Theoretical Foundations and Algorithms for Outlier Ensembles*

Charu C. Aggarwal
IBM T. J. Watson Research Center
Yorktown Heights, NY, USA
charu@us.ibm.com

Saket Sathe
IBM Research - Australia
Melbourne, Australia
ssathe@au.ibm.com

ABSTRACT

Ensemble analysis has recently been studied in the context of the outlier detection problem. In this paper, we investigate the theoretical underpinnings of outlier ensemble analysis. In spite of the significant differences between the classification and the outlier analysis problems, we show that the theoretical underpinnings between the two problems are actually quite similar in terms of the bias-variance trade-off. We explain the existing algorithms within this traditional framework, and clarify misconceptions about the reasoning underpinning these methods. We propose more effective variants of subsampling and feature bagging. We also discuss the impact of the combination function and discuss the specific trade-offs of the average and maximization functions. We use these insights to propose new combination functions that are robust in many settings.

1. INTRODUCTION

The problem of outlier ensembles has recently received increasing attention in the research community [1; 2]. Ensemble analysis is used extensively for high-dimensional outlier detection [3; 12; 13; 14; 18]. In high-dimensional outlier detection, multiple subspaces of the data are explored in order to discover outliers. One of the earliest formalizations [14] of outlier ensemble analysis is based on high-dimensional outlier detection. Other recent methods for ensemble analysis in outlier detection are discussed in [11; 15; 17; 19; 21; 24].

Outlier detection is an unsupervised problem, in which labels are not available with data records. As a result, it is generally more challenging to design ensemble analysis algorithms. For example, in the case of boosting, the classifier algorithm needs to be evaluated in the intermediate steps of the algorithm. Such methods are generally not possible in the case of outlier analysis. As discussed in [2], there are unique reasons for ensemble analysis to be generally more difficult in the case of outlier analysis, as compared to classification. In spite of the unsupervised nature of outlier ensemble analysis, we show that the theoretical foundations of outlier analysis and classification are surprisingly similar.

Several arguments have been recently proposed on the theory behind outlier ensembles. In some cases, incorrect new explanations are proposed to explain experimental results, which can be explained by old and well-known ideas. Such a confusion is an impediment to the proper development of ideas in the field because future papers would likely try to explain ensemble improvements in a similar way; this could cause even further confusion. It is also particularly important to give proper attribution and credit to the well-known ideas that explain these results. Our work establishes a correct theoretical understanding of outlier ensemble analysis in terms of well-known ideas from classification. We will also show how these theoretical results can be leveraged to design several new ensemble algorithms.

This paper is organized as follows. In the next section, we provide a review of the bias-variance trade-off for outlier detection, and its similarity and differences with the corresponding trade-off in classification. The applications of these theoretical foundations are discussed in section 3. Section 4 discusses the application of the theoretical foundations to the bias-variance tradeoff. The experimental results are discussed in section 5. Section 6 discusses the conclusions and summary.

2. THE BIAS-VARIANCE TRADEOFF FOR OUTLIER DETECTION

The bias-variance tradeoff is often used in the context of supervised learning. Although it might seem at first sight that labels are required to quantify the bias-variance tradeoff, it turns out that this quantification is also applicable to unsupervised problems, simply by treating the dependent variable as unobserved.

Most outlier detection algorithms output scores to quantify the "outlierness" of data points. After the scores have been determined, they can be converted to binary labels. All data points with scores larger than a user-defined threshold are declared outliers. An important observation about outlier scores is that they are *relative*. In other words, if all scores are multiplied by the same positive quantity, or translated by the same amount, it does not change various metrics (e.g., receiver operating characteristic curves (ROC)) of the outlier detector, which depend only on the ranks of the scores. This creates a challenge in quantifying the bias-variance trade-off for outlier analysis because the *uniqueness* of the score-based output is lost. This is because the ROC provides only an incomplete interpretation of the scores (in terms of *relative* ranks). It is possible to work with crisper definitions of the scores which allow the use of more conventional error measures. One such approach, which preserves uniqueness of scores, is that the outlier detectors always output standardized scores with zero mean, unit variance, and a crisp probabilistic interpretation. Note that one can always

*Handling Editor: Haixun Wang

apply [2] a standardization step as a post-processing phase to any outlier detector without affecting the ROC; this also has a natural probabilistic interpretation (discussed below). Consider a data instance denoted by $\overline{X_i}$, for which the outlier score is modeled using the training data $\mathcal{D}$. We can assume that an ideal outlier score y_i, exists for this data point, even though it is unobserved. The ideal score is output by an unknown function $f(\overline{X_i})$, and it is assumed that the scores, which are output by this ideal function, also satisfy the zero mean and unit variance assumption over all possible points generated by the base data distribution:

$$y_i = f(\overline{X_i}) \tag{1}$$

The interpretation of the score y_i is that by applying the (cumulative) standard normal distribution function to y_i, we obtain the relative outlier rank of $\overline{X_i}$ with respect to all possible points generated by the base data distribution. In a sense, this crisp definition directly maps the score y_i to its (percentile) outlier rank in $(0, 1)$. Of course, *in practice*, most outlier detection algorithms rarely output scores exactly satisfying this property even after standardization. In this sense, $f(\overline{X_i})$ is like an oracle that cannot be computed in practice; furthermore, in unsupervised problems, we do not have any examples of the output of this oracle.

This score y_i can be viewed as the analog to a numeric class variable in classification/regression modeling. In problems like classification, we add an additional term to the RHS of Equation 1 corresponding to the *intrinsic noise* in the dependent variable. However, unlike classification, where the value of y_i is a part of the *observed* data for training points, the value y_i in unsupervised problems only represents a theoretically ideal value (obtained from an oracle) which is *unobserved*. Therefore, in unsupervised problems, the labeling noise[1] no longer remains relevant, although including it makes little difference to the underlying conclusions.

Since the true model $f(\cdot)$ is unknown, the outlier score of a test point $\overline{X_i}$ can only be *estimated* with the use of an outlier detection model $g(\overline{X_i}, \mathcal{D})$ using base data set $\mathcal{D}$. The model $g(\overline{X_i}, \mathcal{D})$ is only a way of approximating the unknown function $f(\overline{X_i})$, and it is typically computed algorithmically. For example, in k-nearest neighbor outlier detectors, the function $g(\overline{X_i}, \mathcal{D})$ is defined as follows:

$$g(\overline{X_i}, \mathcal{D}) = \alpha \text{KNN-distance}(\overline{X_i}, \mathcal{D}) + \beta \tag{2}$$

Here, α and β are constants which are needed to standardize the scores to zero mean and unit variance. It is important to note that the k-nearest neighbor distance, α, and β depend on the specific data set $\mathcal{D}$ at hand. This is the reason that the data set $\mathcal{D}$ is included as an argument of $g(\overline{X_i}, \mathcal{D})$.

If the function $g(\overline{X_i}, \mathcal{D})$ does not properly model the true oracle $f(\overline{X_i})$, then this will result in errors. This is referred to as *model bias* and it is directly analogous to the model bias used in classification. For example, the use of k-NN algorithm as $g(\overline{X_i}, \mathcal{D})$, or a specific choice of the parameter k, might result in the user model deviating significantly from the true function $f(\overline{X_i})$. A second source of error is the *variance*. The variance is caused by the fact that the outlier score directly depends on the data set $\mathcal{D}$ at hand. Any data set is finite, and even if the *expected* value of $g(\overline{X_i}, \mathcal{D})$ correctly reflects $f(\overline{X_i})$, the estimation of $g(\overline{X_i}, \mathcal{D})$ with limited data would likely not be exactly correct. If the data set $\mathcal{D}$ is relatively small, there will be a variance in the estimation of $g(\overline{X_i}, \mathcal{D})$, which is significant. In other words, $g(\overline{X_i}, \mathcal{D})$ will not be the same as $E[g(\overline{X_i}, \mathcal{D})]$ over the space of various random choices of training data sets $\mathcal{D}$. This phenomenon is also sometimes referred to as *overfitting*. The model variance is high when the same point receives very different scores across different choices of training data sets. Although one typically does not distinguish between training and test points in unsupervised problems, one can easily do so by cleanly separating the points used for model building, and the points used for scoring. For example, a k-NN detector would determine the k closest points in the training data for any point $\overline{X_i}$ in the test data. We choose to demarcate training and test data because it makes our analysis cleaner, simpler, and more similar to that of classification; however, it does not change the basic conclusions. Let $\mathcal{D}$ be the training data, and $\overline{X_1} \ldots \overline{X_n}$ be a set of test points whose (hypothetically ideal but unobserved) outlier scores are $y_1 \ldots y_n$. We use an unsupervised outlier detection algorithm that uses the function $g(\cdot, \cdot)$ to *estimate* these scores. Therefore, the resulting scores of $\overline{X_1} \ldots \overline{X_n}$ using the training data $\mathcal{D}$ are $g(\overline{X_1}, \mathcal{D}) \ldots g(\overline{X_n}, \mathcal{D})$, respectively. The mean-squared error, or MSE, of the detectors of the test points over a particular realization $\mathcal{D}$ of the training data is:

$$MSE = \frac{1}{n} \sum_{i=1}^{n} \{y_i - g(\overline{X_i}, \mathcal{D})\}^2 \tag{3}$$

The *expected* MSE, *over different realizations of the training data*, generated using some random process, is as follows:

$$E[MSE] = \frac{1}{n} \sum_{i=1}^{n} E[\{y_i - g(\overline{X_i}, \mathcal{D})\}^2] \tag{4}$$

The different realizations of the training data $\mathcal{D}$ can be constructed using any crisply defined random process. For example, one might construct each instantiation of $\mathcal{D}$ by starting with a larger base data set $\mathcal{D}_0$ and use random subsets of points, dimensions, and so on. The term in the bracket on the RHS can be re-written as follows:

$$E[MSE] = \frac{1}{n} \sum_{i=1}^{n} E[\{(y_i - f(\overline{X_i})) + (f(\overline{X_i}) - g(\overline{X_i}, \mathcal{D}))\}^2] \tag{5}$$

Note that we can set $(y_i - f(\overline{X_i}))$ on the RHS of aforementioned equation to 0 because of Equation 1. Therefore, the following can be shown:

$$E[MSE] = \frac{1}{n} \sum_{i=1}^{n} E[\{f(\overline{X_i}) - g(\overline{X_i}, \mathcal{D})\}^2] \tag{6}$$

This RHS can be further decomposed by adding and subtracting $E[g(\overline{X_i}, \mathcal{D})]$ within the squared term:

$$E[MSE] = \frac{1}{n} \sum_{i=1}^{n} E[\{f(\overline{X_i}) - E[g(\overline{X_i}, \mathcal{D})]\}^2] +$$

$$+ \frac{2}{n} \sum_{i=1}^{n} \{f(\overline{X_i}) - E[g(\overline{X_i}, \mathcal{D})]\}\{E[g(\overline{X_i}, \mathcal{D})] - E[g(\overline{X_i}, \mathcal{D})]\} +$$

$$+ \frac{1}{n} \sum_{i=1}^{n} E[\{E[g(\overline{X_i}, \mathcal{D})] - g(\overline{X_i}, \mathcal{D})\}^2]$$

[1] If there are errors in the feature values, this will also be reflected in the hypothetically ideal (but unobserved) outlier scores. For example, if a measurement error causes an outlier, rather than an application-specific reason, this will also be reflected in the ideal but unobserved scores.

The second term on the RHS of the aforementioned expression evaluates to 0. Therefore, we have:

$$E[MSE] = \frac{1}{n}\sum_{i=1}^{n} E[\{f(\overline{X_i}) - E[g(\overline{X_i}, \mathcal{D})]\}^2] +$$
$$+ \frac{1}{n}\sum_{i=1}^{n} E[\{E[g(\overline{X_i}, \mathcal{D})] - g(\overline{X_i}, \mathcal{D})\}^2]$$
$$= \frac{1}{n}\sum_{i=1}^{n} \{f(\overline{X_i}) - E[g(\overline{X_i}, \mathcal{D})]\}^2 +$$
$$+ \frac{1}{n}\sum_{i=1}^{n} E[\{E[g(\overline{X_i}, \mathcal{D})] - g(\overline{X_i}, \mathcal{D})\}^2]$$

The first term in the aforementioned expression is the (squared) bias, whereas the second term is the variance. Stated simply, one obtains the following:

$$E[MSE] = \text{Bias}^2 + \text{Variance} \quad (7)$$

This derivation is very similar to that in classification although the intrinsic error term is missing because of the ideal nature of the score output by the oracle. The bias and variance are specific not just to the algorithm $g(\overline{X_i}, \mathcal{D})$ *but also to the random process used to create the training data sets* $\mathcal{D}$. Although we did make an assumption on the scaling (standardization) of the scores, the basic result holds as long as the outputs of the base detector and oracle have the same mathematical interpretation. For example, we could very easily have made this entire argument under the assumption that both the base detector $g(\overline{X_i}, \mathcal{D})$ and the oracle $f(\overline{X_i})$ directly output the relative ranks in $(0, 1)$.

Ensemble analysis is a way of combining different models in order to ensure that the bias-variance tradeoff is optimized. This is achieved in several ways:

1. *Reducing bias:* Some methods such as boosting reduce bias in classification by using an ensemble combination of highly biased detectors. However, it is generally much harder to reduce bias in outlier ensembles because of the absence of ground truth.

2. *Reducing variance:* Methods such as bagging, bragging, wagging, and subagging (subsampling) [6; 7; 8], can be used to reduce the model-specific variance in classification. In this context, most classification methods generalize *directly* to outlier ensembles.

The "unsupervised" nature of outlier detection does not mean that bias and variance cannot be defined. *It only means that the dependent variables are not available with the training data, even though an "abstract," but unknown ground truth does exist.* However, the bias-variance tradeoff does not rely on such an availability *to the base algorithm.* None of the steps in the aforementioned computation of MSE rely on the need for $g(\overline{X_i}, \mathcal{D})$ to be computed using examples of the output of oracle $f(\cdot)$ on points in $\mathcal{D}$. This is the reason that variance reduction algorithms for classification generalize so easily to outlier detection.

3. LEVERAGING BIAS-VARIANCE IN OUTLIER ENSEMBLES

The similarity in the theoretical underpinnings of classification and outlier analysis is very convenient. As long as an ensemble method in classification does not require knowledge of the class labels, it can be extended relatively easily to outlier detection.

3.1 Extending Bagging to Outlier Detection

Bagging is used commonly in classification to reduce variance. Typically, a *bootstrapped* sample (i.e., sample with replacement) is drawn in order to construct the training data. The predicted value of the test point is averaged over multiple training samples because the averaged prediction has lower variance. Although it is possible to use bagging for outlier detection, the main problem with doing so is that many base detectors like *LOF* are not very robust to the presence of repeated points, which increases bias. In some variants of bagging for classification, subsampling is used instead of bootstrapping [6; 7; 8; 20]. In this variant of bagging methods, bootstrapping is not used. Rather, training samples are selected from the data *without* replacement. The prediction of each test point is computed by contructing a model on each subsample, and then averaging the prediction from various subsamples. This variant is referred to as *subagging* or *subsampling* [6; 7; 8; 20]. As in the case of bagging it has been shown [6; 7; 8] that the primary effect of subagging is to reduce the variance. Even though subagging is less popular than bagging, it has been shown that subagging is virtually equivalent to bagging and might even have accuracy and computational advantages under many circumstances [6; 7; 8].

The subsampling (subagging) approach can also be generalized directly to outlier detection. Each point in the data is scored with respect to the subsample by a base outlier detector, whether the point is included in the subsample or not. The scores across different subsamples are then averaged. Recently, this adaptation has been explored for outlier detection [24]. Unfortunately, however, this work does not clarify the direct adaptation from the classification domain, and instead provides a different (and incorrect) theoretical explanation.

3.2 Prevailing Misconceptions on Subsampling

Like classification, the subsampling (subagging) approach can be simply explained with the use of the bias-variance trade-off, by treating the dependent variable as *unobserved* in the unsupervised setting. However, in an attempt to create new theory for outlier ensembles, it has been stated in [24], that the unnormalized k-nearest neighbor distances diverge between outlier regions and inlier regions due to subsampling. Specifically, it is stated [24] that the unnormalized kNN-distances in d-dimensional data set increase proportionally to $(k/n_1)^{1/d}$ for a uniformly distributed outlier region containing n_1 points, and the distances increase proportionally to $(k/n_2)^{1/d}$ for an inlier region of the same size containing $n_2 > n_1$ points (in expectation). It is claimed that the absolute outlier-inlier gap $(k/n_1)^{1/d} - (k/n_2)^{1/d}$ increases if we reduce both n_1 and n_2 by the same factor $f < 1$ via subsampling. Specifically, the multiplicative factor by which the gap increases is $(1/f)^{1/d}$. It has been claimed that such an increase in contrast makes the inversion in scores between outliers and inliers less likely. Henceforth, we refer to this argument as the "outlier-inlier inversion argument."

