBC) and Waveform Database Generator (waveform). While Lazarevic and Kumar consider outlier detection as equivalent to rare class detection, we argue that outliers are bound to be rare, but objects of a rare class are not necessarily outliers. Therefore, we use a different preprocessing for some of the datasets: For Satimage, we combined train and test set and transformed the dataset to an outlier task by taking a sample of 10% from class 2, evaluating the downsampled class as outliers vs. the rest.[2] For Lymphography, we merged the small classes 1&4 as outliers vs. the rest. For Segment, we chose classes GRASS, PATH, and SKY for downsampling, in turn, to 10%, which renders the remaining objects of these classes outliers (resulting in three different datasets). For the datasets WBC and waveform we also select a meaningful outlier class for downsampling ('malignant', and '0', respectively). With this method of using classification data for evaluation of outlier detection methods we are conform with the literature [1, 24, 29, 43, 44].

Overall, this results in 60 synthetic and 7 real data sets.

4.3 Efficiency

For a fair comparison, we use a preprocessing of the neighborhood computation for all methods on equal terms, as facilitated by the framework ELKI [2]. As in our experiments we use 25 ensemble members, we study the runtime of a typical base method (LOF), the subsampling ensemble (10% sample size) and feature bagging, when scaling the number of objects in the database. As demonstrated in Figure 2,

[2] Lazarevic and Kumar used the smallest class 4 as outlier vs. rest, but this is an example where the rare class does not constitute outliers, as the classes 3-7 are all very similar. Accordingly, they report performance very close to a random result on this dataset.

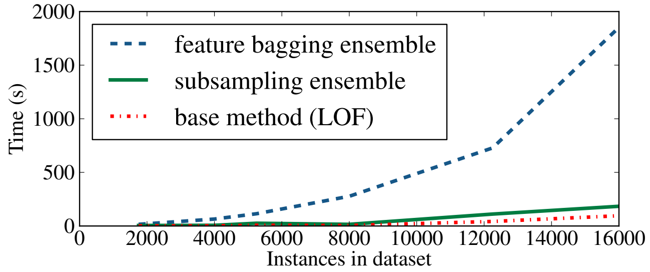


Figure 2: Runtime of LOF, subsampling ensemble, and feature bagging when increasing database size.

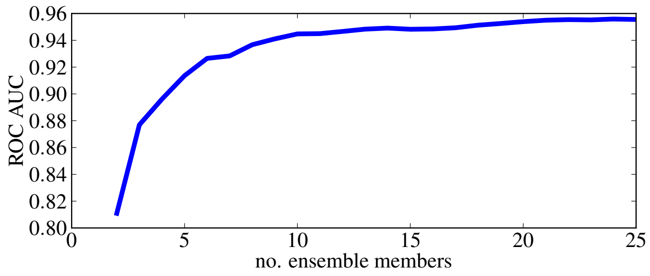


Figure 3: Quality with increasing ensemble size.

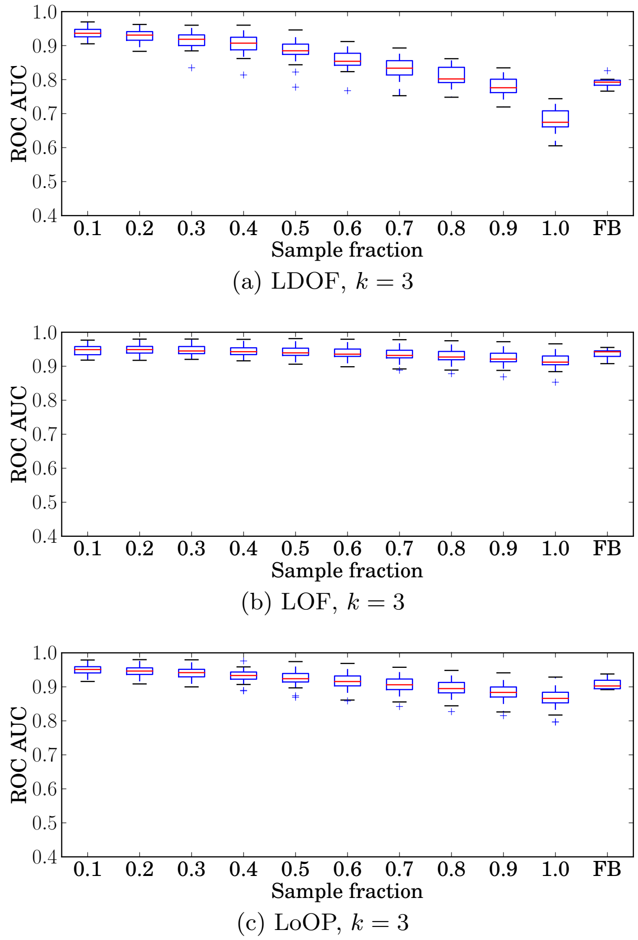
the subsampling ensemble is close to the base method while feature bagging requires a multiple of the runtime.

