

## Chapter 2

# Theory of Outlier Ensembles

*Theory helps us to bear our ignorance of facts.*

George Santayana

### 2.1 Introduction

Outlier detection is an unsupervised problem, in which labels are not available with data records [2]. As a result, it is generally more challenging to design ensemble analysis algorithms for outlier detection. In particular, methods that require the use of labels in intermediate steps of the algorithm cannot be generalized to outlier detection. For example, in the case of boosting, the classifier algorithm needs to be evaluated in the intermediate steps of the algorithm with the use of training-data labels. Such methods are generally not possible in the case of outlier analysis. As discussed in [1], there are unique reasons why ensemble analysis is 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. A number of useful discussions on the theory of classification ensembles may be found in [27, 29, 33]. Further explanations on the use of the bias-variance decomposition in different types of classifiers such as neural networks, support vector machines, and rule-based classifiers are discussed in [17, 30, 31]. It is noteworthy that the bias-variance decomposition is often used in customized ways for different types of base classifiers and combination methods; this general principle is also true in outlier detection.

Several arguments have recently been proposed on the theory explaining the accuracy improvements of outlier ensembles. In some cases, incorrect new arguments (such as those in [32]) are proposed to justify experimental results that can be explained by well-known ideas, and an artificial distinction is made between the theory of classification ensembles and outlier ensembles. A recent paper [4]

clarifies these misconceptions and establishes that the theoretical foundations of outlier analysis are very similar to those of classification. Bias-variance theory [13, 16, 20, 21] is a well-known result in the classification domain, which explains the varying causes of error in different classification methods. This chapter will revisit these results and provide deeper insights behind some of these results. A observation is that even though labels are unobserved in outlier analysis (unlike classification), it does not change the basic foundations of bias-variance theory for outlier ensemble analysis in a significant way. In fact, it was shown in [4] that a minor modification of the existing theory for classification ensembles can also be used for outlier ensembles.

Bias-variance theory characterizes the output of learning processes in terms of the *expected* error of an algorithm over a set of randomized instantiations of the algorithm. It is noteworthy that the applicability of an algorithm on a specific data set is often dependent on a number of randomized choices, some of which are hidden and others are visible. For example, the specific choice of the training data is often achieved through a data collection process that is (often) hidden from the analyst. Nevertheless, the specific choice of the training data induces a random element into the accuracy, which is characterized by a component of bias-variance theory. Similarly, the choice of a specific model or the randomized selection of a particular design choice of the detector is more obviously visible during execution. These randomized choices induce errors, which can be defined as random variables. Bias-variance theory decomposes these randomized errors into two parts, each of which can be reduced with a specific type of ensemble-centric design. Therefore, a proper understanding of the theoretical foundations of outlier ensembles is crucial in designing accurate algorithms that reduce bias or variance.

Intuitively, the model *bias* defines the basic “correctness” of a model. For example, consider a data set in which all the normal points are distributed on the surface of a unit sphere in three dimensions. A single outlier is located in the empty central region of the sphere. In this case, a multivariate extreme value analysis method (e.g., distance from centroid) is the worst possible model to use because it is based on a wrong model of how outliers are distributed. This portion of the error is referred to as the bias. Now, consider a setting in which we used a 1-nearest neighbor algorithm in order to score the points. Even though this model will generally provide good results, it is possible for a *particular* draw of the data from the base distribution to score some of the points on the unit sphere as outliers. This is because such points may be isolated with respect to particular draw, which is regulated by random variance. This portion of the error is referred to as the variance. Bias-variance theory quantifies the error as a combination of these two quantities.

Traditionally, bias-variance theory is defined over a *random process*, which corresponds to the selection of training data sets from a base distribution. Although this view is very useful for explaining several methods like bagging, it is often not quite as useful for explaining the effectiveness of methods like random-forests. In fact, random-forests have not been fully explained [12] even today, even though they are widely recognized to be variance-reduction methods. Therefore, this book will take a more generalized view of bias-variance theory in which the random process

is not only allowed to be draws of the training data but also allowed to be randomized choices made in the base detector itself. This provides a *model-centric* view of bias-variance theory rather than a data-centric view. For the same algorithm, there are therefore multiple ways in which the bias-variance decomposition can be performed. These different ways provide different insights into explaining the effectiveness of the ensemble. We will also explain these differences with a number of simulations on synthetically generated data sets. It is noteworthy that even though the model-centric approach for bias-variance decomposition is proposed for outlier ensembles (in this book), it can be easily extended to classification ensembles, where it has not been explored previously.

This chapter 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. This section will also discuss the effect of the specific random process used for bias-variance decomposition. The applications of these theoretical foundations to the outlier detection problem are discussed in Sect. 2.3. An experimental illustration of bias-variance theory is provided in Sect. 2.4. The effect of using different types of random processes for performing the bias-variance decomposition is also described in this section. Section 2.5 discusses the conclusions and summary.

## 2.2 The Bias-Variance Trade-Off for Outlier Detection

The bias-variance trade-off is often used in the context of supervised learning problems such as classification and regression. Recent work [4] has shown how a parallel bias-variance theory can also be developed for outlier detection. Although labels are not available in outlier detection, it is still possible to create bias and variance quantifications with respect to an unknown but ideal ground-truth. In other words, the bias and variance can be quantified as a *theoretical* construct (with respect to the unobserved ground-truth) but it cannot be evaluated in practice for a particular application. This point of view turns out to be useful in adapting supervised ensemble methods to the unsupervised setting, as long as these methods do not use the ground truth in their intermediate steps. Furthermore, as we will study in this chapter, the bias and variance can be roughly quantified on an experimental basis when rare class labels are used as substitutes for outlier labels in real applications. These relationships between the theoretical foundations of classification ensembles and those of outlier ensembles were first discussed in [4]. The discussion in this section is based on this work.

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 apply [1] 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  $\bar{X}_i$ , 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  $f(\bar{X}_i)$ , 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:

$$y_i = f(\bar{X}_i) \quad (2.1)$$

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  $\bar{X}_i$  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,  $f(\bar{X}_i)$  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 right-hand side of Eq. 2.1 corresponding to the *intrinsic noise* in the dependent variable. However, unlike classification, in which 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<sup>1</sup> 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  $\bar{X}_i$  can only be *estimated* with the use of an outlier detection model  $g(\bar{X}_i, \mathcal{D})$  using base data set  $\mathcal{D}$ . The model  $g(\bar{X}_i, \mathcal{D})$  is only a way of approximating the unknown function  $f(\bar{X}_i)$ , and

---

<sup>1</sup>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.

it is typically computed algorithmically. For example, in  $k$ -nearest neighbor outlier detectors, the function  $g(\bar{X}_i, \mathcal{D})$  is defined as follows:

$$g(\bar{X}_i, \mathcal{D}) = \alpha \text{KNN-distance}(\bar{X}_i, \mathcal{D}) + \beta \quad (2.2)$$

Here,  $\alpha$  and  $\beta$  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,  $\alpha$ , and  $\beta$  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  $g(\bar{X}_i, \mathcal{D})$ . *We note that the above example of  $g(\bar{X}_i, \mathcal{D})$  is only for illustrative in nature and the theoretical results do not assume any particular form of the outlier score such as a density estimator or a  $k$ -nearest neighbor detector.*

If the function  $g(\bar{X}_i, \mathcal{D})$  does not properly model the true oracle  $f(\bar{X}_i)$ , 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  $k$ -nearest neighbor algorithm as  $g(\bar{X}_i, \mathcal{D})$ , or a specific choice of the parameter  $k$ , might result in the user model deviating significantly from the true function  $f(\bar{X}_i)$ . Similarly, if a linear model is used to separate the outliers and inliers, whereas a nonlinear model is more appropriate, then it will lead to a *consistent error* in the scoring process, which corresponds to the bias. 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  $g(\bar{X}_i, \mathcal{D})$  correctly reflects  $f(\bar{X}_i)$ , the estimation of  $g(\bar{X}_i, \mathcal{D})$  with limited data would likely not be exactly correct. In other words,  $g(\bar{X}_i, \mathcal{D})$  will not be the same as  $E[g(\bar{X}_i, \mathcal{D})]$  over the space of various random choices of training data sets  $\mathcal{D}$ . Therefore, variance is a manifestation of *inconsistent behavior* by the algorithm over the space of different random choices of training data sets in which the same point receives very different scores across different choices of training data sets.. This phenomenon is caused by the algorithm adjusting too much to the specific nuances of a data set, and is also sometimes referred to as *overfitting*.