This is, however, an incorrect argument. It is important to understand that downsampling increases the absolute value of the kNN distances (i.e., scales up the scores) because

of greater sparsity of the data. Therefore, if one used the kNN distances as proxies for the outlier scores, then the score differences between the outliers and the inliers will also proportionately increase. This has no *direct* impact on the effectiveness of the outlier detection algorithms, because it is *merely a scaling issue of the scores*. For example, if one multiplied all the outlier scores by $C > 1$, then the absolute divergence between the outliers and inliers will increase, but there will no impact on performance metrics of outlier detection algorithms, such as its receiver operating characteristic. The scenario with subsampling is similar because all expected KNN scores are scaled up in the sparsified subsample by a constant factor of $1/f^{1/d}$. It is important to understand that the absolute divergence of the scores between outliers and inliers has no significance unless it is properly compared to the effect on the *variance* of the scores resulting from this approach. Varance is a key factor regulating the rank-wise correctness of the scores. Variances are scaled up proportionately to C^2, when scores[2] are scaled up by a factor of C. Larger variances make inversion more likely. As we will show in some experimental results in Appendix A, the theoretical claims of "outlier-inlier inversion" are not backed up even over data sets, approximately satisfying the locally uniform assumptions in [24] under which the theoretical results are derived. The inversion argument is quite loosely argued, because it is claimed only for unnormalized k-NN distances in lieu of probability densities; scaling/subsample size impacts the former but not the latter. It does not explain improvements for subsampling in general, or the fact that the experimental improvements in [24] are obtained with the use of distance-normalized algorithms like *LOF*. In fact, as we will see later, *LOF*-like algorithms show much larger ensemble-based improvements as compared to unnormalized algorithms. This behavior is consistent with the bias-variance explanation for outlier ensembles, similar to that in classification.

The paper [24] starts by making the (correct) observation that subsampling [with averaging] reduces the randomness "as expected." This can perhaps be viewed as an informal understanding of variance reduction, which is fairly obvious in such settings because of the earlier subsampling results in the classification domain [6; 7; 8]; even the experimental results in [24] use a classification framework. However, the work in [24] does not try to formally relate to or even cite the existing subsampling results in the classification domain. In fact, the paper explicitly discounts the similarity with the classification problem as a "generic" and "loosely argued" view that does not explain all the performance gains, and it argues for the need for *alternative* theoretical models in outlier ensembles to the bias-variance models popularly used in classification. The result of this alternative analysis is that it does not properly model the bias component, which has a strong impact on the results in the paper. In this context, the paper [24] goes on to make a very surprising (incorrect) statement which seems to support the "outlier-inlier inversion argument": "*Another, more interesting reason for the improved performance is that the base method applied to a smaller subsample of the whole data often shows an improved outlier detection rate, as compared to the same method applied on the whole data set.*" In other words, the statement claims that one can often expect to perform better outlier detection by *randomly* throwing[3] away a majority of the data in the model building phase! Note that this is a statement about the performance of a *single* detector rather than the ensemble, and a set of box-plot figures on the performance of component detectors are also shown to experimentally support this argument in [24]. It is often tempting for researchers to simply accept such engagingly counterintuitive statements without question; however, in this case, this absurd statement is contrary to the most basic principles of statistics. The "less-data-is-better" argument seems almost magical, and it disagrees with everything we know about data science. When speaking of the performance of *individual* ensemble *components* (i.e., base detectors of ensemble), one cannot even fairly compare the subsampled performance of the algorithm with that on the original data set, if the parameters of the algorithm are fixed. For example, if we used a k-NN algorithm with $k = 90$ on a data set with 1000 points, then one must[4] use $k = 9$ for a 10% subsample of size 100 points (i.e., retain same percentile value for k) to ensure that the subsampled algorithm does not have very different bias characteristics. If the value of k is fixed across different subsample sizes, then the specific quirks (i.e., bias) of the detector on a particular data distribution will dominate the performance. In fact, for different choices of k on the same data set and algorithm, the change in bias caused by subsampling could either help or hurt the base detector. The paper [24] only shows experimental scenarios in which the bias component helps the base detector. As a result, an incomplete picture is provided about the effectiveness of subsampling. We can already see that omitting the bias component in any theoretical analysis leads to an incomplete understanding of the effectiveness of subsampling. Although subsampling can improve the accuracy of outlier detectors in general, the reasons for doing so follow trivially from the known results on subsampling [6; 7; 8] in the classification setting, and these are the only valid arguments.

Effects of Bias

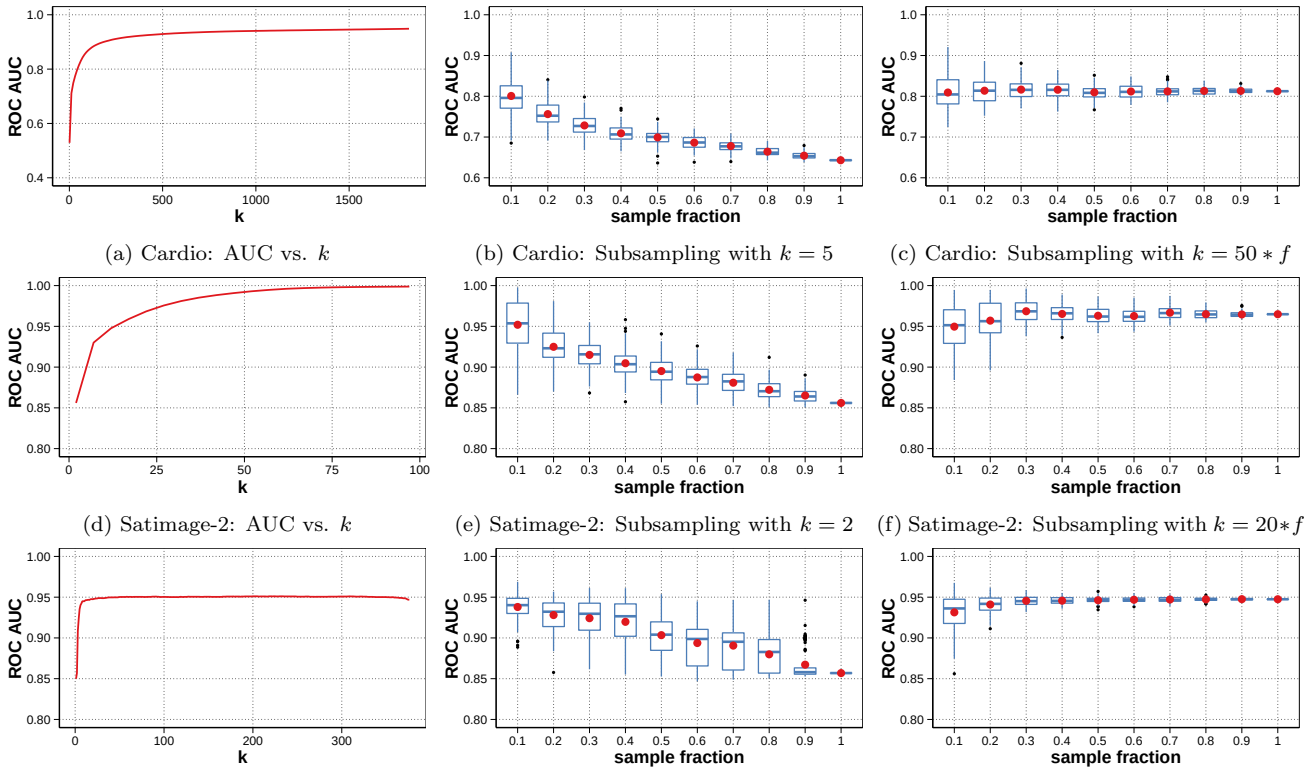
It is noteworthy that if we use random draws of data sets with a particular data size, then the bias of a particular algorithm will depend on the size of the subsample being drawn. A different way of understanding this is that if we apply Equation 7 to only the universe of data sets of a particular size S, the bias term will be sensitive to the value of S. Relative to the full data set, the accuracy can be improved or worsened, depending on whether the bias is increased or reduced. The effect is, of course, highly data distribution-, algorithm-, and parameter-specific. In fact, the improved performance of the individual detectors in [24] (see Figures 4–7 of that paper), is entirely an artifact of this bias but for other data sets/algorithms/parameters, the results could be different. On the other hand, the variance term in Equation 7 will almost always increase with smaller subsamples (i.e., smaller S) because of the statistical unreliability of using less data.

In order to understand this point, consider a data set in

[2]When a random variable is scaled by a factor of $a > 1$, its variance is scaled up by a^2.

[3]In subsampling, only the sampled portion of the data is used for model building, although all points are *scored* against the model.

[4]This is only an approximate adjustment. For some algorithms like *LOF*, the adjustment becomes even more approximate.



(a) Cardio: AUC vs. k (b) Cardio: Subsampling with $k = 5$ (c) Cardio: Subsampling with $k = 50 * f$

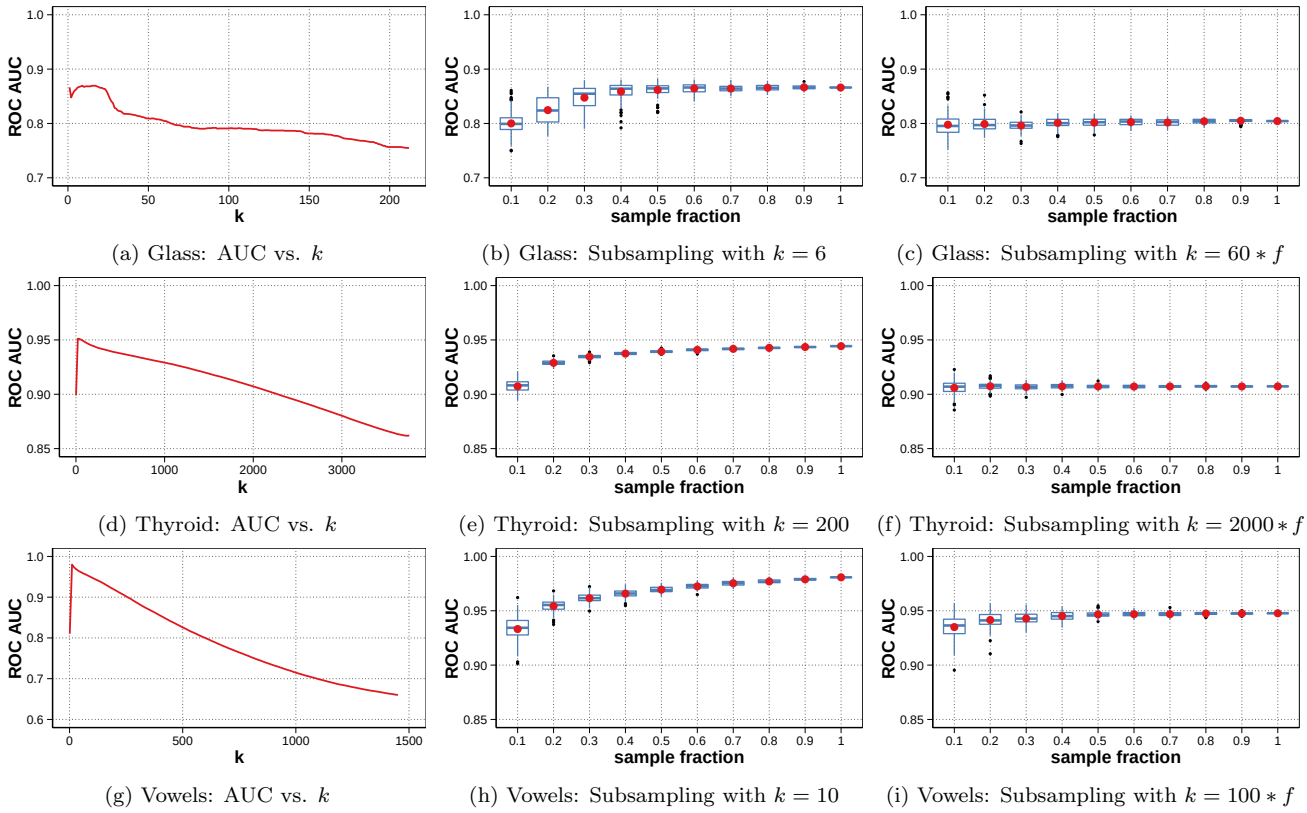
(d) Satimage-2: AUC vs. k (e) Satimage-2: Subsampling with $k = 2$ (f) Satimage-2: Subsampling with $k = 20 * f$

(g) WBC: AUC vs. k (h) WBC: Subsampling with $k = 2$ (i) WBC: Subsampling with $k = 20 * f$

Figure 1: AUC of Avg-KNN increases with k on these data sets. The results show improvement of component detectors at fixed k with smaller subsamples. However, adjusting the value of k by subsample size nullifies (and slightly reverses) this effect because the bias component has been removed and larger subsamples have an inherent statistical advantage.

which a k-NN algorithm shows improved performance with increasing values of k. In this case, the *size* of the sampled data set is important; if one fixed the value of k, and downsampled the data by a factor of $f < 1$, one has effectively increased the *percentile* value of k by a factor of $1/f$. Therefore, if you used a 9-NN algorithm on a sample of 100 points, the bias would be similar to a 90-NN algorithm on a sample of 1000 points, and it would not be comparable to the bias of a 9-NN algorithm on a sample of 1000 points. *In data sets, where the accuracy of a k-NN algorithm increases with k on the full data set, subsampling with fixed k will generally improve the accuracy of an individual detector on a single subsample.* Even though reduced subsample size has a tendency to reduce accuracy because of increased variance, the accuracy can increase when the bias effects in a particular data set are sufficiently large. On the other hand, *in data sets, where the accuracy of a k-NN algorithm reduces with k on the full data set, subsampling with fixed k will generally have significantly reduced accuracy of individual detectors because of the double whammy of greater bias and variance from smaller subsample size.* In general, it is not necessary for a data set to show a monotonic trend with increasing values of k, in which case the bias is entirely unpredictable *and completely dependent on the value of k selected for the base method.* Therefore, no general statement can be made about the base detectors, although the ensemble performance might improve because of the reduced variance of the *ensemble combination*; this is not a new argument [6; 7; 8]. The aforementioned observations for unnormalized kNN-distances are also roughly true for *LOF*-like algorithms, but more approximately so. The improved box-plot performance of *component detectors* in [24] at smaller subsample sizes (see Figures 4–7 of that paper), can be largely attributed to the choice of the parameter k and data sets used.

In order to show this effect, we performed simulations with a number of real data sets with varying accuracy trends with k (described in detail in section 5.1). In this approach, the *average* distance to the k-nearest neighbor distances [4] is reported as the outlier score. We first used the unnormalized distances because the inversion is theoretically claimed [24] for unnormalized distances. Furthermore, adjusting the value of k for bias is easier in this case than in the case of *LOF*, although they are roughly true in the latter case. The data sets with increased accuracy with increasing values of k are shown in Figure 1, and the data sets with reduced accuracy with increasing values of k are shown in Figure 2. We reported the Area Under Curve (AUC) of Receiver Operating Characteristics (ROC) curves. Each row contains three figures for a single data set. The leftmost figure of each row shows the performance of the full data set with increasing values of k. The middle figure of each row shows the performance of the subsample with fixed values of k, but varying subsample size n_i. In the rightmost figure of each row, we adjusted the value of k proportionally to subsample size with the formula $k_i = \lceil k_0 \cdot (n_i/n_0) \rceil$, where n_0 was the size of the



(a) Glass: AUC vs. k (b) Glass: Subsampling with $k = 6$ (c) Glass: Subsampling with $k = 60 * f$

(d) Thyroid: AUC vs. k (e) Thyroid: Subsampling with $k = 200$ (f) Thyroid: Subsampling with $k = 2000 * f$

(g) Vowels: AUC vs. k (h) Vowels: Subsampling with $k = 10$ (i) Vowels: Subsampling with $k = 100 * f$

Figure 2: AUC of Avg-KNN decreases with increasing k on these data sets. The results show significant worsening of component detectors at fixed k with smaller subsamples. However, adjusting the value of k by subsample size ameliorates this effect because the bias component has been removed.