As discussed in Section 3.3, the efficiency depends on the sample size and on the ensemble size. We do not evaluate the ensemble size further, let us just consider an example on one of the synthetic datasets to study the behaviour with adding more ensemble members (Figure 3). We see a strong increase in quality between 2 and 10 ensemble members, then, up to 25 ensemble members, the quality increases further, steadily but slowly. This improved performance comes at moderate runtime cost. Nevertheless, we fix the ensemble size to 25 in the following experiments.

4.4 Effectiveness

For illustration of results with variances we use box plots where the box extends from the lower to upper quartile values of the data, with a line at the median. The whiskers extend from the box to show the range of the data. The length of the whiskers extend to the most extreme data point within 1.5*(75%-25%) data range. Occasionally occurring single data points beyond that range are plotted as flier points past the end of the whiskers. Note however that the source of variance in the plots will differ: in synthetic data, we give the distribution over the 30 datasets, in real data, we give the distribution over the individual ensemble members.

Synthetic Data. First, we show as a statistical assessment the results of the subsample-based ensemble over all the synthetic datasets of batch1. Here the box plots visualize the distribution of the results for the same sample size, the same base method, and the same parametrization of the base method for all datasets in the batch for the subsampling ensemble, the base method (sample size 1.0), and the feature bagging ensemble (FB). Figure 4 shows examples for a fixed $k = 3$ for the base methods LDOF, LOF, and LoOP. The behaviour on batch2 (not shown) follows the same general pattern. We varied k from 2 to 10 and got similar results. The smaller sample size leads to larger improvements.



(a) LDOF, $k = 3$

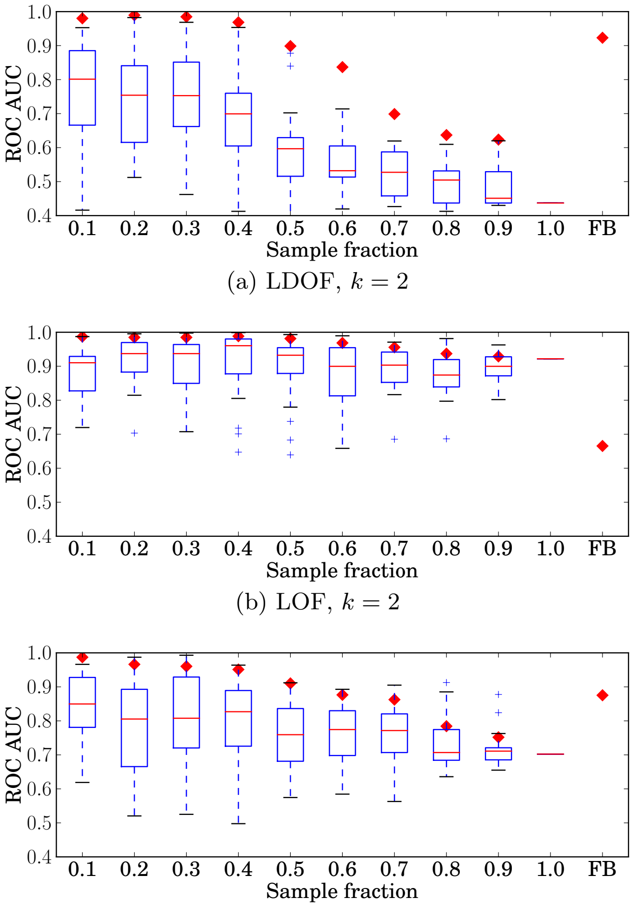
(b) LOF, $k = 3$

(c) LoOP, $k = 3$

Figure 4: ROC AUC for ensembles—different sample sizes as well as feature bagging (FB)—and base method (sample size=1.0), on the 30 datasets of batch1.