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  $k$ -nearest neighbor detector would determine the  $k$  closest points in the training data for any point  $\bar{X}_i$  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<sup>2</sup> the basic conclusions. Let  $\mathcal{D}$  be the training data, and  $\bar{X}_1 \dots \bar{X}_n$  be a set of test points whose (hypothetically ideal but unobserved) outlier scores are  $y_1 \dots y_n$ . It is assumed that these out-of-sample test points remain fixed over different instantiations of the training data  $\mathcal{D}$ , so that one can measure statistical quantities such as the score variance. We use an unsupervised outlier detection algorithm that

---

<sup>2</sup>It is noteworthy that the most popular outlier detectors are based on distance-based methods. These detectors are lazy learners in which the test point is itself never included among the  $k$ -nearest neighbors at prediction time. Therefore, these learners are essentially out-of-sample methods because they do not include the test point within the model (albeit in a lazy way).

uses the function  $g(\cdot, \cdot)$  to *estimate* these scores. Therefore, the resulting scores of  $\bar{X}_1 \dots \bar{X}_n$  using the training data  $\mathcal{D}$  are  $g(\bar{X}_1, \mathcal{D}) \dots g(\bar{X}_n, \mathcal{D})$ , 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:

$$MSE = \frac{1}{n} \sum_{i=1}^n \{y_i - g(\bar{X}_i, \mathcal{D})\}^2 \quad (2.3)$$

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

$$E[MSE] = \frac{1}{n} \sum_{i=1}^n E[\{y_i - g(\bar{X}_i, \mathcal{D})\}^2] \quad (2.4)$$

The different realizations of the training data  $\mathcal{D}$  can be constructed using any crisply defined random process. In the traditional view of the bias-variance trade-off, one might assume that the data set  $\mathcal{D}$  is generated by a hidden process that draws it from a true distribution. The basic idea is that *only one instance of a finite data set* can be collected by the entity performing the analysis, and there will be some variability in the results because of this limitation. This variability will also lead to some loss in the accuracy over a setting where the entity actually had access to the distribution from which the data was generated. To the entity that is performing the analysis, this variability is hidden because they have only one instance of the finite data set. Other unconventional interpretations of the bias-variance trade-off are also possible. 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. In this alternative interpretation, the expected values of the *MSE* is computed over different instantiations of the random process extracting  $\mathcal{D}$  from  $\mathcal{D}_0$ . Finally, one might even view the randomized process of extracting  $\mathcal{D}$  from  $\mathcal{D}_0$  as a part of the base detector. This will yield a randomized *base detector*  $g(\bar{X}_i, \mathcal{D}_0)$ , but a fixed data set  $\mathcal{D}_0$ . Therefore, the random process is now defined with respect to the randomization in base detector, rather than the training data selection process.

These different interpretations will provide different bias-variance decompositions of the same (or almost the same) MSE. We will provide specific examples of the different types of decomposition in a Sect. 2.4 with synthetic simulations. It is important to define the underlying random process clearly in order to properly analyze the effectiveness of a particular ensemble method. Note that even though the training data  $\mathcal{D}$  might have different instantiations because it is generated by a random process, the test points  $\bar{X}_1 \dots \bar{X}_n$  always remain fixed over all instantiations of the random process. This is the reason that we chose to demarcate the training and test data; it allows us to evaluate the effects of changing the training data (with a random process) on the same set of test points. If the predictions of the same test points vary significantly over various instantiations of the random process, we say that the model has high *variance*. Note that high variance will increase the overall

error even if the prediction of the test point is accurate *in expectation*. On the other hand, if the expected prediction of each test point is inaccurate, we say that the model has high *bias*. The basic idea is to decompose the error of the classifier into these two components. This type of decomposition provides the intuition needed to design algorithms that can reduce error by reducing one of these components.

The term in the bracket on the right-hand side of Eq. 2.4 can be re-written as follows:

$$E[MSE] = \frac{1}{n} \sum_{i=1}^n E[\{(y_i - f(\bar{X}_i)) + (f(\bar{X}_i) - g(\bar{X}_i, \mathcal{D}))\}^2] \quad (2.5)$$

Note that we can set  $(y_i - f(\bar{X}_i))$  on the right-hand side of aforementioned equation to 0 because of Eq. 2.1. Therefore, the following can be shown:

$$E[MSE] = \frac{1}{n} \sum_{i=1}^n E[\{f(\bar{X}_i) - g(\bar{X}_i, \mathcal{D})\}^2] \quad (2.6)$$

This right-hand side can be further decomposed by adding and subtracting  $E[g(\bar{X}_i, \mathcal{D})]$  within the squared term:

$$\begin{aligned} E[MSE] &= \frac{1}{n} \sum_{i=1}^n E[\{f(\bar{X}_i) - E[g(\bar{X}_i, \mathcal{D})]\}^2] \\ &\quad + \frac{2}{n} \sum_{i=1}^n \{f(\bar{X}_i) - E[g(\bar{X}_i, \mathcal{D})]\} \{E[g(\bar{X}_i, \mathcal{D})] - E[g(\bar{X}_i, \mathcal{D})]\} \\ &\quad + \frac{1}{n} \sum_{i=1}^n E[\{E[g(\bar{X}_i, \mathcal{D})] - g(\bar{X}_i, \mathcal{D})\}^2] \end{aligned}$$

The second term on the right-hand side of the aforementioned expression evaluates to 0. Therefore, we have:

$$\begin{aligned} E[MSE] &= \frac{1}{n} \sum_{i=1}^n E[\{f(\bar{X}_i) - E[g(\bar{X}_i, \mathcal{D})]\}^2] + \frac{1}{n} \sum_{i=1}^n E[\{E[g(\bar{X}_i, \mathcal{D})] - g(\bar{X}_i, \mathcal{D})\}^2] \\ &= \frac{1}{n} \sum_{i=1}^n \{f(\bar{X}_i) - E[g(\bar{X}_i, \mathcal{D})]\}^2 + \frac{1}{n} \sum_{i=1}^n E[\{E[g(\bar{X}_i, \mathcal{D})] - g(\bar{X}_i, \mathcal{D})\}^2] \end{aligned}$$

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

$$E[MSE] = \text{Bias}^2 + \text{Variance} \quad (2.7)$$

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  $g(\bar{X}_i, \mathcal{D})$  but also to the random process used to create the training data sets  $\mathcal{D}$ . Although this random process is generally assumed to be that of selecting a training data set from a base distribution, it could, in principle, be any random process such as the randomized algorithmic choices inside the base detector. The second view is non-traditional, but is more helpful in explaining the performance of certain types of outlier detectors.

A second issue is about the nature of the assumptions on the scores that were used at the very beginning of the analysis. 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  $g(\bar{X}_i, \mathcal{D})$  and the oracle  $f(\bar{X}_i)$  directly output the relative ranks in  $(0, 1)$ . *In other words, the above arguments are general, and they are not specific to the use of any particular outlier detector or assuming that outlier detectors are density estimators.*

### 2.2.1 Relationship of Ensemble Analysis to Bias-Variance Trade-Off

Ensemble analysis is a way of combining different models in order to ensure that the bias-variance tradeoff is optimized. In general, one can view the output of a base detector  $g(\bar{X}, \mathcal{D})$  as a random variable, depending on a random process over either the selection of the base data  $\mathcal{D}$ , or the construction of the detector  $g(\cdot, \cdot)$  itself, which might be randomized. The overall mean-squared error of this random variable is reduced with respect to the unknown oracle output  $f(\bar{X})$  by the ensemble process. This is achieved in two ways:

1. *Reducing bias:* Some methods such as boosting reduce bias in classification by using an ensemble combination of highly biased detectors. The design of detectors is based on the performance results of earlier instantiations of the detector in order to encourage specific types of bias performance in various components. The final combination is also carefully designed in a weighted way to gain the maximum advantage in terms of overall bias performance. However, it is generally much harder to reduce bias in outlier ensembles because of the absence of ground truth. Nevertheless, some methods have been designed to heuristically reduce bias in the context of outlier ensemble analysis [25, 26, 28].
2. *Reducing variance:* Methods such as bagging, bragging, wagging, and subagging (subsampling) [9–11], can be used to reduce the model-specific variance in classification. In this context, most classification methods generalize *directly* to outlier ensembles. In most of these methods the final ensemble score is computed as an average of the scores of various detectors. The basic idea is that the average



of a set of random variables has lower variance. In a sense, many of the variance reduction methods like bagging try to roughly simulate the process of drawing the data repeatedly from a base distribution. We will explain this point in greater detail in later chapters.

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 trade-off does not rely on such an availability *to the base algorithm*. None of the steps in the aforementioned computation of mean-squared error rely on the need for  $g(\bar{X}_i, \mathcal{D})$  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.

### 2.2.2 Out-of-Sample Issues

It is noteworthy that the test points  $\bar{X}_1 \dots \bar{X}_n$  are cleanly separated from the training data  $\mathcal{D}$  in the aforementioned analysis, and are therefore out-of-sample with respect to  $\mathcal{D}$ . Note that the random process varies the training data sets over different instantiations but the same fixed set  $\bar{X}_1 \dots \bar{X}_n$  of test points is used for each instantiation of the training data. Even in classification, the bias-variance trade-off is always understood in terms of the performance of the detector on out-of-sample test points that are fixed over the various instantiations of the training data.

However, in outlier detection, one typically does not distinguish between the training and test data. A natural question, therefore, arises as to whether this difference can affect the applicability of the bias-variance trade-off. We argue that even when the training data is the same as the test data, the bias-variance trade-off still holds approximately, as long as a *leave-one-out* methodology is used to construct the outlier scores. The leave-one-out methodology means that, when scoring a test point  $\bar{X} \in \mathcal{D}$ , one uses only  $\mathcal{D} - \{\bar{X}\}$  to construct the model. Such an approach is common in outlier detection settings, especially when instance-based methods are used. For example, in a  $k$ -nearest neighbor outlier detector or LOF detector, one typically does not include the data point itself, while computing the  $k$ -nearest neighbors of a data point. In other words, the outlier scores are almost always determined using a leave-one-out methodology.

As a result, the score of each point is computed in out-of-sample fashion, although each test point is drawn from the same data set  $\mathcal{D}$ . The leave-one-out methodology is a special case of the cross-validation methodology in classification, in which the data is divided into several folds, and one fold is classified using the remaining folds (as the training data set). In the particular case of leave-one-out, each fold contains exactly one data point, which is viewed as an extreme case of cross-validation. The cross-validation methodology is known to estimate the bias and variance characteristics of the out-of-sample setting very well, especially when the number of folds is large

(as in the extreme case of leave-one-out). Although there are tiny differences among the training data sets for various test points, the differences are small enough to provide an excellent approximation of the bias-variance characteristics of the out-of-sample setting.