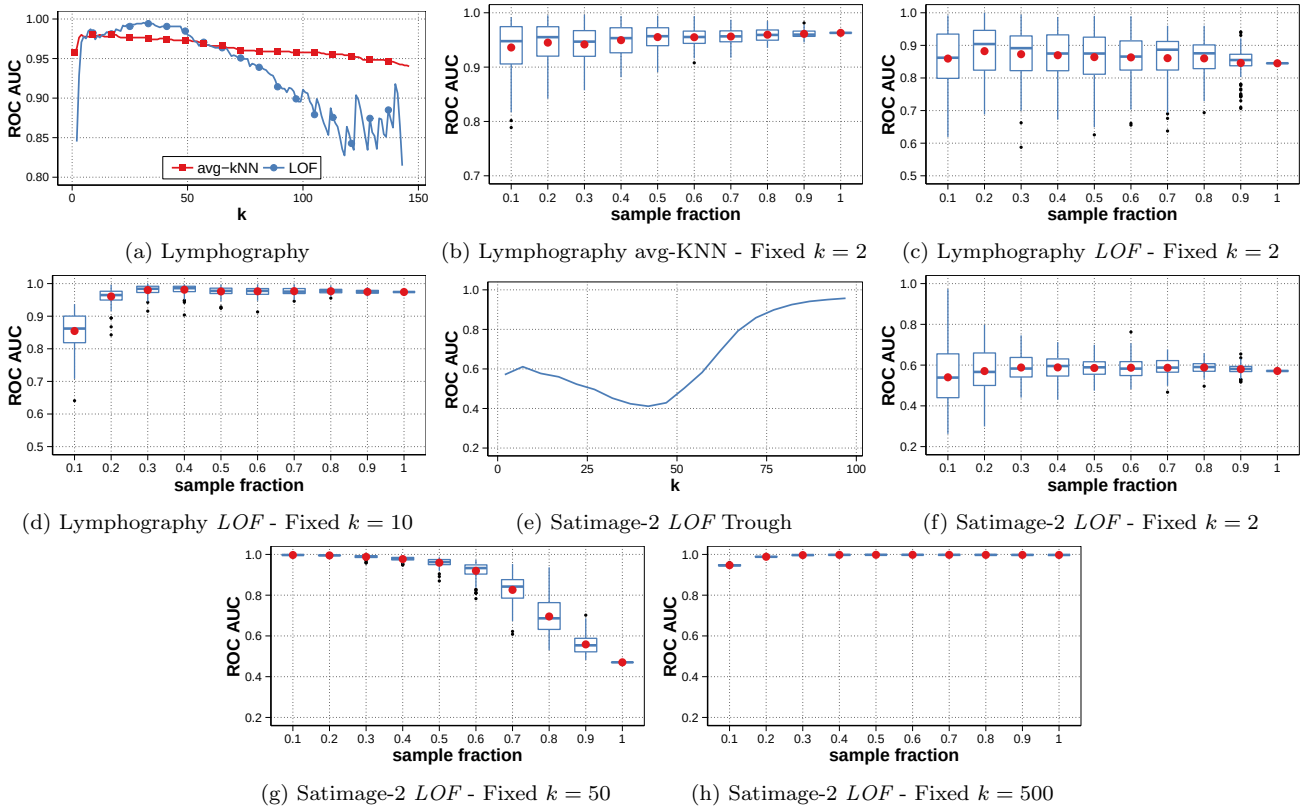
full data set and $k = k_0$ was used for the full data set. The value of k_0 in the rightmost figure was always selected to be 10 times the fixed value of k in the middle figure. As a result, the same[5] value of k was used at subsampling rates of 0.1 in both the fixed-k and adjusted-k cases. However, the performance on the full data would be very different in these cases because of a value of k, which is different by a factor of 10. We ran the base detector 100 times with randomly chosen subsamples, and report the box plots, which show the median (blue line in middle of box) and mean (red dot) performances of the *component detectors*. Note that we are only focusing on component detector performance here in order to understand the bias effects. It is understood that the ensemble will perform better because of known variance reduction effects of subsampling [6; 7; 8]. Nevertheless, we will show in a later section that the performance of component detectors do affect the final ensemble performance to a large extent.

It is evident that for all data sets with increasing accuracy with k, reduction of subsample size improved the performance of the base detector (Figure 1(b), (e), (g)), when the value of k was fixed across different subsample sizes. On the other hand, for data sets with reducing accuracy with increasing value of k, the performance was drastically reduced (Figure 2(b), (e), (g)) by reducing subsample size. In other words, exactly *opposite* trends were obtained in the two types of data sets represented by Figures 1 and 2, respectively.

The most interesting results were for the case where an adjusted value of $k = \lfloor k_0 \cdot (n_i/n_0) \rfloor$ was used. In these cases, the bias effects have been largely removed, and one can see only the impact of the variance. In this case, *consistent* trends were observed in the two types of data sets. In most cases, the accuracy reduced (modestly) with smaller subsample sizes, in *both* types of data sets (Figures 1(c), (f), (i), and Figure 2(c), (f), (i)). This suggests that contrary to the counter-intuitive results suggested in [24], smaller subsamples provide worse performance because of increased variance, once the *data-dependent bias component* has been removed. It is noteworthy that if the optimal value of k_0 on the full data set is less than n_0/n_i, then subsampling with n_i points has an inherent disadvantage for the component detectors, because there is no way of simulating this bias performance on the subsample at any adjusted value of $k \geq 1$. This is a simple artifact of the fact that *randomly* throwing away data leads to irretrievable loss in ability to represent the underlying distribution accurately for outlier detection.

In some data sets, such as the Lymphography data set, we found that the behavior of the algorithm with increasing values of k was algorithm dependent (e.g., Avg-KNN versus LOF-like algorithms). The results are shown in Figure 3(a). The corresponding behavior of the component detectors in subsampling mirrored this behavior. For example, by fixing $k = 2$, the Avg-KNN detector (Figure 3(b)) showed oppo-

[5]The (roughly similar) boxplots show random variations.



(a) Lymphography (b) Lymphography avg-KNN - Fixed $k = 2$ (c) Lymphography LOF - Fixed $k = 2$

(d) Lymphography LOF - Fixed $k = 10$ (e) Satimage-2 LOF Trough (f) Satimage-2 LOF - Fixed $k = 2$

(g) Satimage-2 LOF - Fixed $k = 50$ (h) Satimage-2 LOF - Fixed $k = 500$

Figure 3: Effects of using different algorithms and parameter settings. The bias is highly dependent on the choice of algorithm and parameter setting. However, given the ground-truth, it is easy to predict by plotting AUC versus k

site trends to LOF (Figure 3(c)). This is roughly consistent with the expected trends suggested by Figure 3(a). Furthermore, if the value of k was increased to $k = 10$ for LOF in the subsampling of Lymphography, the results were vastly different, as shown in Figure 3(d). This is simply because LOF already performs quite well at $k = 10$ on the full data set, and subsampling at fraction f and $k = 10$ is (roughly) equivalent to using the algorithm on the full data at a much larger value of k than 10. Such values of $k >> 10$ on the full data would be suboptimal (see Figure 3(a)). In Satimage-2, we found the performance with k to be unpredictable and not monotonic. This result is shown in Figure 3(e). The value of $k = 50$ provided almost the trough in the performance of LOF on the full data set, as shown in Figure 3(e). This value of k seemed to be one of the worst choices for the performance on the full data, and therefore subsampling is guaranteed to improve the bias performance. Therefore, we tried other values of k. The trends[6] at $k = 2$ and $k = 500$ are shown in Figures 3(f) and (h), and they are exactly the opposite of the trends at $k = 50$ (Figure 3(g)).

These results show that the bias induced by subsampling on the component detectors is completely *unpredictable*, *data-dependent*, *algorithm-dependent*, and *parameter-dependent*, although it can be (roughly) predicted simply by plotting[7] the ground-truth AUC performance versus k on the full data set. Of course, since we do not have the ground-truth available in unsupervised problems like outlier detection, there is no way of practically making use of this fact in real settings. The sensitivity of the base detectors to the subsample size also has an important impact on the ensemble performance. As we will show in a later section, even the ensemble performance can be worse than the base detector in some cases. This is because a significant part of the improvements in [24] can be attributed to the better performance of the base detectors at lower subsample sizes. However, since the improvements of the *base* detector with reducing subsample size, as shown in [24], are unpredictable, one cannot bank of it to improve the final ensemble performance of subsampling in every case. In fact, this unpredictable effect, when adverse, can and will swamp the ensemble performance. The main reason that base detectors improve at lower subsample sizes in [24] is not because of the truth of the "outlier-inlier inversion hypothesis" in the base detectors. Rather, the chosen value of k for the base detectors was always around 10% of a near-optimal value on the full data set, and the performance difference between these two values of k on the full data was very large. While discussing parameter choices of various data sets, the authors do state that the value of k is sensitive to the *original* data set size; yet they do not adjust the value of k for subsampled components. The sensitivity of k to data size was used only as a justification for setting k to the larger value of 50 in the Satimage-2 data set because

[6]We used a similar preprocessing phase as in [24] for Satimage-2, which involved sampling one of the classes. The results do vary significantly across different samples and are therefore not exactly comparable to those in [24].

[7]The prediction is even rougher for LOF because of reachability smoothing and the quirky harmonic normalization.

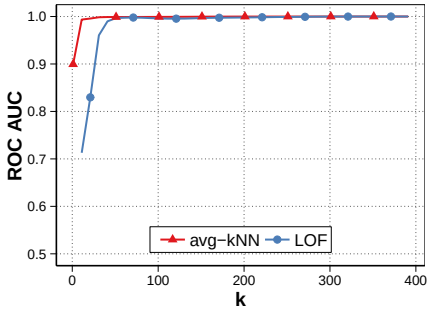


Figure 4: The AUC-vs-k always increases at small values of k in the normal distribution. Therefore, subsampling at very small values of k would be beneficial.

of its large size. All other data sets, including an even larger synthetic data set, were tested at suboptimally small values of $k = 2$ or 3.

Data sets in which pareto-extremes represent outliers often show improved accuracy with increasing values of k. The simplest example is a single Gaussian distribution in which the tails are viewed as outliers. We generated a standard normal distribution of 2000 points where the 3% of points furthest from the mean were tagged as outliers. A plot of the AUC versus k for both the Avg-KNN algorithm and the *LOF* algorithm is shown in Figure 4. It is evident that the AUC increases rapidly with k and stabilizes quickly to almost perfect detection after $k = 50$. Therefore, subsampling at small fixed values of k will show improved bias, although the best improvements will be shown by selecting k to extremely small values in the range $[2, 10]$. However, these same (bias-centered) improvements can be realized simply by running the base method once on the full data with larger values of k. The improved performance with k can also be realized to a limited degree in related data distributions. For example, if one generated multiple Gaussian clusters and treated the pareto-extremes of the clusters as outliers, then the accuracy on the base data will increase with k only for very small values of k. In such cases, subsampling with very small values of k will show improvement on the individual component detectors, because such values of k are suboptimal for the base (full) data.

The effects of the bias in subsampling can help or hurt the component detector performance in an unpredictable way. The salvation lies only in the variance reduction effects of the averaging process in the ensemble, as in classification [6; 7; 8]. However, this salvation is not guaranteed, when there is significant deterioration in base detector performance with respect to full data performance because of the unpredictability in bias characteristics.

A Correct View of Subsampling

It is evident from the aforementioned discussion that all ensemble methods in classification, *which do not require the labels to be observed*, can be trivially generalized to outlier detection. However, the lack of observed labels does cause challenges. For example, when using subagging in classification, one can optimize the parameters for the subsample with cross-validation. This is not possible in unsupervised problems like outlier detection, and the unpredictable performance of the base detectors can sometimes be a liability even after the variance reduction of the ensemble. This tends to make the overall effect of subagging more unpredictable in outlier detection, as compared to data classification. These unsupervised aspects are where outlier ensembles are truly different from classification ensembles.

Although the adaptation of subsampling from classification is a good idea, the paper [24] does not cite, relate to, or credit the existing subsampling (subagging) ideas in classification [6; 7; 8], of which this work is a *direct* derivative. Practically, there are very limited differences (both in theory and experimental frameworks) between subsampling for classification and for outlier detection, compared to other applications like clustering. Only those conclusions in [24], which are consistent with known ensemble theory in data classification, are correct. The portions on improvement of base detectors seem not to be true in general. In fact, the (new) assertions on the improvement of the performance of *individual* detectors can cause confusion in students and young researchers trying to develop new ensemble algorithms. This type of incorrect theory obfuscates what would otherwise be a simple, easily understood, and useful adaptation from classification. It also distracts one from looking for a real solution to the unpredictability of subsampling with base subsample size, which is where the problem is truly different from classification.

The main error in the theoretical results of [24] arises from use of the *unnormalized* k-NN distance gap between outliers and inliers in lieu of probability densities. One cannot make any inferences from this (unnormalized) gap increase without accounting for the corresponding increase in (unnormalized) score variance. Divergence in absolute values of scores makes no difference to the ranks in the outlier scores when all score values are scaled up by the same factor of $(1/f)^{1/d}$. A simple example is where all scores are multiplied by 2, which results in divergence of scores between outliers and inliers but no impact on the outlier detector. This is because variances are scaled up by $2^2 = 4$. Subsampling increases the variances of the scores significantly in a single ensemble component (even after scaling adjustments) because of less training data; this *increases* the probability of inversion. This is the reason that the experiments in Figures 1(c), (f), (i) and the experiments in Figures 2(c), (f), (i), *both* show accuracy reduction after (roughly) adjusting the value of k for the bias effects; another way of understanding this is that less data increases the error from increased variance effects.

Interactions between Base Detector and Ensemble

The overall performance of subsampling will depend on the specific choice of the base detector. For example, if the base detectors output highly correlated scores, then subsampling will not help very much because of poor variance reduction. There are also some unusual cases in which subsampling can perform significantly worse than all the base detectors when measured in terms of the AUC. For example, *LOF* sometimes sets the scores of some points in the neighborhood (see Figure 5) of[8] repeated (duplicate) points to be ∞. This is a weakness in algorithm design, especially since many of these

[8]The *LOF* paper does suggest the use of k-distinct-distances as a *possibility* to fix this problem. The implementation from the LMU group that proposed *LOF* [26] also allows ∞ scores. However, this issue only presents an extreme case of a pervasive problem with *LOF* when k data points are close together by chance at small values of k.

∞ predictions tend to lie in truly dense regions with lots of repeated points. In some unusual cases (see section 5), this can cause *LOF* to have worse-than-random bias (in *expectation*), even when its ROC curves show high values of the AUC over individual detectors. This occurs when different points obtain ∞ scores in different ensemble components. It is only upon averaging the scores, that one finds the ensemble to be worse than its base detectors. In other words, the AUCs of individual base detectors do not reflect the full impact of the ∞ scores, whereas the AUC of the averaged score reflects the *expected* bias of the detector more closely.

Many distance-based detectors can show poor performance when unnaturally small values of k are used on large data sets. However, such values of k might be appropriate for smaller data sets (subsamples). In other words, the optimal value of k typically increases with data size for a particular base distribution. Alternatively, one can fix k at an artificially small value and reduce subsample size to create the illusion of better performance with less data. However, these effects might not be observed at larger values of k.

These implications are important because they show the unexpected interactions that might occur between a base detector and an ensemble method. For example, trying to use bagging instead of subsampling with *LOF* can worsen the ∞ problem because of repetitions in sampled points. Feature bagging can also increase the propensity to create such duplicates in the data set. In all these cases, the performance might seem surprising at first sight, although it can usually be explained from the bias-variance perspective.

Implications for Computational Complexity

It is claimed in [24] that one can improve over a *single* application of the base method on the full data set with subsampling, while also improving accuracy. This is possible only for data sets, such as those in Figure 1, in which the bias helps the component detectors, and therefore a relatively small number of trials is required. When attempting to win only by variance reduction, it is important to use as much of the training data as possible in subsamples. For data sets, like those in Figure 2, where the individual component detectors perform worse than that on the full data sets, many more trials may be required for the variance reduction effects to overcome the bias limitations and it is hard to guarantee improvement in a specific number of trials, if at all.

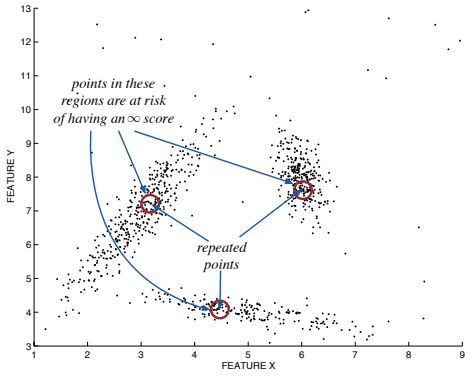


Figure 5: *LOF* can report very large or ∞ scores (false positives) at small k in very dense regions.