Real Data. Having shown the ensemble performances over a set of 30 datasets for the synthetic data, we now analyze the behaviour on individual real datasets. Here, we show in the whisker plots the variance in the ROC AUC achieved by the individual ensemble members based on subsamples of different sample size (zero variance for sample size 1.0, which reflects the performance of the deterministic base method on the complete data), and feature bagging (FB). The ROC AUC of the ensembles (subsampling and feature bagging) are visualized by a diamond.

Figures 5, 6, and 7 show the results for the three base methods on the datasets Lymphography, WBC, and Satimage-2, respectively. We choose the same k for all base methods such that at least some of the base methods get reasonable results. For the larger dataset satimage-2, the k needs to be larger as well. Comparing these plots, we see a different behaviour of the base methods as some datasets are easy for some base methods while some other datasets are relatively hard. In particular, LDOF does not retrieve sensible results on all three datasets. In all cases, however, the subsampling ensemble improves. Feature bagging does

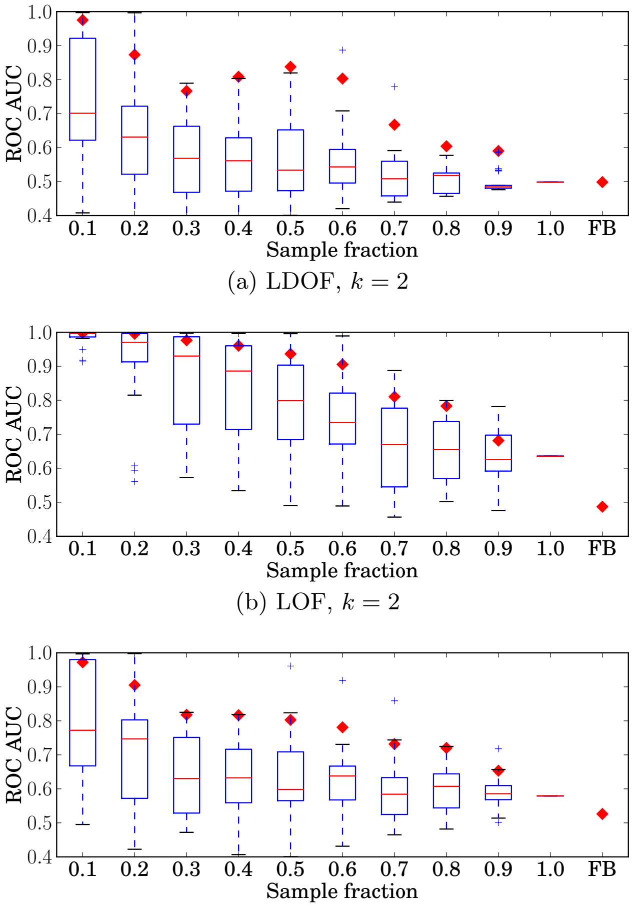


(a) LDOF, $k = 2$

(b) LOF, $k = 2$

(c) LoOP, $k = 2$

Figure 5: ROC AUC for ensemble members of the subsampling ensemble for different sample sizes (boxes), the base method (sample size=1.0), and ensembles (diamonds)—on top of subsamples and feature bags (FB)—on dataset Lymphography.



(a) LDOF, $k = 2$

(b) LOF, $k = 2$

(c) LoOP, $k = 2$

Figure 6: ROC AUC for ensemble members of the subsampling ensemble for different sample sizes (boxes), the base method (sample size=1.0), and ensembles (diamonds)—on top of subsamples and feature bags (FB)—on dataset WBC.

not perform always that convincingly, in some cases it drops to (or below) random quality. Only for LDOF and LoOP on Lymphography (Figures 5(a), 5(c)), feature bagging can recover from the weak performance of the base learner.