### 2.2.3 *Understanding How Ensemble Analysis Works*

The bias-variance trade-off is defined in terms of a random process that creates the different training data sets. Note that the definition of the random process is crucial because the bias and variance are statistical quantities derived from this random process. In fact, for the same algorithm, one might use different random processes to describe it. Correspondingly, the error of the classifier will be decomposed into the bias and the variance in many different ways depending on the random process that is used. Furthermore, the overall error is also different depending on whether one assumes the availability of the base distribution or not. Traditionally, the bias-variance trade-off is understood from the perspective of sampling from a true distribution. In such cases, the errors are computed with respect to the availability of infinite data, and therefore the effect of finite size of the data is included in the error. In other types of model-centric random processes, the availability of the base distribution is not assumed, and therefore the overall error is lower (since it does not include the portion caused by the finiteness of the data). In order to explain this point, we will use a specific example.

Consider a mortgage application in which three banks collect data about the transactions of various customers to make predictions about which (outlier) customers are the ones most likely to default on their mortgage by using<sup>3</sup> outlier analysis. The banks collect different types of data about the customers such as their demographics, their past payment history, their salary, assets, and so on. Therefore, each bank has its own set of training data which might be different. It is assumed that the training data of each bank is drawn from the same base distribution, although each bank receives a different instantiation of the training data. Furthermore, the banks see only their own instantiation of the training data, and they have no access to each other's instantiations. Therefore, even though there is an inherent variance in the output over these instantiations (even if all banks use the same detector), the banks are unable to see each other's data sets or results to fully appreciate the nature of this variance.

Consider a setting in which each of the three banks receives a mortgage application from John. Therefore, John is a test point for which each bank needs to compute the outlier score using the training data. Note that the training data across the three banks are different, whereas the test point (John) is the same. This is the general assumption in the bias-variance setting, where we compute the bias and variance on the same

---

<sup>3</sup>In practice, such unsupervised methods are never used in such real-life scenarios. This example is only for illustrative purposes in order to provide a concrete example of the workings of the bias-variance trade-off.

set of test points using training data, which are generated by different instantiations of the random process. For example, each bank could apply a  $k$ -nearest neighbor outlier detector on its respective training data set to compute the outlier score for John. Furthermore, let us assume for the purpose of argument that each bank uses exactly the same value of  $k$  to execute the algorithm. In other words, the  $k$ -nearest neighbor distance of John is computed with respect to its training data set to report an outlier score. Clearly, each bank would receive a different outlier score for John because of the difference in their training data sets. This difference corresponds to the *variance* in the algorithm over different choices of training data sets. For *theoretical* purposes, it is assumed that each bank uses the same random process to draw the training data from the same base distribution. In *practice*, the banks use some data collection mechanism to create the training data sets, and therefore the assumption of drawing from a base distribution is simply a (hidden) theoretical assumption for the purposes of analysis. The basic idea of variance in the context of a hidden process of generating the training data sets is illustrated in Fig. 2.1. It is appropriate to consider this variance as “hidden” in real settings, because each bank would have only one instance of the training data, and may not notice the fact that some of the error in their computation of John’s scores is explained by the variability in John’s scores by other banks. After all, if all banks get very different scores for John with their data sets, at least some of them are very wrong. In variance-reduction ensemble methods, the goal is to minimize this *hidden* variability across the banks, with each entity

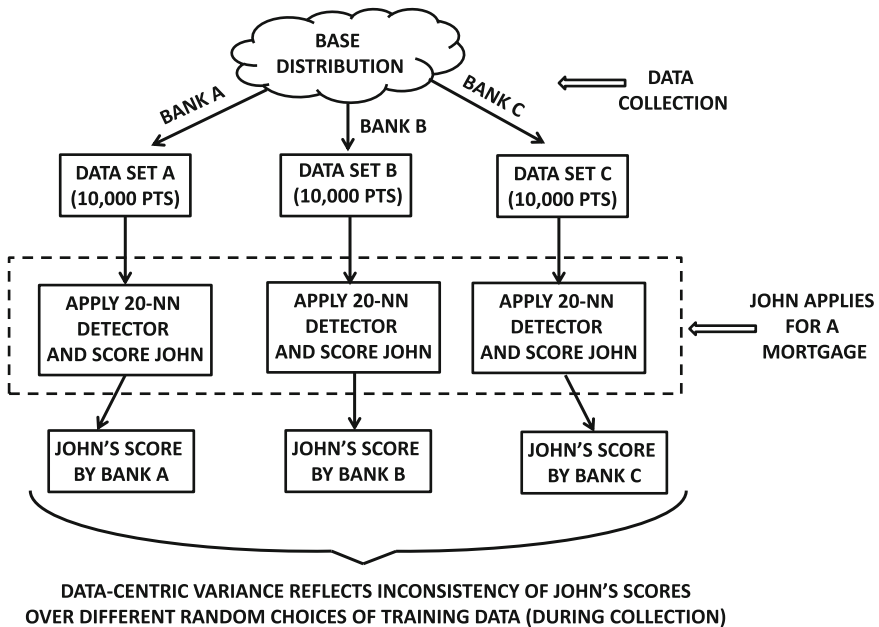
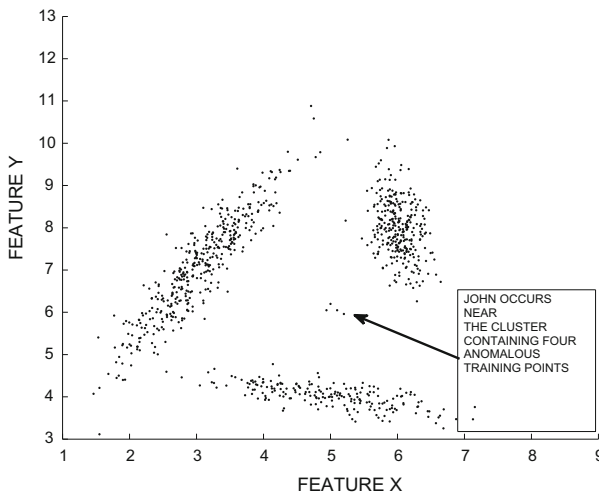


Fig. 2.1 Hidden variance caused by finite data set size

using only their *local* instance of the training data (assuming that each entity used the ensemble on its instance).

The variance in John's scores depends on the size of the training data sets are drawn. One can view the available data as a finite resource that affects the variance in the results. Smaller data sets have a negative impact on the accuracy of the approach because of larger variance; after all, the variance is a component of the mean-squared error. For example, if each bank draws a training data of the same small size from the base distribution (i.e., collects a smaller training data set), the variance of their scores on John would be larger. On the other hand, if each bank decides to use a larger size of the training data, then the variance will be much smaller in the scores of John. In general, when the variance is high, the quality of the scores obtained for John will be lower because the variance is one of the components of the error. For example, for bank A, its contribution to the variance is proportional to  $\{g(\text{John}, \mathcal{D}^A) - E[g(\text{John}, \mathcal{D})]\}^2$ . Here,  $\mathcal{D}^A$  represents the training data of bank A. The expected score  $E[g(\text{John}, \mathcal{D})]$  can be (very roughly) estimated as John's average score over the three different banks. The difference between this expected score and the ground-truth score yields the bias performance. Unfortunately, in unsupervised settings, this ground-truth is usually not available because of which the bias performance remains purely a theoretical construct.

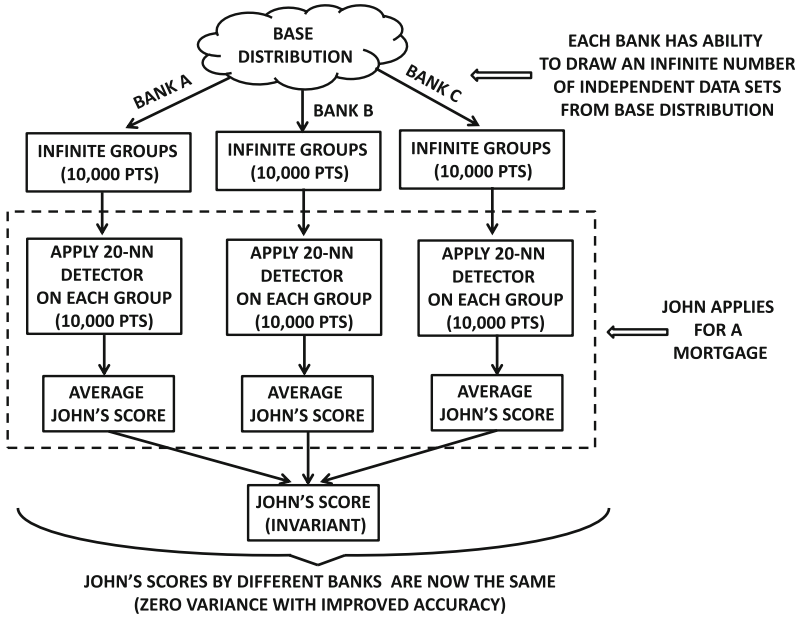
The bias often has a significant impact on the quality of the scores obtained for John. In unsupervised problems, the bias is the most unpredictable part of the performance that cannot be easily controlled. Consider a setting in which John occurs together with a small cluster of similar anomalies. This example is illustrated in Fig. 2.2 where 4 outliers occur together in the vicinity of John. Such scenarios are quite common in real settings; for example a small percentage of mortgage defaulters