In the context of data classification, however, it is well-known [6; 7; 8], that subagging has computational complexity advantages over variance reduction *alternatives* like bagging. However, it needs to be kept in mind that subagging does not have as many problems of bias unpredictability in data classification because of the ability to optimize algorithm parameters with cross-validation. This is not possible in unsupervised problems like outlier detection.

3.3 Variable Subsampling

The unpredictable performance of component detectors in subsampling will also be reflected in the final results from the ensemble, even after variance reduction. In such cases, it is indeed possible for the ensemble to perform *worse* than the base detectors. We will experimentally show several examples of this phenomenon later.

How can one address these challenges and fix subsampling to address challenges, which are specific to outlier detection, and not faced in classification? The simplest solution to this problem is to vary the subsampling rate. As we will see, varying the subsampling rate results in more diverse detectors. Let n_0 be the number of points in the base data set $\mathcal{D}$. The algorithm proceeds as follows:

1. Select f uniformly at random between $\min\{1, \frac{50}{n_0}\}$ and $\min\{1, \frac{1000}{n_0}\}$, where n_0 is the number of points in the original data set $\mathcal{D}$.
2. Select $f \cdot n_0$ randomly sampled points from the original data $\mathcal{D}$, and apply the base outlier detector on this sample to create an outlier detection model. Score each point in $\mathcal{D}$ using this model.

At the end of the process, the scores of each data point in different components are averaged to create a unified score. However, before averaging, the n_0 outlier scores from each detector should be standardized to zero mean and unit variance. This standardization is necessary because subsamples of different sizes will create outlier scores of different raw values for unnormalized KNN-algorithms. We refer to this approach as *Variable Subsampling (VS)*. It is noteworthy that the subsampling approach always selects between 50 and 1000 data points irrespective of base data size. For data sets with less than 1000 points, the maximum raw size would be equal to the size of the data set. For data sets with less than 50 points, subsampling is not recommended.

We now analyze the effect of such an approach on parameter choice, by using the kNN-algorithm as an example. The merit of this approach is that it effectively samples for different values of model parameters. For example, varying the subsample size at fixed k effectively varies the *percentile value of* k in the subsample. In general, holding data size-sensitive parameters fixed, while varying subsample size, has an automatic effect of parameter space exploration. If we view each component detector *after* selecting the subsample size, then it has a bias, which is component dependent. However, if we view the randomized process of selecting the subsample size as a part of the component detector, then every component has the same bias, and the variability in the aforementioned component-dependent bias now becomes a part of this detector variance. One can reduce this variance with ensembling, with the additional advantage that the underlying component detectors of variable subsampling tend to be far less correlated with one another as compared to

fixed subsampling. As a result, one can now aim for better accuracy improvements in the ensemble. Therefore, this approach provides variance reduction not only over different choices of the training data, but also over different randomized choices of k (in an implicit way). In other words, the approach becomes insensitive to specific parameterizations. Although, we have focussed on the parameterization of distance-based detectors here, it is conceivable and likely that such an approach is also likely to make ensembles created with other types of base detectors robust to both parameter and data-size-sensitive design choices. This makes the *VS* approach more general and desirable than simply varying the value of k across detectors; it is independent of the nature of the parameters/design choices in the base detector and it *concurrently* achieves other forms of variance reduction in an implicit way. For data size-sensitive parameters, it is advisable to select them while keeping in mind that subsample sizes vary between 50 and 1000 points. Knowledge of subsample sizes eases the parameter selection process to some extent. For example, for distance-based detectors, we recommend that a value of $k = 5$ will result in a percentile value of k varying between 0.5% to 10% of data size, which seems reasonable.

It is noteworthy that variable subsampling works with raw subsample sizes between 50 and 1000, irrespective of base data size. By fixing the subsample size in a constant range, it would seem at first sight that the approach cannot take advantage of the larger base data sizes. This is, however, not the case; larger data sets would result is less overlap across different subsamples, and therefore less correlation across detectors. This would lead to better variance reduction. The idea is to leverage the larger base data size for better decorrelation across detectors rather than build more robust base detectors with larger subsamples; the former is a more efficient form of variance reduction. After all, the number of points required to accurately model a distribution depends on the absolute subsample size, rather than on the size of the original data set obtained by the data collector. Even if we work under the implicit assumption that a data collector would collect more data for a more complex data distribution, it is unlikely that the required number of data points to accurately model the distribution varies linearly with the collected data size. If desired, one can use other heuristics to increase the robustness of base detector with increasing data size, such as selecting f from $(\min\{1, \frac{50}{n_0}\}, \min\{1, \sqrt{\frac{1000}{n_0}}\})$. The maximum subsampling rate should always reduce with base data size, to increase the de-correlation benefits rather than using it only to improve the base detector.

3.3.1 Computational Complexity of VS

By focusing on an *absolute* size of the subsample, rather than a subsampling *rate*, we have ensured that each detector requires time *linear* in the base data size, rather than quadratic. This is because points in the full data set need to be scored against a subsample of *constant* size. *Therefore, the relative speed-up increases with increasing data size.* In Figure 6, we have *analytically* shown the number of operations of a quadratic base detector, and two variations of the subsampling approach with 100 trials. One is based on a constant maximum subsample size of 1000, and the other is based on a maximum subsample size of $\sqrt{1000n_0}$. We assume that the base detector requires $O(n_0^2)$ operations and a

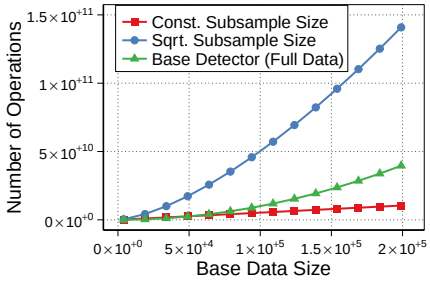


Figure 6: The number of operations required by a quadratic base detector over base data sets of varying size.

100-trial subsampling approach requires $100 \times \frac{n_{max}+50}{2} \times n_0$, where n_{max} is maximum subsample size in a particular type of variable subsampling. For any data set with more than 50000 points, variable subsampling with constant subsample size has a clear advantage over a single application of the base detector, and it would be 20 times faster for a million point data set, although the figure only shows results up to 200,000 points. If one were to extend the X-axis to beyond 5 million points, even the approach using a maximum subsample size of $\sqrt{1000n_0}$ would overtake the base detector. For larger data sizes, most of the base data points might not even be included within one of the 100 subsamples; nevertheless, the accuracy could be superior to that of a model on the (full) base data because increasing data size on a single detector is an inefficient way of reducing variance as compared to variable subsampling. The only way of consistently doing better with less data is to use a better designed technique rather than using an identical method on less data.

3.4 A Review of Feature Bagging

The feature bagging method [14] samples different subsets of dimensions. The basic idea is to sample a number r between $\lfloor d/2 \rfloor$ and $d-1$, and then select r dimensions randomly from the data set. The base detector is applied to this lower-dimensional projection. The scores across various components are then averaged, although other combination methods were also proposed in [14].

Feature bagging (with averaging) is a method that reduces detector variance. Feature bagging with a particular subset of dimensions has a bias that depends on the selected dimensions. However, if one views the step of randomly selecting the subset of dimensions as a *part* of the component detector, then each such (randomized) detector has exactly the same bias, and the aforementioned variability in the bias across different dimension-specific instantiations now becomes a part of this (randomized) detector variance. In such cases, using an average combination is able to achieve variance reduction. The smaller the subset of dimensions selected, the greater the variance reduction. This is because the underlying detectors tend to be relatively uncorrelated if few overlapping dimensions are selected by different detectors. However, if all dimensions are informative, the bias characteristics of such an approach are likely to work against feature bagging because down-selecting the dimensions will lose information.

In this context, it needs to be pointed out that the method in [14] proposes to always randomly select between $\lfloor d/2 \rfloor$

and $d-1$ dimensions; one doesn't always gain the best variance reduction by selecting so many dimensions because of correlations between different detectors. Correlations between detectors hinder variance reduction. One might even select the same subset of dimensions repeatedly, while providing drastically worse bias characteristics. In particular, consider a 6-dimensional data set. The number of possible 3-dimensional projections is 20, the number of possible 4-dimensional projections is 15, and the number of 5-dimensional projections is 6. The total number of possibilities is 41. Therefore, most of the projections (and especially the 4 and 5-dimensional ones) will be repeated multiple times in a set of 100 trials, and not much variance can be reduced from such repetitions. On the other hand, the 3-dimensional projections, while more diverse in overlap and repetition, will have deteriorated bias characteristics. This will also be reflected in the final ensemble performance. Here, it is important to note that the most diverse dimensions provide the worst bias characteristics and vice versa. How can one improve both simultaneously?

3.5 Rotated Bagging (RB)

A natural solution is to devise a randomized sampling scheme that reduces the correlations among detectors. We propose to use *rotated bagging*, in which the data is rotated to a random axis system before selecting the features. The random rotation provides further diversity. A salient observation is that real data sets often have significant correlations, and the projections along different directions are correlated with one another. This means that we can afford to use a much lower dimensionality than $d/2$ to represent the data without losing too much information. In real data sets, the implicit dimensionality usually does not grow much faster than $\sqrt{d}$ with increasing dimensionality d. Therefore, we propose to use $2+\lceil\sqrt{d}/2\rceil$ orthogonal directions from the rotated axis-system as the set of relevant feature bags. Using a lower dimensional projection helps in increasing diversity and therefore it leads to better variance reduction. At the same time the $2+\lceil\sqrt{d}/2\rceil$ dimensions are able to roughly capture most of the salient modeling information in the data because of the random orientation of the axis system. In other words, one is able to increase the potential of better variance reduction without compromising bias too much.

The approach is not designed to work for 3 or less dimensions. Therefore, a constant value of 2 is added up front to prevent its use in such cases. The component detectors will be more uncorrelated in high dimensional cases, which yields a better opportunity for variance reduction. The overall algorithm works as follows:

1. Determine a randomly rotated axis system in the data.
2. Sample $r = 2 + \lceil\sqrt{d}/2\rceil$ directions from rotated axis system. Project data along these r directions.
3. Run the outlier detector on projected data.

After running the detector, the scores can be averaged with a primary goal of variance reduction. It is important to use standardization on the scores before the combination. However, other choices for combination are possible, which will be discussed in a later section.

How can one determine $r = 2 + \lceil\sqrt{d}/2\rceil$ randomly rotated mutually orthogonal directions? The basic idea is to generate a $d \times r$ random matrix Y, such that each value in the matrix is uniformly distributed in $[-1, 1]$. Let the tth column of Y be denoted by $\overline{y_t}$. Then, the r random orthogonal directions $\overline{e_1} \ldots \overline{e_r}$ are generated using a straightforward Gram-Schmidt orthogonalization of $\overline{y_1} \ldots \overline{y_r}$ as follows:

1. $t = 1$; $\overline{e_1} = \frac{\overline{y_1}}{|\overline{y_1}|}$
2. $\overline{e_{t+1}} = \overline{y_{t+1}} - \sum_{j=1}^{t} (\overline{y_{t+1}} \cdot \overline{e_j})\overline{e_j}$
3. Normalize $\overline{e_{t+1}}$ to unit norm.
4. $t = t + 1$
5. if $t < r$ go to step 2

Let the resulting $d \times r$ matrix with columns $\overline{e_1} \ldots \overline{e_r}$ be denoted by E. The $n_0 \times d$ data set D is transformed and projected to these orthogonal directions by computing the matrix product DE, which is an $n_0 \times r$ matrix of r-dimensional points. We refer to this approach as *Rotated Bagging (*RB*)*.

3.6 Variable Subsampling with Rotated Bagging (VR)

It is possible to combine the base detectors in variable subsampling and rotated bagging to create an even more diverse base detector. This will help in variance reduction. Furthermore, because of the reduction in terms of *both* points and dimensions, significant computational savings are achieved. The combined base detector is created is as follows:

1. Project the data into a random $2+\lceil\sqrt{d}/2\rceil$-dimensional space using the rotation method of the previous section.
2. Select a variable size subsample using the approach described in section 3.3.
3. Score each point using the reduced data set.

The scores of these individual detectors can then be combined into the final ensemble score. It is important to use Z-score normalization of the scores from the base detectors before combination. We refer to this approach as variable subsampling with rotated bagging (*VR*).

3.7 Observations on Computational Benefits

Rotated bagging has clear computational benefits because one is using only $\sqrt{d}$ dimensions. With increasing dimensionality the benefit increases. When combined with variable subsampling, the benefits can be very significant. For example, for a data set containing ten million points and 100 dimensions (i.e., a billion entries), each ensemble component would use a data matrix of size at most 1000×7 (i.e., less than a ten-thousand entries). In space-constrained settings, this can make a difference in terms of being able to use the approach at all. For 100 trials, the ensemble (containing quadratic base detectors) would be hundreds of times faster than a single application of the base method on the full data.

3.8 Other Variance Reduction Methods

As discussed earlier, the similarity of the bias-variance tradeoff in outlier detection to that of classification means that one can trivially adapt many classification ensemble algorithms to outlier detection. For example, bagging, bragging, wagging, subagging, and various forms of diversity incorporation can be easily adapted to outlier detection. With some

methods such as bootstrapped aggregation, care should be taken to use detectors that perform robustly in the presence of repeated instances. Using *LOF* as a base detector would be a bad idea, without proper handing of repeated instances within the implementation. There is even a rich literature on diversity incorporation by artificially adding training data or otherwise perturbing the training data [16]. Note that the work in [25] is a variation of this basic idea, although it provides a different theoretical justification. New theoretical arguments do not need to be invented for the effectiveness of these methods, because they follow *trivially* from the arguments used in classification. Even the benchmarking of these outlier detection ensembles is done within a supervised framework.

3.9 Ideas for Bias Reduction

Bias reduction is, however, a completely different matter. In outlier detection, it is very hard to reduce bias in a controlled way, although some heuristic ideas are possible based on common observations about "typical" data and how the outlier scores might behave in typical data. Even then, there is no guarantee that such heuristic methods will always reduce bias. The main problem with attempting bias reduction is that most such methods in classification use knowledge of the labels in intermediate steps. This is not possible in unsupervised problems like outlier detection.

An example of a bias reduction approach, which is used commonly in classification, is *boosting* [10]. Boosting uses the labels for evaluation in the intermediate steps of the algorithm. This requirement rules out its adaptation to outlier detection. It has been suggested [23] that one might be able to substitute internal validity measures for the ground truth in methods like boosting. However, the problem with such an approach is that internal validity measures have built-in biases of their own and the results can be misleading. Trying to use an internal validity measure for boosting is a circular argument because all validity measures need to use a model that will have a built-in bias; the bias reduction of the boosted algorithm would then be at the mercy of the quirks (i.e., bias) of this internal validity model. In general, internal validity measures are not fully trusted even in clustering where they have specific biases in favor of particular algorithms. In the context of outlier detection, the problem is even more significant because a small number of errors in evaluating outlier points can have drastic results.

One commonly used heuristic approach, which is discussed in [2], is to remove outliers in successive iterations in order to build a successively more robust outlier model iteratively. This is a type of sequential ensemble. The basic idea is that outliers interfere with the creation of a model of normal data, and the removal of points with high outlier scores will be beneficial for the model in the next iteration. Although it is not guaranteed that the correct data points might be removed, the advantages outweigh the risks, and the approach has indeed been used successfully in the past [5] in an indirect (non-ensemble) form. A softer version of this approach is to simply down-weight points with high outlier scores in subsequent iterations to ensure that outlier points do not overly influence the normal model of points. One can implement this type of down-weighting with biased sampling; this has the additional benefit of reducing variance. Of course, such an approach is not exactly the same as how boosting is understood in the classification literature, where one combines the knowledge from multiple components in a more holistic way. Nevertheless, it has the same overall effect of bias reduction.

4. OUTLIER SCORE COMBINATION

Given the outlier scores from various detectors, a final step of ensemble-based approach is to combine the scores from various detectors. Let us consider the case of a set of m independent detectors, which output the scores $s_1(i) \ldots s_m(i)$ for the ith data points. When the scores are produced by detectors of different types, it is assumed that they are standardized. There are two commonly used combination functions:

1. *Averaging:* The average of the scores $s_1(i) \ldots s_m(i)$ is reported as the final score of the ith data point.
2. *Maximum:* The maximum of $s_1(i) \ldots s_m(i)$ is reported as the outlier score.