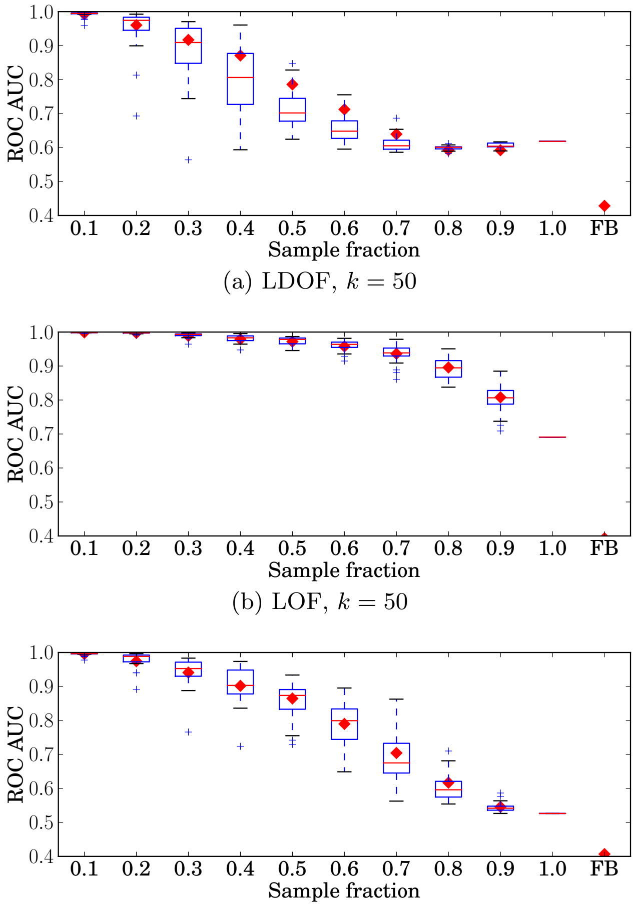
As a general picture from these and other results, we see that the smaller sample size actually has the larger potential of improvement. Although the smaller sample keeps not as much information about the dataset (and the unknown underlying density-distribution), from the point of view of ensemble learning, these findings make sense, as the smaller samples will actually provide the most diverse ensemble members, and it also shows the practical applicability of the reasoning we provided in Section 3.2. In most cases, we find the 10%-sample to work best. However, the break-even point between too much loss of information and too high similarity of ensemble members differs from dataset to dataset. We have also examples where the 10%-sample is already too small such as in Figure 5(a). That is possibly related to the fact that the lymphography data are relatively small.

However, we fix the sample size to 0.1 for the following experiments and explore the behaviour of base method, subsampling ensemble and feature bagging ensemble over a range of k. We see, as an example, in Figure 8, a slight but steady increase of the ROC AUC with k for the base methods and the subsampling ensemble while the feature bagging ensemble appears to be much more instable. While increasing k does not, in general, increase the quality of the results, we observe the same pattern of stability of the base method and the subsampling ensemble and higher variance of the feature bagging ensemble on other datasets as well.

For the three datasets based on segment, for $k = 20$ (again a selection that gives reasonable results for most of the base methods), we show results for all three base methods in Figure 9. Again, the subsampling ensemble compares favourably against the base method as well as against feature bagging.

5. CONCLUSION

Although we compared the sample-based ensemble against feature bagging [30], let us finally note that these two approaches are not strictly competitors. Feature bagging is likely to be an interesting approach in the context of very



(a) LDOF, $k = 50$

(b) LOF, $k = 50$

(c) LoOP, $k = 50$

Figure 7: ROC AUC for ensemble members of the subsampling ensemble for different sample sizes (boxes), the base method (sample size=1.0), and ensembles (diamonds)—on top of subsamples and feature bags (FB)—on dataset Satimage-2.

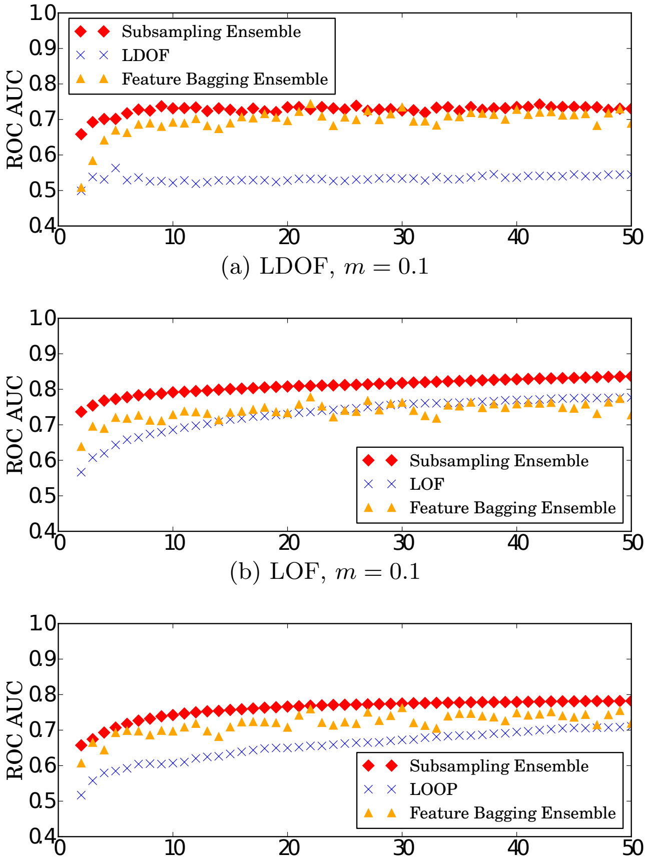
high-dimensional data [45]. Sampling should be helpful when the datasets are growing too large. On the other hand, feature bagging is not meaningful for low-dimensional data, as the ensemble members are bound to be too similar. And sampling on too small data is probably not too promising. However, these two problems (too small datasets with only a few dimensions) are not really problems of todays research. It might be an interesting question for future work to investigate the integration of both techniques, building ensembles on subsets of features and subsets of data objects simultaneously.