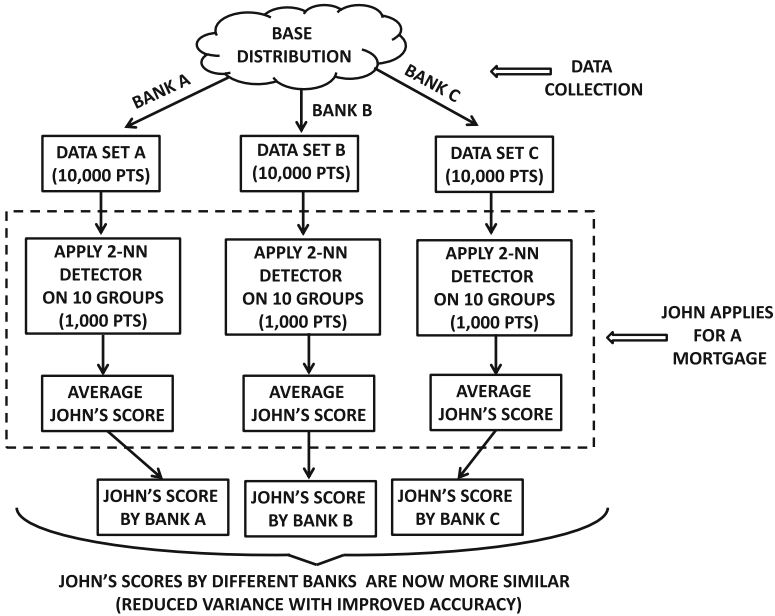
**Fig. 2.2** Effects of parameter choice on bias

might exhibit their anomalous behavior for the same underlying causes (e.g., low salary and high indebtedness). The training data for each bank will typically contain this small cluster, and therefore the choice of the value of  $k$  is important for the  $k$ -nearest neighbor detector. If each bank used a small value of  $k$  such as 1, then they would *consistently* obtain incorrect outlier scores for John. This is because the *expected* score  $E[g(\text{John}, \mathcal{D})]$  of the 1-nearest neighbor detector no longer reflects  $f(\text{John})$ , when the value of  $k$  is set to 1. Given that the training data sets (of size 10,000) of all banks are drawn from the same base distribution, all of them are likely to contain a small number of training points in the anomalous region populated by John. Therefore, even though John should be considered an anomaly, it will typically not be reflected as such with the use of  $k = 1$ . This portion of the error  $\{f(\text{John}) - E[g(\text{John}, \mathcal{D})]\}^2$  corresponds to the (squared) bias. Note that the bias heavily depends on the data distribution at hand. For example, in the particular case of Fig. 2.2, the use of a small value of  $k$  leads to high bias. It is also possible (and relatively easy) to construct data distributions in which small values of  $k$  lead to less bias. For example, in a data distribution with randomly distributed (but isolated) outliers, small values of  $k$  would be more effective. In a data distribution with anomalous clusters of varying sizes, different choices of  $k$  would exhibit better bias on different points. This aspect of outlier detection is very challenging because it is often difficult to tune outlier ensemble algorithms to reduce the bias. Furthermore, it is often more difficult to predict the bias trends in an unsupervised problem like outlier detection as compared to classification. When a  $k$ -nearest neighbor classifier is used in the supervised setting, the bias generally increases with the value of  $k$  and the variance generally reduces. However, in outlier detection (with a  $k$ -nearest neighbor detector), the bias could either increase or decrease with  $k$  but the variance almost always reduces with increasing  $k$ . It is noteworthy that it is not always necessary for the bias and variance trends to be opposite one another even in classification, although such a trade-off is *often* observed because of better ability to supervise the tuning of the algorithm towards a pareto-optimal frontier of the bias-variance trade-off.

How can one reduce the variance in the scores? Consider an ideal *theoretical* setting, in which each bank could draw as many training data sets as they wanted from the base distribution. In such a case, each bank could repeat the process of drawing training data sets over an infinite number of trials, determine the score for John over each training data set, and average John's scores over these different computations. The result would be that all banks would obtain exactly the same score for John, and the quality of the score would also be better because the variance component in the error has dropped to 0. This scenario is shown in Fig. 2.3a. However, this is impossible to achieve in practice, because each bank only has *one* instantiation of a finite data set, and therefore one cannot fully simulate the draws from a base distribution, which is an infinite resource of data. The finite nature of the resource (training data) ensures that some portion of the variance will always be irreducible. Therefore, we somehow need a way to make use of this finite resource (data) as *efficiently* as possible to minimize the variance. For example, running the detector once on a data set containing 10,000 points is not the most efficient way of reducing variance for the resource of 10,000 points. Very often, the quality of the scores



(a) Reducing variance to 0 with infinite data resources



(b) Reducing variance partially with finite data resources

Fig. 2.3 Reducing variance in data-centric ensembles

obtained from using a 20-NN detector on 10,000 points are only marginally better than those obtained using a 2-NN detector on 1,000 points, and one does not gain much from increasing the size of the data by 10 times.

In order to reduce the variance more *efficiently*, we need multiple instantiations of the training data set, which are derivatives of the original training data instance and then average the scores from the different instantiations *at the very end*. By doing so, one is roughly trying to simulate the process of drawing the scores from the base distribution by using the finite resource (data set) available to us. To achieve this goal, one can divide the data set of 10,000 points into 10 equal parts and then compute the outlier score of John with respect to each of these 1,000 points. Note that the value of  $k$  can be adjusted to the same relative value to ensure greater similarity in the two settings. For example, if we use  $k = 20$  with 10,000 points, we can use  $k = 2$  with 1,000 points. By making this adjustment, the bias performance is roughly similar in the two cases for an exact  $k$ -NN detector. By averaging John's outlier score across the 10 runs, the variance is greatly reduced, and the error closely reflects the bias of using  $k = 2$  on 1000 points (or  $k = 20$  on 10,000 points). Of course, the variance of an outlier detector with  $k = 2$  on 1,000 sampled training points is greater than that on an outlier detector with  $k = 20$  on 10,000 sampled training points. However, using averaging across 10 buckets of 1,000 points is a more efficient way of reducing variance rather than simply increasing the size of the base data to 10,000 points. As a result, higher-quality results will generally be obtained by the bucketing approach because the variance of John's score will be smaller. For example, if all three banks used this approach to determine John's outlier score, their scores will be more similar to one another with the bucketing approach. This is because they have reduced the variance component of their scores. This scenario is illustrated in Fig. 2.3b. Interestingly, this example is a simple variant of a well-known technique in classification referred to as subagging [9–11], and it provides a simple idea of how a *variance-reduction* scheme works in ensemble analysis. The idea is to use the finite data set available to us in the most efficient way possible to reduce variability in the scores caused by the finite nature of the data set. It is noteworthy, that unlike the case of Fig. 2.3a, some part of the variance is irreducible in Fig. 2.3b because we only have access to a finite resource. It is also important to note that this type of simulation is imperfect because it improves the accuracy in the vast majority of the cases, but can also occasionally fail in some circumstances. This example provides an understanding of the type of tricks that one commonly uses in order to gain accuracy improvements. Since variance reduction is more common in outlier ensemble analysis, as compared to bias reduction, much of our discussion will be based on this aspect.

## 2.2.4 Data-Centric View Versus Model-Centric View

The aforementioned discussion of the bias-variance trade-off is designed from the perspective of a random process for choosing the training data set  $\mathcal{D}$ . In other words, the expected values in the bias term  $\frac{1}{n}[f(\bar{X}_i) - E[g(\bar{X}_i, \mathcal{D})]]^2$  and the variance term  $\frac{1}{n} \sum_{i=1}^n E[\{E[g(\bar{X}_i, \mathcal{D})] - g(\bar{X}_i, \mathcal{D})\}^2]$  are both computed over various random choices of training data sets. The basic idea here is that the training data set is chosen from a true distribution by a random process that is hidden from us. This hidden process induces variability in the scores. Furthermore, we have only one instantiation of this hidden process, because the analyst (e.g., the bank in the previous section) has only one instantiation of the data set, and one must reduce the variability in the scores from this single instantiation. This is a more challenging setting than a case in which an infinite resource of data, such as the base distribution, is available to each bank. In the example in the previous section, all three banks were able to make their outlier scores more similar to one another by using ensembles on their *own* data sets (i.e., without using each other's data sets) with the use of draws from the base distribution. In the process, the quality of John's scores typically improves for each of the three banks because the variance component in the error has been reduced.

A common way in which the variability in the scores is reduced by many ensemble analysis methods, is by using a random process to construct randomized derivatives of the data set, and then averaging the scores from these different data sets. One can view each derivative as a sample from the base distribution, although it does not represent a theoretically ideal simulation because there will be correlations and overlaps among these derivative data sets, which would not have occurred if they had truly been drawn directly from the base distribution. This is the reason that ensemble analysis only provides imperfect simulations that almost always improve accuracy, but are not guaranteed to do so.

In our earlier example with the banks, the process of dividing the training data set into ten segments and then averaging the scores reduces the variability of the output. A theoretical explanation of this (imperfect) variance reduction process is provided in Chap. 3. However, from the perspective of bias-variance theory and ensemble analysis, this random process need not be applied to the training data set  $\mathcal{D}$ . The randomness could easily be directly injected into the detector  $g(\bar{X}_i, \mathcal{D})$  rather than the data sets. This leads to a model-centric view of the bias-variance trade-off. We emphasize that this view of bias-variance is unconventional, and we have not seen it discussed elsewhere. The traditional random process of sampling from a base distribution is excellent for explaining certain ensemble methods like bagging and subsampling, but its often not very good at explaining many other types of variance reduction methods in which the detector is itself randomized.