Which of these methods of model combination is better? It has been suggested [23] that the averaging variant is better and that the maximum function overestimates the absolute scores [23] by picking out the larger errors. On the other hand, the work in [14] shows some comparative experimental results between the averaging function and a rank-based variant of the maximization function (referred to as *breadth-first* combination in [14]). The results are data-dependent and do not seem to show clear superiority of one method over the other.

A clearer picture may be obtained from the bias-variance trade-off. The effect of averaging is very clear because it results in a reduction of the variance (as in classification). We argue, however, that the specific choice of the combination function often depends on the data set at hand. In real settings, one is often able to de-emphasize irrelevant or weak ensemble (poorly biased) components with the maximization function. Therefore, one is often able to reduce bias. However, the maximization function *might* increase variance, especially for small training data sets. The specific effect will depend on the data set at hand, which is also reflected in the results of [14]. This is yet another example of the power of the venerable bias-variance *trade-off* in understanding all types of ensemble analysis. In our experiments, we found that it was (mostly) in smaller data sets and subsample sizes (i.e., where variance was large), that averaging performed better than maximum,

Next, we explain the bias reduction effects of the maximization combination. In many "difficult" data sets, the outliers may be well hidden, as a result of which many ensemble components may give them inlier-like scores. In such cases, the scores of outlier points are often *relatively* underestimated in most ensemble components as compared to inlier data points. In order to explain this point, let us consider the feature bagging approach of [14], in which the outliers are hidden in small subsets of dimensions. In such cases, depending on the nature of the underlying data set, a large majority of subspace samples may not contain many of the relevant dimensions. Therefore, most of the subspace samples will provide significant underestimates of the outlier scores for the (small number of) true outlier points and mild overestimates of the outlier scores for the (many) normal points. This is a problem of *bias*, which is caused by the well hidden nature of outliers. We argue that such kinds of

bias are inherent[9] to the problem of outlier detection. The scores of outlier points are often far more *brittle* to small algorithm modifications, as compared to the scores of inlier points. Using a maximization ensemble is simply a way of trying to identify components in which the outlier-like behavior is best magnified. Of course, it is fully understood that any bias-reduction method in an unsupervised problem like outlier detection is inherently heuristic, and it might not work for a specific data set. For example, if a training data set (or subsample) is very small, then the maximization function will not work very well because of its propensity of pick out the high variance in the scores.

Clearly, there are trade-offs between the use of the maximization and averaging function and it is difficult to declare one of them as a clear winner. This point also seems to be underscored by the experimental results presented in [14], where the relative behavior of the two methods (i.e., averaging versus maximum rank) depends on the specific data set. In this paper, we will provide experimental results which show further insights.

4.1 A Simple Example

In order to illustrate this point, we will provide a simple example of a toy data set $\mathcal{T}$ and ensemble scheme, where outliers are well hidden in the data set. Consider the case, where a data set has exactly n data points and d dimensions. For the purpose of discussion, we will assume that the value of d is very large (e.g., a few hundred thousand). Assumes that the data set contains a single outlier. For the $(n-1)$ normal data points, the data is uniformly distributed in [-1, 1]. The distribution for the outlier point is slightly different in that a randomly chosen dimension has a different distribution. On exactly $(d-1)$ dimensions, the outlier point is again uniformly distributed in $[-1, 1]$. On the remaining (randomly chosen) dimension, the value of the corresponding attribute is in the range $[2, 3]$.

Note that the single outlier can be trivially discovered by many simple heuristics, although many off-the-shelf distance-based algorithms might not do very well because of the averaging effects of the irrelevant dimensions. In practice, an outlier detection algorithm is not optimized to any particular data set, and one often uses detectors which are not optimized to the data set at hand.

For example, consider the case where the base detector is an extreme value analysis method [22] in which the distance from the data mean is reported as the outlier score. Note that the data distribution of $\mathcal{T}$ is such that the mean of the data can be approximated to be the origin in this case. The ensemble method is assumed to be a variant of the feature bagging scheme [14], in which each dimension in the data is selected exactly once and the detector is applied on this 1-dimensional data set. The process is repeated for each of the d dimensions, and the final score can be reported using either the averaging or the maximum function over these d different scores. Therefore, our simple ensemble-based approach has d components. We will derive the probability that the score for an outlier point is greater than that for an inlier point under both the averaging and maximization schemes. In other words, we would like to compute the probability of a rank inversion in the two cases.

[9]The original *LOF* paper recognized the problem of dilution from irrelevant ensemble components and therefore suggested the use of the maximization function.

The averaging function will yield a combination score for the inlier points, which has a expected value of 0.5 because each score is randomly distributed in $(0, 1)$. The variance of the score is $1/(12 \cdot d)$ over the different ensemble components. On the other hand, by using the same argument, the outlier point will have an expected score of $[0.5(d-1)+2.5]/d = 0.5+2/d$, because the irrelevant dimensions contribute $0.5(d-1)/d$ to the expected value, and the single relevant dimension contributes $2.5/d$ to the expected score. The variance of the score is $1/(12d)$. Therefore, the difference M between the two scores will be a random variable with an expected mean of $\mu = 2/d$ and a variance of $\sigma^2 = 1/(6d)$. Furthermore, this random variable M will be normally distributed when d becomes large. Note that an inversion between the outlier and a randomly selected inlier occurs when M is negative. Let $Z \sim \mathcal{N}(0, 1)$ be a random variable drawn from the standard normal distribution with 0 mean and unit variance. Therefore, we have:

$$\begin{aligned} P(\text{Inversion}) &= P(M < 0) \\ &= P(Z < (0-\mu)/\sigma) = P(Z < -2\sqrt{6/d}) \end{aligned}$$

Note that the expression $2\sqrt{6/d}$ tends to zero with increasing dimensionality, and the resulting probability evaluates to almost 0.5. This means that with increasing dimensionality, an inlier is almost equally likely to have a larger outlier score than a truly outlier point. In other words, the averaging approach increasingly provides performance that is similar to a random outlier score for each point. This is because the data point becomes increasingly hidden by the irrelevant dimensions, and the averaging function continues to dilute the outlier score with increasing dimensionality.

Nevertheless, the maximization function always discovers the correct *relative* score of the outlier point with respect to the inlier points because it it always reports a value in the range $[2, 3]$, which is greater than the outlier score of the other data points. In other words, the maximization ensemble properly corrects for the natural bias of outlier detection algorithms, in which the scores of well-hidden outliers are often more unstable than inliers. In the easy cases, where most outliers are "obvious" and can be discovered by the majority of the ensemble components, the averaging approach will almost always do better by reducing variance effects. However, if it can be argued that the discovery of "obvious" outliers is not quite as interesting from an analytical perspective, the maximization function will have a clear advantage.

4.2 Using Ranks

A related question is whether using ranks as base detector output might be a better choice than using absolute outlier scores. After all, the metrics for outlier detection are based on the rank-wise AUCs rather than the score-wise MSEs. Ranks are especially robust to the instability of *raw* scores of the underlying detectors. For example, the ∞-problem of *LOF* would affect the absolute scores but it would affect the ranks to a much smaller degree. However, ranks do lose a lot of relevant information, when scores convey the true degree of outlierness. In such cases, using ranks could increase bias-centric errors, which might also be manifested in the ranks of the final combination score. Therefore, while ranks might work well with some base detectors, their improved behavior is certainly not universal.

4.3 Balanced Choices

Clearly, the bias-variance trade-off suggests that different combination functions might do better in different settings. Therefore, it is natural to balance the effort in reducing bias and variance by combining the merits of the two methods. We propose two schemes, both of which normalize to Z-scores before applying the combination:

AOM Method: For m ensemble components, we divide the components into approximately m/q buckets of q components each. First, a maximization is used over each of the buckets of q components, and then the scores are averaged over the m/q buckets. Note that one does not need to assign equal resources to maximization and averaging; in fact, the value of q should be selected to be less than m/q. For our implementations, we used 100 trials, with $q = 5$. We refer to this method as *AOM*, which stands for *Average of Maximum*.

Thresh Method: A method suggested in [2], for combining the scores of multiple detectors, is to use an absolute threshold t on the (standardized) outlier score, and then adding the (thresholded and standardized) outlier scores for these components. The threshold is chosen in a mild way, such as a value of $t = 0$ on the standardized score. Note that values less than 0 almost always correspond to strong inliers. The overall effect of this approach is to reward points for showing up as outliers in a given component, but not to penalize them too much for showing up as strong inliers. For our implementations, we always used a threshold value of $t = 0$ on the Z-score. An important point is that such an approach can sometimes lead to tied scores among the *lowest ranked* (i.e., least outlier-like) points having a score of exactly $m * t$. Such ties are broken among the lowest ranked points by using their average standardized score across the m ensemble components. As a practical matter, one can add a small amount $\epsilon * avg_i$ proportional to the average standardized score avg_i of such points, to achieve the desired tie-breaking. We refer to this approach as *Thresh*.

The *AOM* combination scheme is particularly useful when the maximum number of trials is not a concern from the computationally efficiency perspective. For example, with averaging, we found that it was often hard to do much better by significantly increasing the number of trials beyond a certain point. However, to saturate the benefits of combining maximization and averaging (e.g., *AOM*) one would need a larger number of trials. Nevertheless, in this paper, we show that even with the same number of trials, schemes such as *AOM* perform quite well. With faster base detectors, one can run a far larger number of trials to gain the maximum accuracy improvements from *both* bias and variance reduction; indeed many of the ensemble methods proposed in this paper also provide the dual benefit of greater speed. The *Thresh* method can be viewed as a faster way of combining bias and variance reduction, when computational efficiency is important. Other ideas for combining bias and variance reduction include the use of *Maximum-of-Average (MOA)*.

5. EXPERIMENTAL RESULTS

In this section, we provide experimental results showing the relative effectiveness of various ensemble methods. We used the average k-NN and *LOF* methods as base detectors.

5.1 Data Set Descriptions

Table 1: Summary of the data sets.

Data Set	Points	Attributes	Percentage outliers (%)
Glass	214	9	4.2
Lymphography	148	18	4.1
WBC	378	30	5.6
Vowels	1456	12	3.4
Thyroid	3772	6	2.5
Satimage-2	5803	36	1.2
Cardio	1831	21	9.6
Optdigits	5216	64	2.9
Musk	3062	166	3.2

We used nine data sets from the UCI Machine learning repository[10]. In some cases, further preprocessing was required. In cases where one of the classes was already rare, it was labeled as the outlier class. In cases where a data set contained relatively balanced classes, downsampling was necessary to create an outlier class. In some cases, multiple large classes were combined to create inliers and multiple minority classes were combined to create outliers. In the following, we provide a brief description of the data preparation process.

The Glass data set contained attributes regarding several glass types. Here, points of class 6 were marked as outliers, while all other points were inliers. For the Lymphography data set classes 1 and 4 were outliers while the other classes were inliers. The Wisconsin-Breast Cancer (Diagnostics) data set (WBC) contained *malignant* and *benign* classes, and we started with a processed version[11] of the data set. We further downsampled the *malignant* class to 21 outliers, while points in the *benign* class were considered inliers. In the Japanese Vowels (Vowels) data set, we treat each *frame* in the training data as an individual data point, whereas the UCI repository treats a block of frames (utterance) as an individual point. In this case, class (speaker) 1 was downsampled to 50 outliers. The inliers contained classes 6, 7 and 8. Other classes were discarded. The ANN-Thyroid data set is the same as that in [13]. In the Statlog (Landsat Satellite) data set, the training and test data were combined. Class 2 was down-sampled to 71 outliers, while all the other classes were combined to form an inlier class. Our modified data set is referred to as Satimage-2. The Cardiotocography (Cardio) data set contained measurements taken from foetal heart rate signals. The classes in the data set were *normal*, *suspect*, and *pathologic*. The *normal* class formed the inliers, while the *pathologic* (outlier) class was down-sampled to 176 points. The *suspect* class was discarded. In Optdigits, instances of digits 1-9 where inliers and instances of digit 0 were down-sampled to 150 outliers. The Musk data set contained several musk and non-musk classes. We combined non-musk classes j146, j147, and 252 to form the inliers, while the musk classes 213 and 211 were added as outliers without down-sampling. Other classes were discarded. Refer to Table 1 for details of data sets.

5.2 Ensemble Combination Methods

In each case, 100 trials of the base detector were used. The base methods are combined in four different ways.

1. *Averaging:* This is the averaging combination method,

[10] `http://archive.ics.uci.edu/ml/datasets.html`
[11] `http://www.ipd.kit.edu/~muellere/HiCS/`

in which the scores from different base detectors are averaged. In the case of the k-NN detector, the scores are also normalized to Z-values before averaging. All the three new schemes, corresponding to (variable subsampling (VS), rotated bagging (RB), and variable subsampling with rotated bagging (VR), are always normalized of Z-values before averaging, because of the large variations in the scores produced by these methods. The results for averaging are shown as a triangle in each box plot of Figures 7, 8, and 9.

2. *Maximization:* All scores from all algorithms are first converted to Z-values. Then, the maximum scores across all ensemble components for each data point are reported. The maximization ensemble scores are shown with an 'x' in Figures 7, 8, and 9.

3. *Average-of-Maximum (AOM):* The 100 trials were divided into 20 buckets of size 5 each. The maximum Z-score was taken over each bucket of size 5. Then, the resulting 20 scores for each point were averaged. The ensemble performance is shown with a circle in Figures 7, 8, and 9.

4. *Threshold sum (Thresh):* All non-negative Z-scores for each data point were added up over the 100 components to create the unified score. Tie-breaking of lowest-ranked points is performed as discussed earlier. The ensemble performance is shown with a square in Figures 7, 8, and 9.

5.3 Normalization of Base Detectors

The outlier scores in an average k-NN detector is not comparable in different ensemble components, especially when using methods like feature bagging and variable subsampling. Therefore, all ensemble scores using the average k-NN detectors were normalized to Z-values. For LOF, which already produces normalized scores, the scores were not re-normalized to Z-scores for the averaging ensemble in the case of fixed subsampling and feature bagging. This was done in order to be consistent with their original implementations. However, for the maximization and balanced methods, we *always* re-normalized to Z-scores across all ensemble and base methods (including LOF). Furthermore, for the new methods proposed (Variable Subsampling, Rotated Bagging, and the combination), we always re-normalized, irrespective of the nature of the combination method used. This is because these detectors often contained components with such widely varying bias characteristics, that re-normalization was essential to make them comparable. An important quirk in the case of LOF was the case when at least one outlier score was ∞. In such cases, the ∞-scores were excluded while computing the mean and standard deviation for normalization.

5.4 Performance Results

We performed the tests over the nine data sets discussed earlier. We tested using both the average KNN-detector and the LOF detector at values of $k = 5$ and $k = 10$. Two different values of k were used because the performance results (and even the trends with varying subsampling rates) were found to be sensitive to the values of k. In each case, we show the following 14 ensemble methods:

1. *Fixed subsampling:* This is the approach used in [24] at varying subsampling rates starting from 0.1 to 1.0. Note that this results in a total of 10 box plots. The box-plot at 1.0 corresponds to the base detector.

2. *Variable subsampling (VS):* This approach always samples between 50 and 1000 points from the data set. When the data set contained less than 1000 points, the upper bound was set to the data size. Note that this type of variable subsampling explores components with different bias characteristics within the ensemble. This scheme is annotated as VS in the figures.

3. *Feature bagging:* This is the feature bagging method discussed in [14]. This scheme is annotated as FB in the figures.

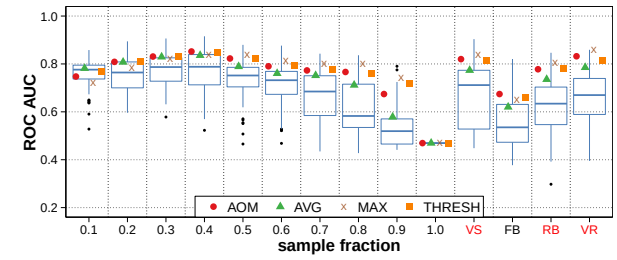
4. *Rotated Bagging:* This is the rotated bagging scheme discussed in the paper, which is annotated as RB.

5. *Variable Subsampling with Rotated Bagging (VR):* This is the combination of the variable subsampling approach with rotated bagging. The scheme is annotated as VR in the figures.