Acknowledgments

This work has been partially supported by NSERC (Canada), FAPESP (Brazil), and CNPq (Brazil).

6. REFERENCES

[1] N. Abe, B. Zadrozny, and J. Langford. Outlier detection by active learning. In *Proc. KDD*, pages 504–509, 2006.



(a) LDOF, $m = 0.1$

(b) LOF, $m = 0.1$

(c) LoOP, $m = 0.1$

Figure 8: ROC AUC for base methods and corresponding ensembles varying k on dataset waveform.

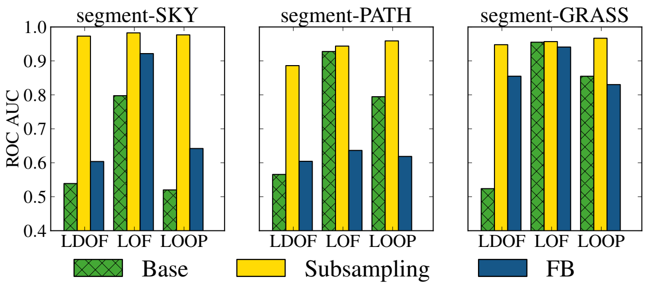


Figure 9: ROC AUC for all methods, $k = 20$, on different datasets (variants of segment).

[2] E. Achtert, S. Goldhofer, H.-P. Kriegel, E. Schubert, and A. Zimek. Evaluation of clusterings – metrics and visual support. In *Proc. ICDE*, pages 1285–1288, 2012.

[3] E. Achtert, H.-P. Kriegel, E. Schubert, and A. Zimek. Interactive data mining with 3d-parallel-coordinate-trees. In *Proc. SIGMOD*, 2013.

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

[5] F. Angiulli and C. Pizzuti. Fast outlier detection in high dimensional spaces. In *Proc. PKDD*, pages 15–26, 2002.

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

[7] S. D. Bay and M. Schwabacher. Mining distance-based outliers in near linear time with randomization and a simple pruning rule. In *Proc. KDD*, pages 29–38, 2003.

[8] A. Bertoni and G. Valentini. Ensembles based on random projections to improve the accuracy of clustering algorithms. In *WIRN / NAIS*, pages 31–37, 2005.

[9] M. M. Breunig, H.-P. Kriegel, P. Kröger, and J. Sander. Data Bubbles: Quality preserving performance boosting for hierarchical clustering. In *Proc. SIGMOD*, pages 79–90, 2001.

[10] M. M. Breunig, H.-P. Kriegel, R. Ng, and J. Sander. LOF: Identifying density-based local outliers. In *Proc. SIGMOD*, pages 93–104, 2000.

[11] G. Brown, J. Wyatt, R. Harris, and X. Yao. Diversity creation methods: a survey and categorisation. *Information Fusion*, 6:5–20, 2005.

[12] T. G. Dietterich. Ensemble methods in machine learning. In *Proc. MCS*, pages 1–15, 2000.

[13] S. Dudoit and J. Fridlyand. Bagging to improve the accuracy of a clustering procedure. *Bioinformatics*, 19(9):1090–1099, 2003.

[14] X. Z. Fern and C. E. Brodley. Random projection for high dimensional data clustering: A cluster ensemble approach. In *Proc. ICML*, pages 186–193, 2003.

[15] A. Frank and A. Asuncion. UCI machine learning repository. `http://archive.ics.uci.edu/ml`, 2010.

[16] A. L. N. Fred and A. K. Jain. Robust data clustering. In *Proc. CVPR*, pages 128–136, 2003.

[17] J. Gao and P.-N. Tan. Converting output scores from outlier detection algorithms into probability estimates. In *Proc. ICDM*, pages 212–221, 2006.