In practice, there are several ways in which one can use the bias-variance trade-off from the perspective of an ensemble method:

1. Consider an ensemble method like subsampling in which the training data sets from drawn from the same base data set  $\mathcal{D}_0$ . How should one view the random process for analyzing the bias-variance trade-off? In practice, one would need to



break up the random process into two separate steps. The first random process, which is hidden, produces a training data set  $\mathcal{D}_0$  from an unknown base distribution. The second random process of ensembling produces different choices of training data sets  $\mathcal{D}$ , each of which is a subset of  $\mathcal{D}_0$ . This is the scenario discussed in the previous section. This is equivalent to saying that each data set  $\mathcal{D}$  is directly selected from the (hidden) base distribution, although different instantiations of the data might have overlaps because of the dependencies in the sampling process. Therefore, the ensemble method simulates the process of generating training data sets from the base distribution although the simulation is imperfect because of dependencies among the base detectors caused by the finiteness of the resource with which we are attempting to perform the simulation.

For methods like subsampling there is an alternative way in which one can view the random process for the bias-variance trade-off. For example, one can perform bias-variance analysis under the assumption that the random process generates the training data directly from  $\mathcal{D}_0$ , and simply omit the first step of sampling  $\mathcal{D}_0$  from the base distribution. The use of such a random process results in dividing the same error into bias and variance in a different way as the case in which we assume the existence of a base distribution. Furthermore, the overall error is also different because we no longer have the variability of drawing  $\mathcal{D}_0$  from the base distribution. We will provide a better understanding of these decompositions later in this chapter. The main point to keep in mind is that the bias and variance will depend not only on the choice of the detector  $g(\bar{X}_i, \mathcal{D})$  but also on the random process to construct the training data sets  $\mathcal{D}$ . Traditionally, the selection of  $\mathcal{D}_0$  from a base distribution is always assumed in order to capture the variance of the hidden process that generates a finite data set. Note that this type of randomized process is relevant to the data-centered ensembles discussed in Chap. 1.

2. The random process injects randomness within the detector  $g(\bar{X}_i, \mathcal{D})$  but the data set  $\mathcal{D}$  is fixed. For example, while using a  $k$ -nearest neighbor detector, one might simply choose the value of  $k$  randomly from a range. Therefore, the bias and variance of a randomized detector is defined by its randomized design choices over a *fixed* data set. In such a case, it is important to note that the expectation  $E[MSE]$  of bias-variance theory is no longer over the randomized choices of training data, but over the randomized choices in model design. For example, a user might not be certain over the value of  $k$  to use, and might guess the choice of  $k$ , which is virtually equivalent to making a random choice within a range. By modeling  $k$  to be drawn randomly from a range of values, the bias-variance decomposition provides a model-centric variability, which is specific to parameter choice. However, by ensembling over different values of  $k$ , one is able to reduce this randomness.

This second form of the bias-variance trade-off is unconventional, but it is more useful for the analysis of model-centered ensembles discussed in Chap. 1. The key point to understand is that the bias-variance trade-off is designed with respect to a random process, and this random process can be different, depending on the kind of ensemble algorithm one is looking at. Furthermore, this form of the bias-variance trade-off is more general because data-centered ensembles can be

considered special cases of model-centered ensembles. For example, the data selection process in bagging can be considered a part of the randomized algorithm  $g(\bar{X}, \mathcal{D}_0)$ , where  $\mathcal{D}_0$  is the fixed base data from which the points are selected. The randomized algorithm  $g(\bar{X}_i, \mathcal{D}_0)$  samples the points randomly with replacement, and therefore the data sampling process is part of the detector  $g$ . A data-centric view would be that the algorithm  $g$  is deterministic but the data set  $\mathcal{D}$  is selected randomly from  $\mathcal{D}_0$  in order to run  $g(\bar{X}_i, \mathcal{D})$ . Furthermore, one can assume that  $\mathcal{D}_0$  is itself selected from an unknown base distribution by a hidden process. This is equivalent to saying that the data set  $\mathcal{D}$  is directly selected from the (hidden) base distribution, although different instantiations of the data might have overlaps because of the dependencies in the sampling process. The data-centric view is more useful in methods like bagging (and its variants like subbagging/subsampling), which reduce the variance resulting from the *hidden* process. Therefore, for any given ensemble algorithm, it is important to properly select the random process that best explains its performance. The model-centric view is more useful in methods that reduce the uncertainty arising from model selection choices. An example is the choice of the parameter  $k$  in distance-based algorithms. In fact, different values of  $k$  may be more suitable for different data points, and the ensemble will often do better than the median performance over these different choices. Such methods cannot be explained with a data-centric view. We believe that one of the reasons that methods like random forests have not been properly explained [12] in supervised settings like classification is that the literature has generally taken an (inflexible) data-centric view to the bias-variance trade-off.

3. It is possible for the random process to choose both the detector  $g(\bar{X}_i, \mathcal{D})$  and the data set  $\mathcal{D}$ . For example, one might use different choices of the parameter  $k$  in a distance-based algorithm over different bags of the data.

Variance reduction methods can be explained very easily with the use of the bias-variance trade-off. The basic idea in all variance reduction algorithms follows the same framework:

1. Use a data-centric or model-centric random process to generate randomized outputs from various base detectors. For example, one might generate randomized versions of the data sets (subsets of points or dimensions), or one might generate randomized versions of the detector  $g(\bar{X}_i, \mathcal{D})$ . An example of the latter case is one in which we use random choices of the parameters of the algorithm.
2. Average the outputs of these base detectors to reduce variance. The basic idea is that the average of a set of random variables has lower variance than the individual variables. The variance is best reduced when the outputs of various detectors are uncorrelated with one another.

This basic idea is invariant across classification and outlier detection, and therefore virtually all variance-reduction ensembles from classification can be generalized easily to outlier detection. In particular, many of the natural ensemble methods like bagging [5], subbagging [10] and random forests [6, 18] have corresponding analogs

in outlier analysis. It is noteworthy that the generation of randomized variants of the detector has a detrimental effect on the bias, as compared to a fully optimized algorithm. For example, if one applied a detector to a subset of points or with a random choice of the parameters, one might not do as well (in bias performance) as compared to an algorithm with all the points or specific choices of the parameters. However, the variance reduction effects of averaging are often sufficient to compensate for the poorer bias performance of individual components in such settings. In fact, it is possible (and common) for the ensemble performance to be better than the *vast majority* of the base component detectors. This type of performance jump is, however, not guaranteed and it is sometimes also possible for the ensemble performance to degrade below the median base performance. Section 3.3 of Chap. 3 discusses the circumstances under which such methods can degrade.

Let us try to understand the effect of a model-centered ensemble on the bias-variance trade-off. The crucial point to understand is that the randomized process for model-centered ensembles is inherently different from the randomized process used in the case of data-centered ensembles. We restate the bias-variance trade-off introduced earlier in this chapter:

$$E[MSE] = \frac{1}{n} \sum_{i=1}^n \{f(\bar{X}_i) - E[g(\bar{X}_i, \mathcal{D})]\}^2 + \frac{1}{n} \sum_{i=1}^n E[\{E[g(\bar{X}_i, \mathcal{D})] - g(\bar{X}_i, \mathcal{D})\}^2] \quad (2.8)$$

In the original statement of the bias-variance trade-off, the expectation  $E[MSE]$  is *typically computed over different choices of training data sets drawn from a base distribution*. However, in the model-centric point of view, we assume that we have a set of  $m$  alternative models, and we select one of these  $m$  alternative *models* over the same data set. Therefore, the data set is fixed, whereas the model might be randomized. For example, the choice of the parameter  $k$  in a  $k$ -NN detector might provide one of these alternative models. In this case, the *expectation in the bias-variance trade-off is over the different choices of models*. We emphasize that this is an unconventional view of the bias-variance trade-off and is generally not discussed elsewhere in the literature. Although it is more general to think of a model-centric view of the bias-variance trade-off, the expectation in the traditional view of the bias-variance trade-off is usually computed over different choices of training data sets. However, a model-centric view of the bias-variance trade-off helps to explain many types of ensembles, which cannot be easily explained purely by using a data-centric view. The difference between the model-centric view and data-centric view of outlier ensembles roughly corresponds to the categorization of ensemble analysis into data-centric ensembles and model-centric ensembles [1].

Why is this approach to the bias-variance trade-off more general? This is because one can also understand the data-centric processes of selecting a randomized derivative of the data set as a special case of this setting. For example, in the case where John was scored on 10 different randomly drawn partitions of the training data sets, one can view the randomized process of creating 10 partitions of the data as a part of the model itself. Therefore the bias and variance is computed over this *random*

*process* of creating the 10 partitions, rather than over random process of drawing data from a true distribution. In such a case, the variance of John's score can be viewed as its expected variation over all possible groupings of the data over a fixed base training data (without taking into account the additional hidden random process of drawing the training data). The bias is defined by determining John's expectation score over all possible groupings, and then computing the difference between John's (unknown) ground-truth score and the expected score. The key here is that the expectation is *over the process of creating the different groupings* rather than the choice of the data set. This distinction is crucial because this different random process will have a different bias and a different variance. Although the mean-squared error will always be the same for a particular algorithm, this model-centric decomposition will provide a different view of the bias and variance, *because it is a model-centric view of the bias-variance trade-off*. In other words, one can decompose the error of a randomized detector into bias and variance in multiple ways, depending on the kind of random process that one is looking at. Any data-centric ensemble can be analyzed either from the perspective of a data-centric bias-variance trade-off, or a model-centric bias-variance trade-off. In the latter case, the process of extracting the training data is considered a part of the model. However, a model-centric ensemble can be analyzed only from the perspective of a model-centric bias-variance trade-off.