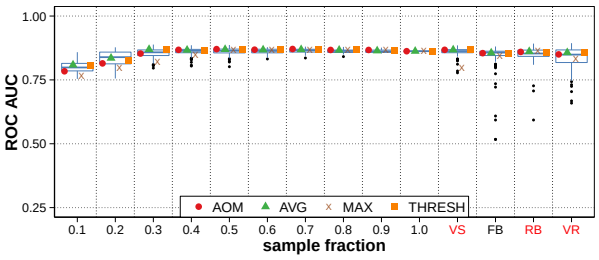
The performances of each of the methods are discussed in Figures 7, 8, and 9, respectively. For each data set, there are four figures corresponding to the two detectors, and values of k set to 5 and 10, respectively. The box-plots in each figure are shown at varying levels of fixed subsampling rates, feature-bagging, and other methods introduced in this paper. Here, we summarize the key findings of our method:

1. Contrary to the claims in [24], smaller subsamples do not always lead to superior performance for the *base detectors*. In particular, the trends depend on the AUC-vs-k curves as discussed earlier, and also on the selected value of k. If the selected value of k is suboptimally small, then it is possible for subsampling to improve the base detector performance. Note that we used reasonably small values of k in our experiments ($k = 5$ and $k = 10$), and yet, the subsampling did not always improve the base detector performance. In fact, for the case of the unnormalized average k-NN detector, significant inversion in base detector performance was observed for only 3 of the 9 data sets. Furthermore, the trends are sometimes different between LOF and average k-NN, and also between $k = 5$ and $k = 10$. This makes the trends unpredictable but they can be fully explained by the AUC-vs-k trends. The inversion was observed more frequently in the larger data sets because the values of k set to 5 and 10 are suboptimally small for such cases. Furthermore, a better *ensemble lift* was obtained with smaller subsample sizes (but not in base detectors) because of better variance reduction.

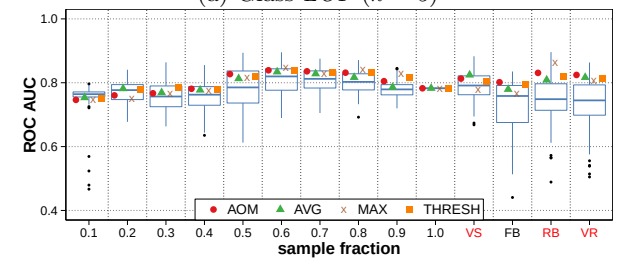
2. In cases, where smaller subsamples showed poor base detector performance, the effects on the ensemble performance were quite significant. In many cases, the better variance reduction of smaller subsamples is not able to overcome the poor base detector performance. This is particularly true for the average k-NN detector algorithm. However, variable subsampling could often perform robustly irrespective of the AUC-vs.-k



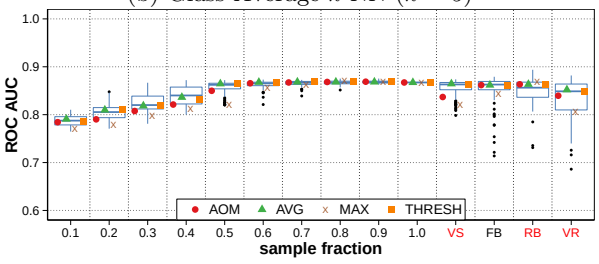
(a) Glass LOF ($k = 5$)



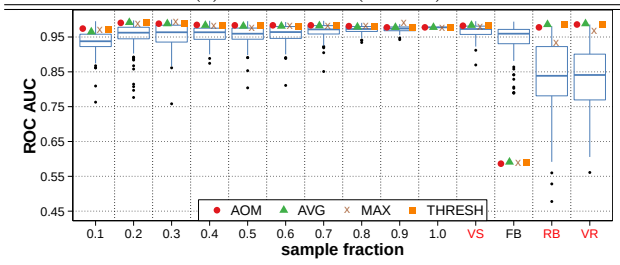
(b) Glass Average k-NN ($k = 5$)



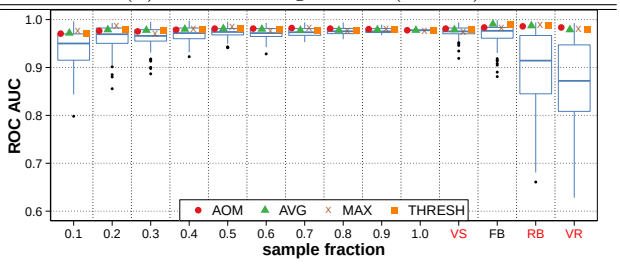
(c) Glass LOF ($k = 10$)



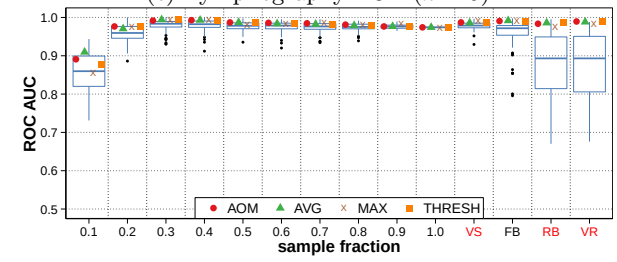
(d) Glass Average k-NN ($k = 10$)



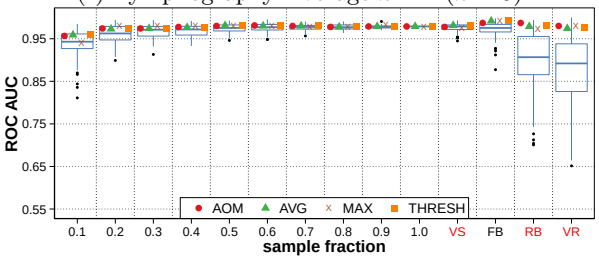
(e) Lymphography LOF ($k = 5$)



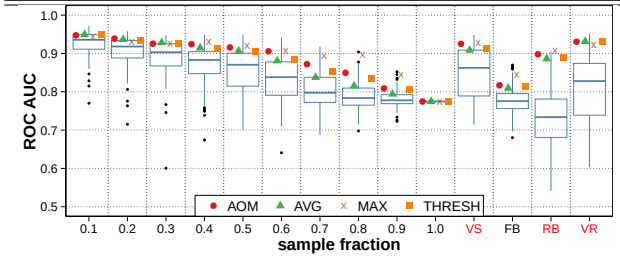
(f) Lymphography Average k-NN ($k = 5$)



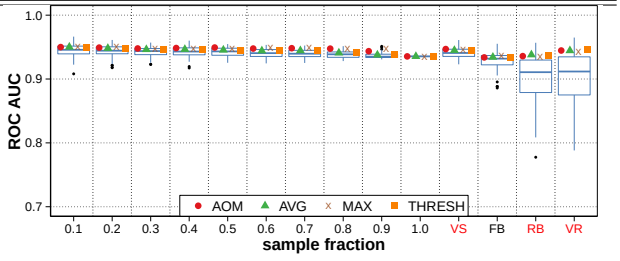
(g) Lymphography LOF ($k = 10$)



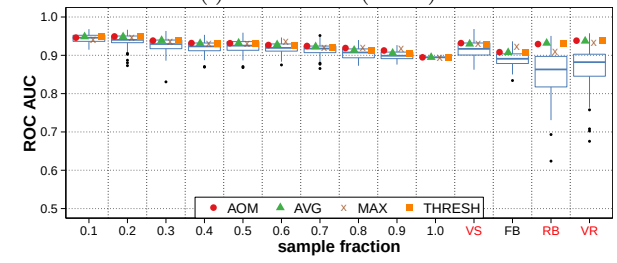
(h) Lymphography Average k-NN ($k = 10$)



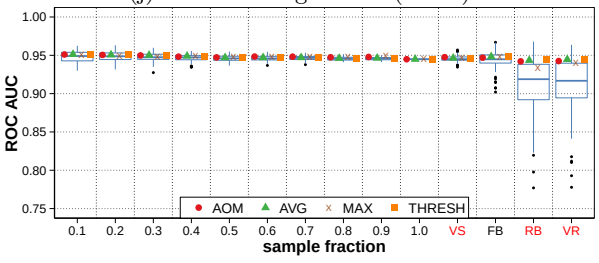
(i) WBC LOF ($k = 5$)



(j) WBC Average k-NN ($k = 5$)

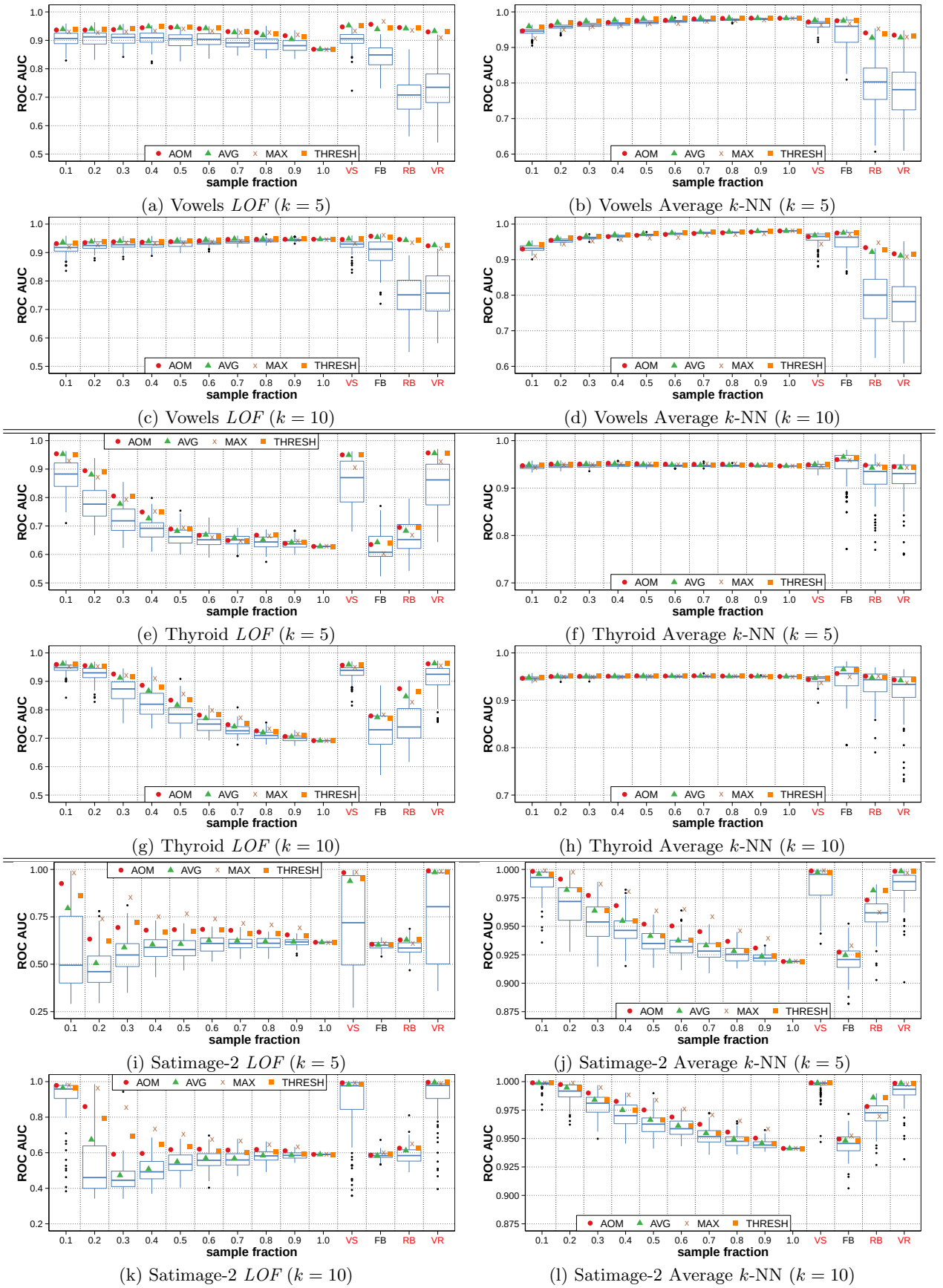


(k) WBC LOF ($k = 10$)



(l) WBC Average k-NN ($k = 10$)

Figure 7: Performance of all ensemble methods (Glass, Lymphography, and WBC)



(a) Vowels LOF ($k = 5$)

(b) Vowels Average k-NN ($k = 5$)

(c) Vowels LOF ($k = 10$)

(d) Vowels Average k-NN ($k = 10$)

(e) Thyroid LOF ($k = 5$)

(f) Thyroid Average k-NN ($k = 5$)

(g) Thyroid LOF ($k = 10$)

(h) Thyroid Average k-NN ($k = 10$)

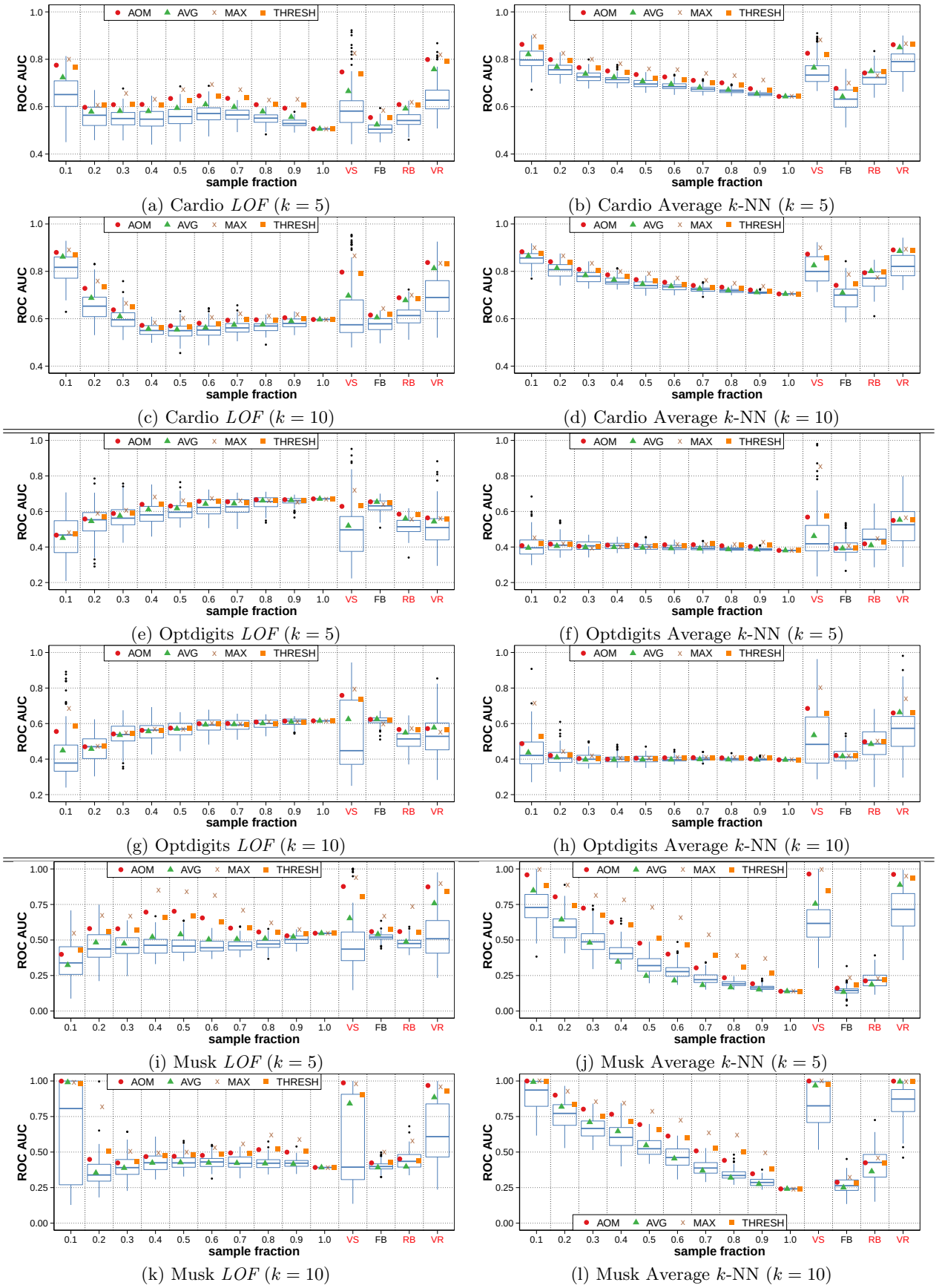
(i) Satimage-2 LOF ($k = 5$)

(j) Satimage-2 Average k-NN ($k = 5$)

(k) Satimage-2 LOF ($k = 10$)

(l) Satimage-2 Average k-NN ($k = 10$)

Figure 8: Performance of all ensemble methods (Vowels, Thyroid, Satimage-2)



(a) Cardio LOF ($k = 5$)

(b) Cardio Average k-NN ($k = 5$)

(c) Cardio LOF ($k = 10$)

(d) Cardio Average k-NN ($k = 10$)

(e) Optdigits LOF ($k = 5$)

(f) Optdigits Average k-NN ($k = 5$)

(g) Optdigits LOF ($k = 10$)

(h) Optdigits Average k-NN ($k = 10$)

(i) Musk LOF ($k = 5$)

(j) Musk Average k-NN ($k = 5$)

(k) Musk LOF ($k = 10$)

(l) Musk Average k-NN ($k = 10$)

Figure 9: Performance of all ensemble methods (Cardio, Optdigits, and Musk)

trends, choice of k, and the base algorithm. This is because variable subsampling was able to effectively ensemble over different *percentile* values of k by varying subsample size. Therefore, irrespective of the effect of subsample size on the bias, the variable subsampling approach was able to perform effectively. More importantly, variable subsampling reduces the *unpredictability* in performance for a fixed subsample size.