[18] J. Ghosh and A. Acharya. Cluster ensembles. *WIREs DMKD*, 1(4):305–315, 2011.

[19] A. S. Hadi, A. H. M. Rahmatullah Imon, and M. Werner. Detection of outliers. *WIREs Comp. Stat.*, 1(1):57–70, 2009.

[20] S. T. Hadjitodorov, L. I. Kuncheva, and L. P. Todorova. Moderate diversity for better cluster ensembles. *Information Fusion*, 7(3):264–275, 2006.

[21] L. K. Hansen and P. Salamon. Neural network ensembles. *IEEE TPAMI*, 12(10):993–1001, 1990.

[22] W. Jin, A. Tung, and J. Han. Mining top-n local outliers in large databases. In *Proc. KDD*, pages 293–298, 2001.

[23] W. Jin, A. K. H. Tung, J. Han, and W. Wang. Ranking outliers using symmetric neighborhood relationship. In *Proc. PAKDD*, pages 577–593, 2006.

[24] F. Keller, E. Müller, and K. Böhm. HiCS: high contrast subspaces for density-based outlier ranking. In *Proc. ICDE*, 2012.

[25] E. M. Knorr and R. T. Ng. A unified notion of outliers: Properties and computation. In *Proc. KDD*, pages 219–222, 1997.

[26] 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.

[27] H.-P. Kriegel, P. Kröger, E. Schubert, and A. Zimek. LoOP: local outlier probabilities. In *Proc. CIKM*, pages 1649–1652, 2009.

[28] H.-P. Kriegel, P. Kröger, E. Schubert, and A. Zimek. Interpreting and unifying outlier scores. In *Proc. SDM*, pages 13–24, 2011.

[29] H.-P. Kriegel, M. Schubert, and A. Zimek. Angle-based outlier detection in high-dimensional data. In *Proc. KDD*, pages 444–452, 2008.

[30] A. Lazarevic and V. Kumar. Feature bagging for outlier detection. In *Proc. KDD*, pages 157–166, 2005.

[31] H. V. Nguyen, H. H. Ang, and V. Gopalkrishnan. Mining outliers with ensemble of heterogeneous detectors on random subspaces. In *Proc. DASFAA*, pages 368–383, 2010.

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

[33] S. Papadimitriou, H. Kitagawa, P. Gibbons, and C. Faloutsos. LOCI: Fast outlier detection using the local correlation integral. In *Proc. ICDE*, pages 315–326, 2003.

[34] S. Ramaswamy, R. Rastogi, and K. Shim. Efficient algorithms for mining outliers from large data sets. In *Proc. SIGMOD*, pages 427–438, 2000.

[35] P. J. Rousseeuw and M. Hubert. Robust statistics for outlier detection. *WIREs DMKD*, 1(1):73–79, 2011.

[36] E. Schubert, R. Wojdanowski, A. Zimek, and H.-P. Kriegel. On evaluation of outlier rankings and outlier scores. In *Proc. SDM*, pages 1047–1058, 2012.

[37] E. Schubert, A. Zimek, and H.-P. Kriegel. Local outlier detection reconsidered: a generalized view on locality with applications to spatial, video, and network outlier detection. *Data Min. Knowl. Disc.*, 2012.

[38] T. Soler and M. Chin. On transformation of covariance matrices between local Cartesian coordinate systems and commutative diagrams. In *ASP-ACSM Convention*, pages 393–406, 1985.

[39] A. Strehl and J. Ghosh. Cluster ensembles – a knowledge reuse framework for combining multiple partitions. *J. Mach. Learn. Res.*, 3:583–617, 2002.

[40] A. Topchy, A. Jain, and W. Punch. Clustering ensembles: Models of concensus and weak partitions. *IEEE TPAMI*, 27(12):1866–1881, 2005.

[41] G. Valentini and F. Masulli. Ensembles of learning machines. In *Proc. Neural Nets WIRN*, pages 3–22, 2002.

[42] N. H. Vu and V. Gopalkrishnan. Efficient pruning schemes for distance-based outlier detection. In *Proc. ECML PKDD*, pages 160–175, 2009.

[43] J. Yang, N. Zhong, Y. Yao, and J. Wang. Local peculiarity factor and its application in outlier detection. In *Proc. KDD*, pages 776–784, 2008.

[44] K. Zhang, M. Hutter, and H. Jin. A new local distance-based outlier detection approach for scattered real-world data. In *Proc. PAKDD*, pages 813–822, 2009.

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