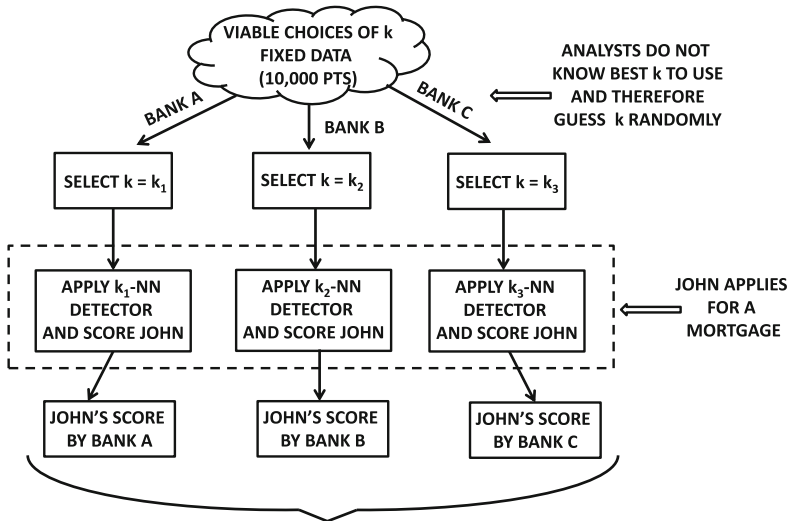
The model-centric view of the bias-variance trade-off can also handle settings that cannot be easily handled by the data-centric view. Just as data-centric ensembles are designed to address the uncertainty in choice of training data sets, model-centric ensembles are designed to address the uncertainty in the choice of models. The design choice of a model plays a crucial role in many problems like outlier detection. In unsupervised problems like outlier detection, there is significantly greater uncertainty in the design choices of the models, as compared to supervised problems like data classification. This is because one can use techniques like cross-validation in classification in order to make important choices, such as the optimal value of  $k$  in a  $k$ -nearest neighbor classifier. However, since the ground-truth is not available in problems like outlier detection, it is impossible to know the optimal value of the parameter  $k$ . Just as the choice of training data set is imposed on the analyst (and creates some hidden variance), the choice of such model parameters (or other design choices) creates uncertainty for the analyst. In this case, the analyst has greater control on selecting such parameters (as compared to training data determination), but may often set these values in an arbitrary way because of lack of guidance. Such arbitrary choices (implicitly) result in a kind of variance in the output of the detector because they might vary with the specific analyst, and one cannot easily view any of these choices as inherently better than the other in an unsupervised setting. In other words, all choices of the parameter  $k$  within a reasonable range are as good as random guessing, and the variability caused by this random guessing is not very different in principle than the variability caused by different random choices of training data sets.

It is noteworthy that in the data-centric view of outlier ensembles, design-choices (such as the parameter  $k$ ) affect the bias of the model *over the space of different randomly selected training data sets* (see Fig. 2.2). However, in the model-centric

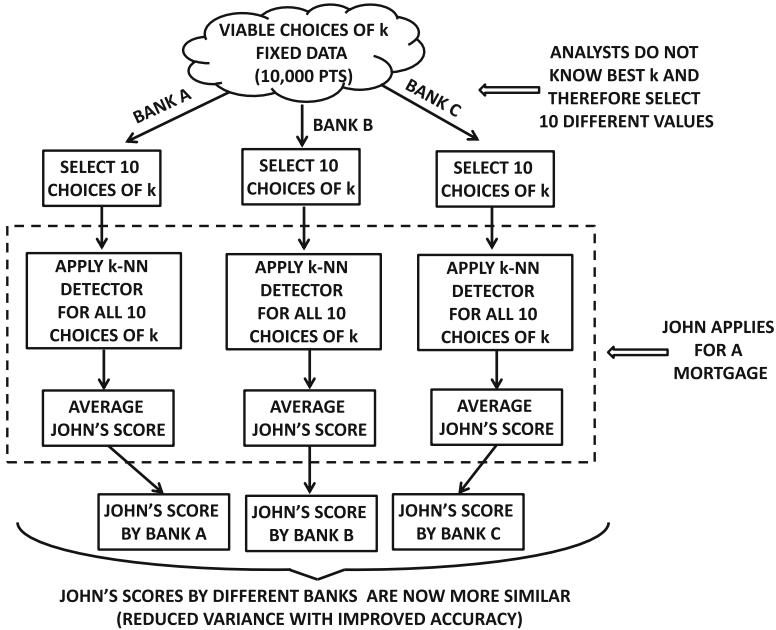
view, this variability in bias (caused by changing  $k$ ) is now viewed as a part of the variance of the randomized model-centric process of selecting  $k$ , which can be reduced with ensemble analysis. It is here that the model-centric view of the bias-variance trade-off is particularly useful.

In order to understand this point, let us revisit the problem in which John's mortgage application is scored by banks A, B, and C with the use of an outlier detector. Consider a setting, where the three banks have access to *exactly the same training data set* of 10,000 points. Note that this setting is already different from the previous case because we are no longer assuming that the different banks have different training data sets drawn from the same base distribution. However, in this case, the uncertainty is caused by the fact that none of the analysts seem to know what value of  $k$  to use for their  $k$ -nearest neighbor detector. For some data sets, a value of 5 might be effective, whereas for others a value of 100 might be effective. Therefore, there is significant uncertainty about the effect of choosing a particular value of  $k$ , particularly in the context of unsupervised problems in which the specific accuracy cannot be known even after the fact. As a result, the different analysts use different values of  $k$  for their respective detectors. The result of this approach is that they all obtain different results, and there is an inherent variability in their results caused by the specific choice of the parameter  $k$ . This variability is shown in the different outputs for John in Fig. 2.4a. However, it is possible for the analysts to run the detector over 10 different randomly chosen values of  $k$ , and average the performance of the detector over these choices to reduce the variance. As a result, John's scores from the three banks become more similar. Furthermore, the quality of the scores is improved because of reduced variability. This scenario is shown in Fig. 2.4b. It is important to note that in this model-centric view, the data set is assumed to be fixed, and the variability is caused because of uncertainty in the specific choice of the model. In the model-centric view, one is often improving the bias performance of individual models in the data-centric view by averaging over the variability in the bias over different randomized models. For example, a specific choice of  $k$  has a particular bias in the data-centric view; however this variability in bias over different choices of  $k$  in the data-centric view is converted to variance in the model-centric view. One can then reduce this aspect of the variance with the ensemble approach. In practical settings, this often means that one might be able to obtain better results with the ensemble scheme, compared to any particular value of  $k$ . For example, a value of  $k \leq 4$  is clearly suboptimal to discover John as an outlier. It may be possible that for some other test points, a value of  $k = 5$  may be too large and may therefore provide incorrect results. By ensembling over a "well-chosen" range of  $k$ , it is often possible to obtain better results than any specific value of the parameter (over all points). This is because of the *variability* in performance of different portions of the data over different values of  $k$ . Herein, lies the power of variance reduction in the model-centric setting. This principle is also related to the notion of reducing *representational bias* [13] of any specific model design by ensembling over different randomized models. We will discuss this issue in greater detail in Chap. 3.

The differences between data-centric and model-centric views of the bias-variance trade-off are shown in Fig. 2.5. The traditional view, which is the data-centric view,

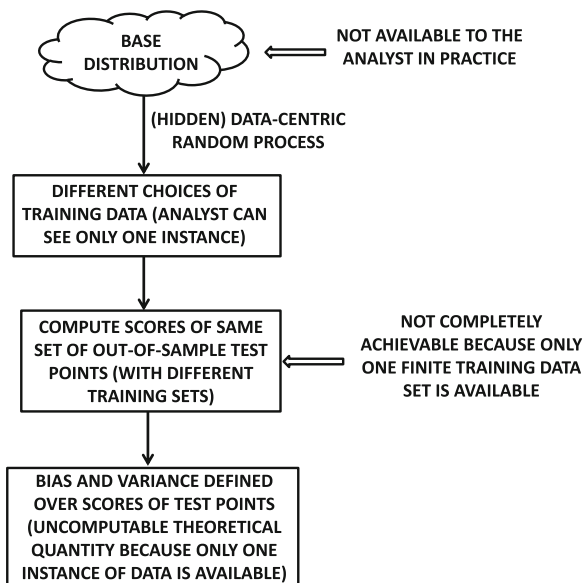


(a) Model-centric random process with high variance

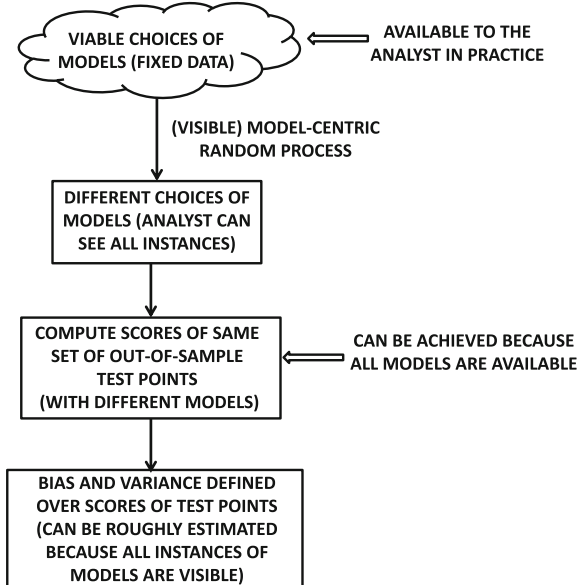


(b) Model-centric random process with low variance

**Fig. 2.4** Reducing variance in model-centric ensembles



(a) Conventional data-centric view (random process is over space of data sets)