3. We (surprisingly) found that the average k-NN detector usually performed better than *LOF* both on the base detector and the final ensemble performance (see section 5.6 for more details). The average k-NN detector was also relatively stable to the selection of various subsample cohorts, and this is reflected by the "thin" boxplots of these methods where the upper and lower ends of the boxes are close together. The thin box plots occur frequently in the fixed subsampling method, where every ensemble component has similar bias. The *LOF* method showed more variation across different executions of the base detector. This was primarily due to the instability of harmonic normalization. This instability can be viewed as a type of diversity that helps variance reduction and results in better *incremental* improvement of *LOF* over base detectors. However, the instability of *LOF* also led to poorer (bias-centric) performance of the base detectors; as a result the overall ensemble performance of *LOF* is poorer. The thin box plots of the average k-NN detector (for fixed subsampling) also meant that one could not obtain much variance reduction advantages from subsampling in the case of the superior (average k-NN) detector. This type of bias-variance trade-off is common in ensemble settings, where one must design the ensemble components to extract the maximum advantages.

4. Since variable subsampling showed more variance across different base components, the box-plots are thicker even in the case of the k-NN detector, and a greater advantage of subsampling was obtained. However, variable subsampling might sometimes have poorer *median* base detector accuracy compared to the *best* fixed-subsampling rate. The final ensemble performance of *VS* was often competitive to or better than the best rate of fixed subsampling (which varied across data sets). Note that it is impossible to know the optimal (fixed) subsampling rate for a particular data set *a priori*. Variable sampling solves this dilemma and thereby reduces the unpredictability of the approaach.

5. Rotated bagging often performed better than feature bagging. In most cases, rotated bagging did not perform as well as variable subsampling. The real advantage of rotated bagging was obtained by combining it with subsampling.

6. When rotated bagging was combined with variable subsampling, the performance was improved slightly over many larger/high dimensional data sets. More importantly, since the combination approach reduces the data set size both in terms of the number of points and the number of dimensions, the approach is extremely fast. Therefore, the primary advantage of the combination approach is one of efficiency. These efficiency advantages can also be made to translate to better accuracy if needed. Like all other ensemble methods, we used only 100 trials for *VR*. Because of the greater computational efficiency of *VR*, it is possible to use many more trials (at the same computational cost) to achieve even better accuracy.

7. Although the averaging ensemble performed quite well, it was certainly not the best combination method over all data sets. In fact, the maximization ensemble performed better than averaging in most of the larger data sets. However, it often performed rather poorly in smaller data sets (or subsample sizes) because it fails to reduce variance. This suggests that the maximization ensemble should not be used for smaller data sets (and subsample sizes) where it can pick out the unstable noise in the scores. However, both the balanced choices (which combine bias and variance reduction), almost always performed better than averaging. Furthermore, we used 100 trials for all combination methods; this often saturates variance reduction for averaging but not for methods like *AOM*, where further gains are possible (by increasing the averaging component) when computational time is not an issue.

8. Feature bagging (with *LOF*) sometimes performed worse than applying *LOF* on the base data with all dimensions included. This was, in part, because of the loss of information associated with dropping dimensions. However, this cannot fully explain the performance in some data sets like Lymphography. In Lymphography, the box plots of the component *LOF* detectors in feature bagging were excellent (see Figure 7(e)) with (most) AUCs above 0.8; yet, the ensemble provided near-random performance. Note that the average k-NN detector does not show this behavior, and the peculiar performance is restricted to *LOF*. Furthermore, we found the ensemble performance to vary significantly across runs in such cases. What explains this unusual behavior?

   This is a case where dropping dimensions leads to repeated instances in the data set. As a result, some (inlier) points have ∞ scores for *LOF*. When the scores are averaged across many components, a very large fraction of the inliers end up with ∞ scores. Because of the ∞ scores, the bias performance of the base detectors are unusually poor, but it is realized only in the AUC of the ensemble, rather than the AUC of the base detectors. This is because most base detectors contain only a small number of ∞ values (or a small number of base detectors contain most of the ∞ values). Therefore, the expected scores of many data points are ∞ over many runs in spite of the high AUCs. By increasing the number of trials to 1000, virtually all data points get ∞ scores. This example also illustrates that the variance reduction of averaging is optimized towards metrics like MSE (as in classification), which may not always be translated to correctness in ranks. Therefore, the *rank-centric* AUCs can occasionally perform worse than all the base detectors in some unusual settings. In some cases, rank-centric detector outputs can be effective for ensembling [14], although the behavior is not universal across detectors

or data sets. The unusual behavior in Lymphography occurs at $k = 5$ rather than at $k = 10$, although some runs at $k = 10$ also deteriorated. This is because harmonic normalization is more unstable at small values of k, where small groups of repeated points (or tight clusters) can throw off the computation. This is also a cautionary tale for attempting to use *LOF* with methods like bagging, which create repeated points in the data via bootstrapping. Although it is possible to use bagging for variance reduction in outlier detection, care must be taken to use base detectors, which are not sensitive to the presence of repeated points.

5.5 Recommendations for Score Combination

The aforementioned experiments suggest a number of general principles for score combination from methods:

1. The averaging method is a low risk-low reward scheme, as it *always* reduces variance. The performance improves over the base detectors most of the time, although significant improvements are usually not observed. It is particularly desirable for smaller data sets, because of its robustness.

2. The maximization method is a high-risk-high-reward scheme, which provides (heuristic) bias-centric improvements in many cases, but it can sometimes also increase variance. Therefore, it occasionally deteriorates below the base detector, especially in smaller data sets and subsample sizes, where it is contraindicated. The maximization function often emphasizes different outliers than the averaging function, which are well-hidden. Often, an analyst may be interested at looking at a different set of results to obtain a different perspective.

3. The balanced schemes provide a reasonably modest reward, at low risk. The gains over averaging were significant enough in so many cases, that these methods could be considered more desirable techniques than pure averaging. These schemes gain their power from their ability to combine the bias reduction in the maximization scheme with variance reduction to significantly lower the risk profile of the maximization detector, while retaining most of the performance gains.

5.6 Impact of Base Detectors

An important observation in Figures 7, 8, and 9, is that the *LOF* method generally gains greater advantage from the ensembling method as compared to the average K-NN method. This is not particularly surprising; the harmonic mean normalization is somewhat unstable, and therefore *LOF* has a better *scope* for improvement as compared to the average k-NN methods. However, how does the *final* ensemble performance of *LOF* compare to the average k-NN detector? It is here that we found some surprising results.

It is generally an article of faith in the research community that *LOF* is a superior detector compared to unnormalized k-NN methods. It is indeed true that *LOF* generally performs better than an *exact* k-nearest neighbor detector, in which the distance to the exact k-nearest neighbor is used as the outlier score. Most existing comparisons between *LOF* and unnormalized distances are based on the exact k-nearest neighbor, and the performance of the average k-NN detector has rarely been compared comprehensively to *LOF*. A surprising result was that we found the average k-NN detector to be superior even to *LOF* on the vast majority of data sets we tested.

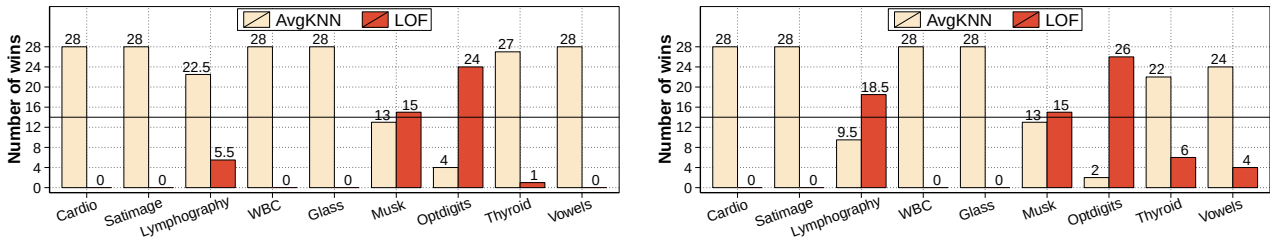
Subsampling is a useful tool for comparing two detectors. By fixing k and varying the subsampling rate, one can effectively test varying bias settings on the data set because the percentile value of k varies with data size. Furthermore, subsampling provides two different measures for evaluation corresponding to base detector performance and ensemble performance. Note that each individual figure (in Figures 7, 8, and 9) contain 14 boxplots including the base detector. For each of the 14 ensemble methods tested (including the base detector), we computed the number of times the average KNN-detector performed better than *LOF*, at both $k = 5$ and $k = 10$. Therefore, there are $14 \times 2 = 28$ comparisons for each data set. For example, the 28 box plots for the *LOF* performance on glass data set in Figures 7(a) and (c), are compared with the (corresponding) 28 box plots for the average k-NN detector in Figures 7(b) and (d). We compared both the base detector performance and the ensemble performance. Therefore, we used either the median of the box plot (base detector performance), or the ensemble performance of the averaging combination method. The former is shown in Figure 10(a), whereas the latter is shown in Figure 10(b). A tie[12] was given a credit of 0.5. Note that the sum of the average k-NN bars and *LOF* bars must always add up to 28 in each case. What is truly astonishing is that the average k-NN detector almost always outperforms *LOF* on the base detector performance, as shown in Figure 10(a). There were several data sets, where the average k-NN detector scored a "clean sweep" (28 wins out of 28) irrespective of the subsampling rate or the value of k, which was chosen. Furthermore, the average k-NN detector also outperforms *LOF* on the final *ensemble* performance, although the performance was slightly less dominant in this case. Note that *LOF* gains a bigger lift from ensembling; however, this lift is often not sufficient to compensate for the poor performance of the base detector.

5.6.1 Is Local Normalization Overrated?

Average k-NN detectors are generally superior to exact k-NN detectors, because they are more robust to local noise in the data distribution [4], but this fact has not received sufficient attention in the research community. For this reason, many of the existing comparisons between *LOF* and unnormalized detectors are often performed using the *exact* k-NN detector, which is a suboptimal implementation of unnormalized detectors. Average k-NN detectors benefit from lower variance. It is noteworthy that *LOF* also uses (roughly) the average k-NN distance[13] in its numerator and harmonically averaged k-NN (in its locality) as its denominator. In this sense, *LOF* is different from exact k-NN distance in two ways, one of which is also reflected in the average k-NN detector. Therefore, it would seem that *LOF* achieves most of its advantage over the exact k-NN de-

[12] A tie on the AUC is an extremely unusual occurrence but it can sometimes occur in smaller data sets, when the number of outliers is small. When the AUC was the same up to at least 4 decimal places, we treated the performance as a tie. This occurred in the case of one of the base detectors on the full data.

[13] *LOF* also uses reachability smoothing.



(a) Median of base detector performance

(b) Ensemble performance (averaging)

Figure 10: *LOF* is often outperformed by the average KNN-detector on both the base and ensemble performance

tector, *not from its local normalization in the denominator*, but from having a more robust numerator. Indeed, local normalization has many problematic issues of incorporating irrelevant noise from dense regions– a specific example is the ∞-problem of *LOF*. While this specific problem can be fixed by using modifications of *LOF*, soft versions of this problem cannot be fixed, when k data points in very dense regions are very close together by chance. In a dense data set, large values of k are required for robustness against false positives (because of harmonic normalization), whereas in a sparse data set, smaller values of k are more appropriate to avoid false negatives. In a data set with widely varying density across different regions, it is impossible to avoid both at a particular value of k. While this problem is also encountered in unnormalized algorithms, local normalization exacerbates this problem more than in unnormalized algorithms. Ironically, the local scheme (*LOF*) has a locality problem in terms of parameter setting. While we recognize that all outliers may not be represented among the rare classes of public data sets, these applications are quite natural. After all, such applications form the primary use-case of unsupervised outlier detection methods when labels are unobserved. Furthermore, because of the typical homogeneity of the *relevant* causality of outliers in most application-centric settings (e.g., cancer or no cancer), *interesting* outliers are often global. In such cases, straightforward average k-NN methods and multivariate extreme value analysis methods (e.g., Mahalanobis method [22]) tend to perform rather well. This strongly suggests that the true benefits of local normalization need to be seriously re-examined by the research community from an application-centric perspective.

5.7 Other Implications

The advantages of fixed-rate subsampling are quite limited when using relatively stable detectors such as the average k-NN detectors. In such cases, the variance reduction lift is quite small, as compared to *LOF*. Note that in all the results presented in Figures 7, 8, and 9, the ensemble performance improves over the median more significantly in *LOF*, as compared to the average k-NN detector. This is primarily because of the instability of harmonic normalization in *LOF*; as in classification, unstable algorithms are always better for variance reduction. Unfortunately, this instability is also reflected in a poorer overall performance of *LOF*, as compared to the average k-NN detectors. In most cases, this pervasive bias cannot be compensated by better variance reduction. One weakness of *fixed-rate* subsampling is that it is generally unable to obtain much lift over the median performance with stable detectors. Variable subsampling is still able to obtain a better lift even with stable detectors because it ensembles over more diverse components.

6. CONCLUSIONS

In this paper, we present theoretical foundations of outlier ensembles and their applications. The bias-variance theory in outlier detection is almost identical to that in classification. Even though outlier detection is an unsupervised problem like clustering, ensemble analysis in outlier detection is more similar to classification as compared to clustering. In particular, most variance-reduction methods can be adapted easily from classification to outlier detection, although other methods like boosting are more challenging to adapt. We use our theoretical results to design several robust variations of feature bagging and subsampling techniques. We also provide a better understanding of the effectiveness of various combination methods, and propose two new combination methods based on bias-variance theory. The results presented in this paper have the potential to motivate the development of new outlier ensemble algorithms along the lines of well-known classification ensemble algorithms.

7. REFERENCES

[1] C. Aggarwal. Outlier Analysis, *Springer*, 2013.

[2] C. Aggarwal. Outlier ensembles: Position paper, *SIGKDD Explorations*, 14(2), 2012.

[3] C. Aggarwal, P. Yu. Outlier detection in high-dimensional data. *SIGMOD*, 2001.

[4] F. Angiulli, C. Pizzuti. Fast outlier detection in high dimensional spaces. *PKDD*, pp. 15–26, 2002.

[5] D. Barbara, Y. Li, J. Couto, J. Lin, S. Jajodia. Bootstrapping a data mining intrusion detection system. In *ACM SAC*, pp. 421–425, 2003.

[6] P. Buhlmann. Bagging, subagging and bragging for improving some prediction algorithms, *Recent advances and trends in nonparametric statistics*, Elsivier, 2003.

[7] P. Buhlmann, B. Yu. Analyzing bagging. *Annals of Statistics*, pp. 927–961, 2002.

[8] A. Buja, W. Stuetzle. Observations on bagging. *Statistica Sinica*, 16(2), 323, 2006.

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

[10] Y. Freund, R. Schapire. A Decision-theoretic generalization of online learning and application to boosting. *Computational Learning Theory*, 1995.

[11] J. Gao, P.-N. Tan. Converting output scores from outlier detection algorithms into probability estimates. *ICDM Conference*, 2006.

[12] Z. He, S. Deng, X. Xu. A unified subspace outlier ensemble framework for outlier detection. *WAIM*, 2005.

[13] F. Keller, E. Muller, K. Bohm. HiCS: High-contrast subspaces for density-based outlier ranking. *ICDE*, 2012.

[14] A. Lazarevic, V. Kumar. Feature bagging for outlier detection, *ACM KDD Conference*, 2005.