(b) Unconventional model-centric view (random process is over space of models)

**Fig. 2.5** Different perspectives on the bias-variance trade-off

is shown in Fig. 2.5a. Here, the expectation  $E[MSE]$  (in the bias-variance equation) is computed *over different random choices of training data sets*. Since a particular analyst usually has access to only one training data set and does not have access to the base distribution, it is generally hard to estimate the bias and variance in this setting. In other words, the *finiteness* of the available data causes limitations in our ability to estimate the bias and variance of a particular model. One typically reduces variances in such settings by using methods like bagging, which can be (approximately) viewed as the process of drawing different training data sets from the same base distribution. By averaging the results from the different draws, one is able to effectively reduce variance, although the approximate process of performing the draws leads to significant constraints. A more unconventional view of the bias-variance trade-off, which is the model-centric view, is shown in Fig. 2.5b. Here, the expectation is computed *over different randomized choices of models* over the same training data set. In this case, the expectation is computed over different randomized choices of models. Note that this process is often directly controlled by the analyst, and (in most cases) it is not a hidden process outside the control of the analyst. Therefore, the analyst can actually run the model over different randomized choices of the model and estimate the bias and variance as follows:

1. The analyst can run the model over a very large number of trials and compute the expected score over each out-of-sample test instance. The bias of the model is the difference between the true (ground-truth) scores and the averaged scores. Of course, since one typically does not have the ground-truth available in problems like outlier detection, it may often not be possible to compute the bias in practice, except for some bench-marking settings in which ground-truths are set on the basis of some assumption.
2. The analyst can determine the variance in the scores over different test instances relatively easily. Therefore, the variance can be easily computed whether the ground-truth is available or not.

In unsupervised problems, it is hard to compute the bias in both model-centric and data-centric settings. However, it is much easier to estimate the variance in most model-centric settings, unless some part of the random process is hidden from the analyst. Furthermore, it is also generally much easier to develop variance-reduction algorithms in the unsupervised setting, as compared to bias-reduction algorithms.

## 2.3 Examples and Applications of the Bias-Variance Tradeoff

In the following, we provide a few examples of how the bias-variance trade-off is used in different settings related to classification, and the corresponding adaptations to outlier detection. These examples also show that many ideas from classification can be adapted easily to outlier detection. However, the adaptation is not always a simple matter because of the unsupervised nature of outlier detection. In general, we will see



that variance-reduction methods are much easier to adapt to the unsupervised settings, as compared to bias-reduction methods such as boosting. The main difference among these methods is in terms of how the combination is constructed in order to reduce either the bias or the variance. Correspondingly, the accuracy improvements of each of these methods can be explained with an appropriate definition of the random process over which the corresponding bias-variance trade-off is defined.

### 2.3.1 Bagging and Subsampling

Bagging and subsampling are well-known methods in classification for reducing variance [5, 6, 9–11]. The basic idea is to draw different training data sets from the same base data by sampling with or without replacement. The predictions over these different components are averaged to yield the final result. The basic idea in these methods is to reduce the variance with the averaging process. We describe these two approaches in some detail below, along with their corresponding effects on bias and variance:

1. *Bagging*: In bagging, samples of the data are drawn from a base data set with replacement, so that a bootstrapped sample is constructed. The bootstrapped sample typically has the same size as the original data set, although it is not essential to impose this restriction. Because of the fact that the sampling is performed with replacement, the sample will contain duplicates, and many points from the original data set will typically not be included. In particular, when a sample of size  $n$  is drawn from a data set of size  $n$ , the probability that a particular data point is not included is given by  $(1 - 1/n)^n \approx 1/e$ . A separate classification (or outlier detection) model is constructed on each of the bootstrapped samples, and the predictions across different samples are then averaged in order to create a final score. This basic idea of bagging is discussed in [1].

The basic idea in bagging is to reduce the variance of the prediction. Each individual detector has roughly similar bias characteristics as the original data. However, by averaging, it is possible to significantly reduce the variance of the prediction. As a result, the accuracy of the prediction is improved. The theoretical and intuitive arguments for bagging are very similar in the case of classification and outlier detection. A detailed discussion of bagging together with experimental results is provided in the variance reduction chapter (Chap. 3). To the best of our knowledge, this book provides the first detailed experimental results on bagging in the context of outlier detection.

2. *Subsampling*: Subsampling is a straightforward variation of bagging. The approach is also referred to as *subbagging*. In subsampling, samples of the data are constructed *without* replacement. Therefore, the sampled data set is much smaller than the original data set. Note that if we use the same algorithm parameters over a smaller data set, the subsampled data set is likely to have very different bias characteristics. Nevertheless, the variance reduction is often likely to more

significant because the individual ensemble components are more diverse. The basic idea of subsampling was proposed in [9–11] and it can be generalized trivially to outlier detection with virtually no changes. The first use of subsampling for outlier detection was proposed in the context of *edge* subsampling in the graph domain [3]. Subsampling methods were also sometimes used in the context of *efficiency improvements* [23], although the accuracy improvements were only limited with this specific implementation. Subsequent discussions on the use of subsampling for outlier detection with the use of distance-based detectors are provided in [4, 32]. However, the work in [32] provides an implementation of subsampling that has unpredictable bias-centric effects and also provides an incorrect theoretical explanation. The incorrectness of this reasoning was clarified in [4], and the theoretical foundations for outlier analysis/ensembles were also established in this work. These foundations were used to propose more accurate and reliable subsampling by varying the subsampling rate [4]. A detailed discussion of subsampling will also be provided in Chap. 3.

Note that the natural adaptation of the bagging family of techniques from classification to outlier detection has also been published in an earlier position paper [1] on outlier detection. The key idea of subsampling (as it applies to classification) should be credited to the original subsampling papers [9, 10]. Although subsampling methods were subsequently investigated in [3, 23, 32], a proper explanation of the effects of subsampling, as it applies to outlier detection, may be found in [4]. Further exposition of these effects are provided in greater detail in Chap. 3.

Both the methods of bagging and subsampling rely on a random process generating the ensemble components, and then using a combination method to reduce the variance of the final output. However, these principles apply to any approach for randomly perturbing a detector to improve the variance-reduction characteristics of the method. Bagging and subsampling methods are discussed in detail in Chap. 3.

### 2.3.2 Feature Bagging

Feature bagging is a method that is used commonly in classification [7, 8, 19, 24] to create individual ensemble methods with sufficient diversity. The averaging of the predictions of different ensemble members can reduce (model-centric) variance. Subsequently, a natural adaptation of the feature bagging method to outlier detection was proposed in [22].

The basic idea in feature bagging, as it applies to outlier detection, is to sample a number  $r$  between  $\lfloor d/2 \rfloor$  and  $d - 1$ , where  $d$  is the total number of dimensions in the data. Subsequently,  $r$  randomly chosen dimensions are sampled from the underlying data and the outlier detection algorithm is applied to this  $r$ -dimensional projection. Note that the individual ensemble components in feature bagging often have deteriorated (model-centric) bias characteristics because of the fact that dimensions are dropped from the data set. On the other hand, they are somewhat diverse, and

therefore variance can be reduced by the averaging process. The deteriorated bias characteristics of feature bagging are somewhat of a concern because they can sometimes affect the final ensemble performance as well. Nevertheless, in most cases, it has been shown that the use of feature bagging generally improves accuracy. This improvement in accuracy is attributed to variance reduction. However, one needs to be careful of using the right random process to describe the bias-variance trade-off of this ensemble method. In particular, feature bagging can be best explained with a model-centric random process, even though the approach seems to be a data-centric ensemble at first sight. Feature bagging methods are discussed in detail in Chap. 3. A proper theoretical explanation of feature bagging is also provided in the same chapter.

### 2.3.3 *Boosting*

The boosting method is used popularly in classification [14, 15], but it is harder to generalize to outlier detection. Boosting uses a combination of highly biased detectors, so that the final detector has less bias than the individual components. The basic idea is to create biased data sets in which the misclassified training examples are given greater weight. The basic assumption is that the errors in the misclassified examples are caused by instance-specific bias, and weighting them to a larger degree will result in a training model that will classify them correctly. This is achieved by using a base detector with low variance, so that most of the error is caused by the bias component. A weighted combination detector is created to combine the bias characteristics of various components to create a final detector, which has lower bias than its individual components.

An important observation about boosting is that it requires the computation of accuracy on the training examples. The accuracy computation of a classifier requires the comparison of the predictions with the ground truth. This can often be difficult in an unsupervised problem like outlier detection. Nevertheless, a number of heuristic methods can also be used in the context of the outlier detection problem. Such methods are discussed in Chap. 4.

## 2.4 Experimental Illustration of Bias-Variance Theory

In this section, we will provide an experimental illustration of bias-variance theory with the use of a number of synthetic data sets. We will also show the impact of using ensemble methods on the bias-variance analysis. Since variance reduction is particularly valuable in the context of bias-variance theory, much of our focus will be on the effect of ensemble methods on variance. In particular, we will show the following effects:

1. We show the effect of methods like subsampling on variance reduction.
2. We show the effect of finiteness of the data set on the limits of data-centric variance reduction.
3. We show the differences in the data-centric and model-centric view of the bias-variance trade-off.

These different insights set the stage for introducing the different ensemble methods discussed in subsequent chapters.