[15] F. T. Liu, K. M. Ting, Z.-H. Zhou. Isolation forest. *ICDM Conference*, 2008.

[16] P. Melville, R. Mooney. Creating diversity in ensembles using artificial data. *Information Fusion*, 6(1), 2005.

[17] B. Micenkova, B. McWilliams, I. Assent. Learning representations for outlier detection on a budget. *CoRR abs/1507.08104*, 2015.

[18] E. Muller, M. Schiffer, T. Seidl. Statistical selection of relevant subspace projections for outlier ranking. *ICDE Conference*, 2011.

[19] H. Nguyen, H. Ang, V. Gopalakrishnan. Mining ensembles of heterogeneous detectors on random subspaces. *DASFAA*, 2010.

[20] D. Politis, J. Romano, and M. Wolf. Subsampling. *Springer*, 1999.

[21] S. Rayana, L. Akoglu. Less is more: Building selective anomaly ensembles. *SDM Conference*, 2015.

[22] M. Shyu, S. Chen, K. Sarinnapakorn, L. Chang. A novel anomaly detection scheme based on principal component classifier. *ICDMW*, 2003.

[23] A. Zimek, R. Campello, J. Sander. Ensembles for unsupervised outlier detection: Challenges and research questions, *SIGKDD Explorations*, 15(1), 2013.

[24] A. Zimek, M. Gaudet, R. Campello, J. Sander. Subsampling for efficient and effective unsupervised outlier detection ensembles, *KDD Conference*, 2013.

[25] A. Zimek, R. Campello, J. Sander. Data perturbation for outlier detection ensembles. *SSDBM*, 2014.

[26] `http://elki.dbs.ifi.lmu.de/wiki/Algorithms`

APPENDIX

A. INVERSION ANALYSIS

We have already discussed the reasons for the invalidity of the "outlier-inversion argument [24]" in section 3. Here, we experimentally show the invalidity of the arguments for synthetic data sets, which are generated under the same theoretical assumptions of locally uniform distributions.

We used two 1-d locally uniform distributions and a 2-d distribution with clusters of uniformly distributed points. Consider a data set $\mathcal{D}$ containing the points $\overline{X_1} \ldots \overline{X_n}$, with local probability densities $f_1 \ldots f_n$, which are known from the parameters of the generating distribution. Therefore, these represent ground-truth scores. Let the corresponding scores output by the outlier detection algorithm be $r_1 \ldots r_n$. We say that an inversion has occurred if $f_1 < f_2$ and $r_1 < r_2$. In other words, if a data point with a lower probability density (i.e., in a sparse region), has smaller 1-NN distance than a data point in a dense region, then an inversion is assumed to have occurred. Note that this is the key metric that is analyzed in [24]. For each of the $n \cdot (n-1)/2$ pairs of points in the data set, we computed a non-inversion credit $C(\overline{X_i}, \overline{X_i})$ as follows:

$$C(\overline{X_i}, \overline{X_j}) = \begin{cases} 0 & f_i < f_j \text{ and } r_i < r_j \\ 0 & f_i > f_j \text{ and } r_i > r_j \\ 1 & f_i < f_j \text{ and } r_i > r_j \\ 1 & f_i > f_j \text{ and } r_i < r_j \\ 0.5 & f_i = f_j \text{ or } r_i = r_j \end{cases} \tag{8}$$

The average non-inversion credit $NI(\mathcal{D})$ over all pairs of data points in data set $\mathcal{D}$ is defined as follows:

$$NI(\mathcal{D}) = \frac{\sum_{i<j} C(\overline{X_i}, \overline{X_j})}{n(n-1)/2} \tag{9}$$

In other words, this measure computes the fraction of pairs of points in which the inversion does not occur. Larger values indicate that outliers and inliers will not be inverted. In the ideal case, when no inversions occur, the the value of $NI(\mathcal{D})$ is 1. A value of 0.5 would be expected from a random detector.

Since our primary argument on the effectiveness of subsampling is based on variance, one of the challenges that we faced in our testing was the effect of correlations across multiple ensemble components. Because of the overlaps among the training data sets from various subsamples, the outlier scores (1-NN distances) from various ensemble components are correlated. As a result the variance reduction effects of averaging were curtailed, when the subsamples were large. The problem is that the base data set is finite, and larger subsamples from a base data set always lead to correlated detectors. Correlated detectors generally have a negative effect on any form of bagging or subsampling.

Note that this problem would not be encountered if the base data set were of infinite size. In such a case, the results of any pair of subsamples would be truly independent, and the full effect of variance reduction could be realized. Fortunately, it is indeed possible to simulate such a scenario. In the case of synthetic data sets, the base distribution from which the data set is generated are known, and therefore the subsamples of the desired size can be generated each time from the base distribution. The original base data $\mathcal{D}$ is only used to test the outlier scores against each such generated model. Therefore, we generated two different variants of base detectors and ensembles:

1. We constructed the base detectors by drawing subsamples from the original data set $\mathcal{D}$. This data set was also used as the test data set, but the 1-NN computation of each point in the test data $\mathcal{D}$ was computed only on the subsample of $\mathcal{D}$. The average of the 1-NN

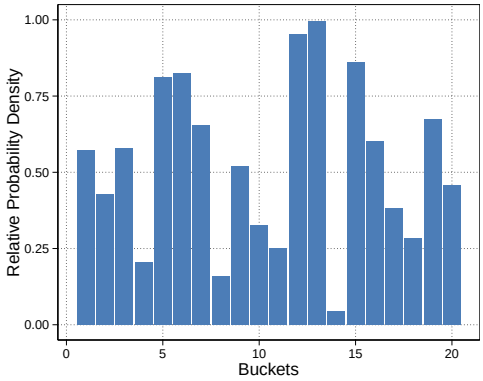
scores provided the ensemble score. The resulting base detector was referred to as *BASE-F* and the ensemble detector was referred to as *ENSEMBLE-F*. The "-F" corresponds to the fact that the base data is finite.

2. In this case, the test data set is fixed to the original data set $\mathcal{D}$, but the subsamples are drawn from an infinite base data set of the same distribution as the test set. This scenario is simulated by generating the subsamples and the test set from the same probability distribution. Note that it is not meaningful to talk of sampling "rates" in this case, because the training data set size is infinite. However, in order to ensure comparability of results with the finite base data, we defined the sampling rate of the subsample with respect to the original (test) data set $\mathcal{D}$. Note that the same test data set $\mathcal{D}$ is used in both finite and infinite sampling. The resulting base detector was referred to as *BASE-I* and the ensemble detector was referred to as *ENSEMBLE-I*. The "-I" at the end of the name refers to the fact that subsampling is performed from a infinite data set. Using an infinite base data has the advantage that it allows us to test whether outlier-inlier inversion results for smaller subsamples are indeed true once the effects of correlation between base detectors have been removed.

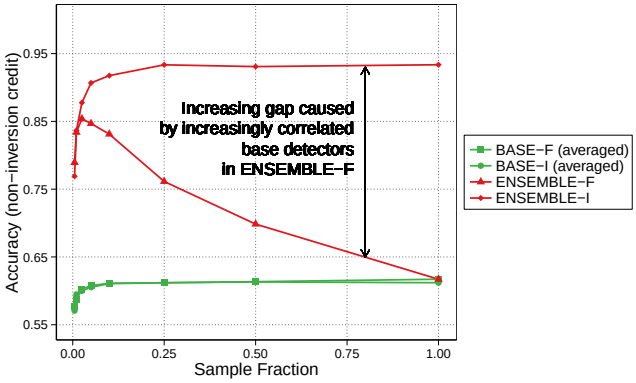
The results in this section used 300 trials. The accuracy of the base detector is computed by averaging the accuracy over each of these 300 instantiations, whereas the accuracy of the ensemble approach is computed using the averaged 1-NN score of the ensemble.

First, we used a data set $\mathcal{D}$ containing 2000 points drawn from locally uniform distributions in a single dimension. We chose the locally uniform distribution because it approximates the conditions under which the theoretical results of [24] are proposed. The data distribution is shown in Figure 11(a). In this case, the data is distributed in 20 1-d bucket. All 1-d points in the ith bucket take on uniformly random values in the range $(i, i+1)$. The relative number of points in each bucket is a uniform random variable drawn from $(0, 1)$, and it is illustrated on the Y-axis of Figure 11a). Therefore, the lower bars correspond to regions which are outlier regions in this 1-d data, albeit uniformly distributed. The values on the Y-axis of Figure 11(a), are used as the ground-truth values of f_i in Equation 8 for the corresponding data points in that bucket. The 1-NN distance is used as r_i in Equation 8. The fraction of non-inversions (i.e., $NI(\mathcal{D})$) of the base system (a 1-NN detector) and ensemble systems both for the case of finite and infinite sampling are illustrated in Figure 11(b). Note that the performance of both base detectors *improves with* the sampling rate, and no advantage was observed for smaller subsamples. This is because the variance effects dominate, and random draws of smaller subsamples have larger variance. It is noteworthy that this choice of the base detector (absolute k-NN) is the same as the one for which the "outlier-inlier inversion argument" in [24] is constructed. Yet, this inversion was not observed in Figure 11(b). The main improvements were achieved with the use of the variance reduction impact of the ensemble. The *ENSEMBLE-F* detector did indeed perform quite well for smaller subsamples, but the improvements were achieved *because of less correlation among the base components*, and therefore better variance reduction. When the subsample size was exactly equal to the size of the full data, no performance improvement was observed because of perfect correlations among the base detectors in *ENSEMBLE-F*. This is substantiated by the fact that the performance of the *ENSEMBLE-I* detector *improves* with increasing subsample size, when the correlations are removed. The gap between the two reflects the gap in variance reduction which arises as a result of increasingly correlated base detectors in *ENSEMBLE-F*. The performance of *ENSEMBLE-I* almost always improves with increasing subsample size, which is a result of the statistical effects of using more data. If the outlier-inlier inversion results claimed in [24] had been indeed true, one would expect that to do better at smaller subsamples in *ENSEMBLE-I*. However, these effects were not observed. We repeated the same experiment with the use of 40 buckets instead of 20 and present the results in Figure 11(c) and (d). The results are very similar to the case of Figures 11(a) and (b).

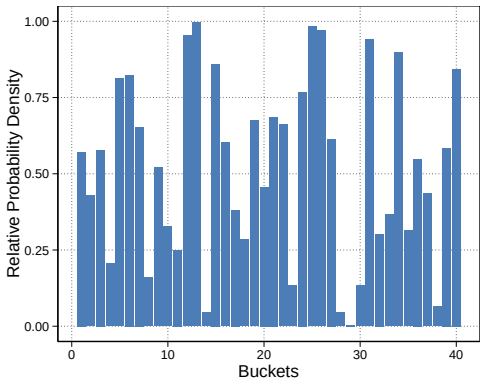
We also tested the effects with 2-d locally uniform distributions of 2000 points. In this case, 30 clusters of uniformly distributed squares were generated, with lower-left corners chosen uniformly at random in $(0, 1)$. Each square had a side of length $1/15$. The relative number of points in each cluster was a uniform random variable in $(0, 1)$, and it represented the ground-truth value of f_i in Equation 8. The corresponding scatter plot is shown in Figure 11(e). The corresponding effects on the non-inversion credit with increasing subsample size are illustrated in Figure 11(f). As in the case of the 1-d distributions, the non-inversions reduced with increasing subsample size. The ensemble based approach *ENSEMBLE-F* initially improved with increasing subsample size, and then the performance started reducing because of increasing correlations among detectors. Here, we have also shown the effect of increasing the number of ensemble components in Figure 11(g) and Figure 11(h). The former (Figure 11(g)) is for the case of the 20-bucket 1-d distribution, whereas the latter is for the case of the 2d-distribution. Both the finite and infinite cases are shown in the same plot. It is noteworthy that larger subsamples generally level off sooner and no advantage is observed by increasing the number of ensemble components. Smaller subsamples initially perform poorly, but because of increasing variance reduction, they can often perform better with increasing number of ensemble components. However, there is a limit to this improvement. Subsamples, which are too small, lose too much information in individual detectors to be effective overall, even with a large number of components. For example, at the lowest sampling rate of 0.005, each subsample contained only 10 points, which was not sufficient to meaningfully represent the 20 or 30 clusters. Therefore, the ensemble performance at this sampling rate could not outperform the ensemble performance at higher sampling rates, even after increasing the number of ensemble components. Note that for the case of *ENSEMBLE-I*, larger subsampling rates almost always provided better performance because the ensemble components were independent, and one could make better use of the greater amount of data. In other words, no outlier-inlier inversion was observed. This is not surprising; the fact that "more-data-is-better" is in tune with the basics of statistics. Clearly, the only significant effect is the variance reduction effect, as in classification.



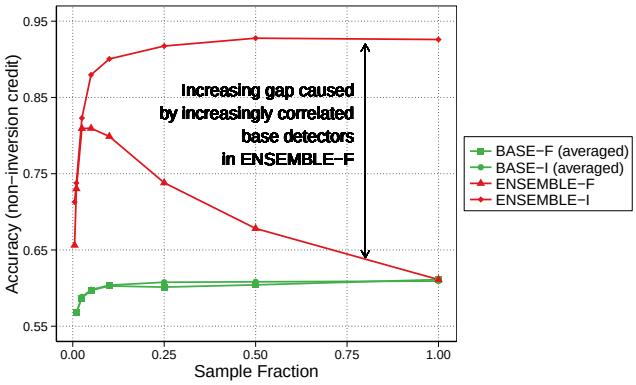
(a) 1D Histogram Distribution (20 Buckets)



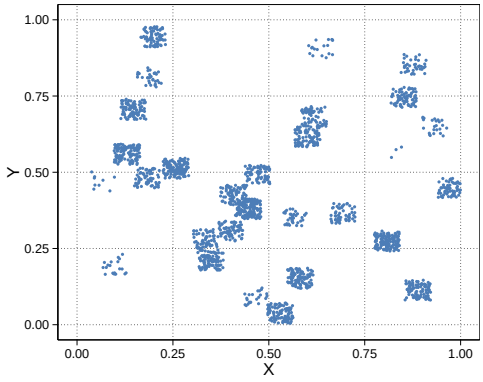
(b) Ensemble/Base Performance (1-d Histogram - 20 Buckets)



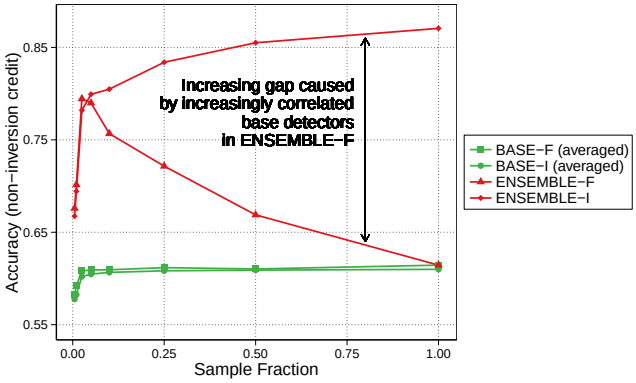
(c) 1D Histogram Distribution (40 Buckets)



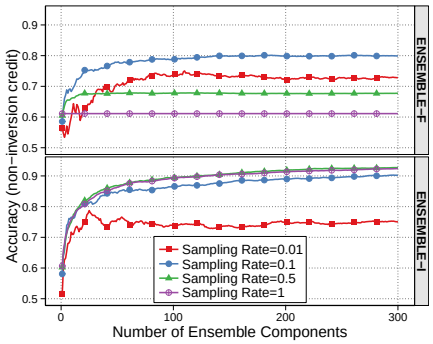
(d) Ensemble/Base Performance (1-d Histogram - 40 Buckets)



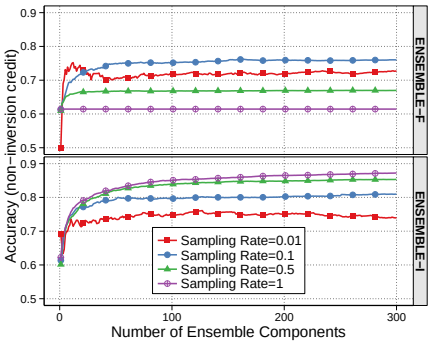
(e) 2D Scatterplot



(f) Ensemble/Base Performance (Increasing sampling rate)



(g) Increasing ensemble components Performance of *ENSEMBLE-F*



(h) Increasing ensemble components Performance of *ENSEMBLE-I*

Figure 11: Effectiveness of base and ensemble on locally uniform data sets (Sampling "rates" for infinite data set are defined with respect to finite base data set $\mathcal{D}$).