### 2.4.1 *Understanding the Effects of Ensembles on Data-Centric Bias and Variance*

In this section, we will study the effect of ensemble methods like subsampling on *data-centric* bias and variance. In this case, the assumption is that the training data sets are drawn from the same base distribution. The data-centric bias and variance are only theoretical quantities in real settings (which cannot be actually computed) because they are based on the variability of drawing training data sets from a base distribution. In practice, this base distribution is not available to the analyst, but only a *single finite instantiation* of the data set is available. This finite instantiation can be viewed as a finite resource that must be exploited as *efficiently* as possible to maximize the benefits of ensemble analysis. Knowing the base distribution is equivalent to having an infinite resource of data at one's disposal. Although the data-centric bias and variance are difficult to quantify in real settings (because of the finiteness of the data resource), we can still use a synthetic setting in which it is assumed that the base distribution is known. This synthetic setting can be used to show the effects of ensemble analysis on various measures of outlier detection accuracy, such as the rank-wise accuracy, the mean-squared error, bias, and variance. A preliminary version of these results is available in [4], although this expanded version provides significantly more insights in terms of data-centric and model-centric analysis.

In order to show the effects of ensemble analysis, we use some simulations with the subsampling method [3, 4, 9–11, 23, 32] discussed earlier in this chapter. This approach can be viewed as a variant of the example of scoring mortgage applications with averaged predictions on randomized partitions of the data. Instead of creating randomized partitions, we draw random subsamples of the training data, and then score each point with respect to the subsample whether that point is included in the subsample or not. The scores of a test point across the different subsamples are then averaged in order to provide the final outlier score of that point.

The subsampling approach has been used earlier for both classification and outlier detection. The use of subsampling for classification is discussed in [9–11]. The earliest accuracy-centric work on subsampling in outlier detection was done in the context of *edge* subsampling for graph data (for detecting edge outliers), and the approach was also used in the context of efficiency-centric improvements for outlier detection in multidimensional data [23]. Note that the former implicitly subsamples *entries*

in an adjacency matrix, whereas the latter subsamples *rows* in a multidimensional data matrix. The approach was also explored for nearest neighbor detectors like the average  $k$ -nearest neighbor detector [4] and the LOF method [4, 32]. The work in [4] already provides a number of experimental illustrations of bias-variance theory although it does not specifically decompose the error into the bias and variance components. This section will show some further simulations with synthetic data, which explain the nature of the bias-variance decomposition discussed in [4].

In the following, we will use some simple synthetic distributions to generate data sets. One advantage of using synthetic distributions is that we can explicitly test the effects of drawing truly independent training data sets from an infinite resource of data; these independent data sets can be used to properly characterize the bias and variance performance of training data sets drawn from a particular base distribution.

We used two 1-dimensional locally uniform distributions and a 2-dimensional distribution with clusters of uniformly distributed points. Consider a data set  $\mathcal{D}$  containing the points  $\bar{X}_1 \dots \bar{X}_n$ , with local probability densities  $f_1 \dots f_n$ , 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  $r_1 \dots r_n$ . 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-nearest neighbor distance than a data point in a dense region, then an inversion is assumed to have occurred. For each of the  $n \cdot (n - 1)/2$  pairs of points in the data set, we computed a non-inversion credit  $C(\bar{X}_i, \bar{X}_j)$  as follows:

$$C(\bar{X}_i, \bar{X}_j) = \begin{cases} 0 & f_i < f_j \text{ and } r_i < r_j \\ 0 & f_i > f_j \text{ and } r_i > r_j \\ 1 & f_i < f_j \text{ and } r_i > r_j \\ 1 & f_i > f_j \text{ and } r_i < r_j \\ 0.5 & f_i = f_j \text{ or } r_i = r_j \end{cases} \quad (2.9)$$

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

$$NI(\mathcal{D}) = \frac{\sum_{i < j} C(\bar{X}_i, \bar{X}_j)}{n(n - 1)/2} \quad (2.10)$$

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 value of  $NI(\mathcal{D})$  is 1. A value of 0.5 would be expected from a random detector. Therefore, the non-inversion credit provides an intuitive idea of how well a particular detector performs in a given setting.

Since our primary argument on the effectiveness of subsampling is based on variance reduction, 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-nearest neighbor distances) from various ensemble components are correlated. As a result, the variance reduction effects of averaging were curtailed. The problem is that the base data set is finite, and larger subsamples from a base data set always lead to correlated detectors. Even though one can view the process of drawing a subsample from the base data as equivalent to drawing the sample from the base distribution over a *single* sample, this is not true over multiple samples in which the finiteness of the base data comes into play and causes correlated samples. Correlated detectors generally have a negative effect on any form of variance reduction in ensemble analysis. Furthermore, one cannot meaningfully estimate the bias performance of a particular detector from a data-centric point of view, if the base distribution is unavailable. If the base distribution is available, one can use the approach of Fig. 2.1a to repeatedly draw training data sets and average the results to reduce the variance to 0. The remaining error reflects the bias-performance of the algorithm.

In this section, we simulate the scenario where the base distribution is available. In effect, the availability of the base distribution provides the resources of an infinite data set. One can study the effects of such infinite resources on a procedure such as subsampling to see how much one can improve the performance. In such a case, the results of any pair of subsamples (drawn from the base distribution) would be truly independent, and the full effect of variance reduction could be realized because of the infiniteness of this resource. The original base data  $\mathcal{D}$  is only used to test the outlier scores against each such generated model, whereas the training data sets are generated directly from the base distribution. We also study the limits of this variance reduction caused by the finite data size available in real settings. As we will show, a portion of the variance caused by training data variability is always irreducible in the setting, where a finite data set is used for variance reduction. Therefore, we generated two different variants of base detectors and ensembles:

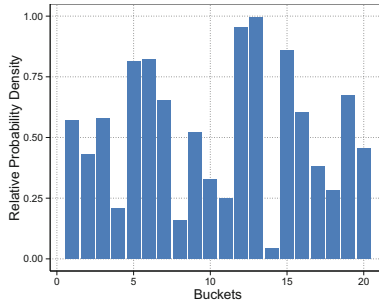
1. We constructed the base detectors by drawing subsamples from the original data set  $\mathcal{D}$ . This data set was also used as the test data set, but the 1-NN computation of each point in the test data  $\mathcal{D}$  was computed only on the subsample of  $\mathcal{D}$ . The average of the 1-NN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

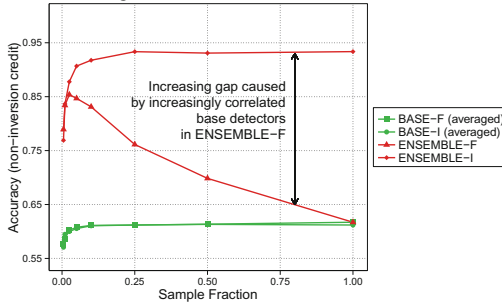
The results in this section used 300 trials. The accuracy of the base detector is computed by averaging the accuracy over each of these 300 instantiations, whereas the accuracy of the ensemble approach is computed using the averaged 1-NN score of the ensemble. We used 300 trials because the accuracy usually leveled out after this point, and not much advantage was obtained by further increasing the number of trials.

First, we used a data set  $\mathcal{D}$  containing 2000 points drawn from locally uniform distributions in a single dimension. The data distribution is shown in Fig. 2.6a. In this case, the data is distributed in 20 1-dimensional buckets. All 1-dimensional points in the  $i$ th 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 Fig. 2.6a. Therefore, the lower bars correspond to regions which are outlier regions in this 1-dimensional data, albeit uniformly distributed. The values on the  $Y$ -axis of Fig. 2.6a, are used as the ground-truth values of  $f_i$  in Eq. 2.9 for the corresponding data points in that bucket. The 1-NN distance is used as  $r_i$  in Eq. 2.9. 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 Fig. 2.6b. Note that the performance of both base detectors *improves with* the sampling rate, and no advantage was observed for smaller subsamples. 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. We repeated the same experiment with the use of 40 buckets instead of 20 and present the results in Fig. 2.6c, d. The results are very similar to the case of Fig. 2.6a, b. Note that some forms of the bias-variance trade-off [17] explicitly take this correlation into account. This form of the decomposition is referred to as the *bias-variance-covariance* decomposition.

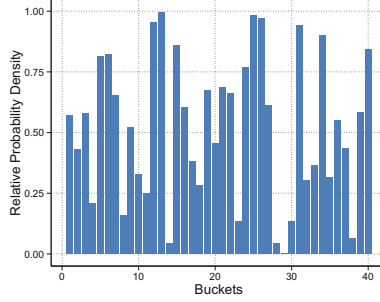
We also tested the effects with 2-dimensional locally uniform distributions of 2000 points. In this case, 30 clusters of uniformly distributed squares were generated, with lower-left corners chosen uniformly at random in  $(0, 1)$ . Each square had a side of length  $1/15$ . The relative number of points in each cluster was a uniform random variable in  $(0, 1)$ , and it represented the ground-truth value of  $f_i$  in Eq. 2.9. The corresponding scatter plot is shown in Fig. 2.7a. The corresponding effects on



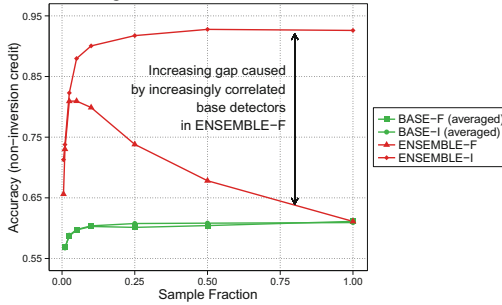
(a) 1D Histogram Distribution (20 Buckets)



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



(c) 1D Histogram Distribution (40 Buckets)

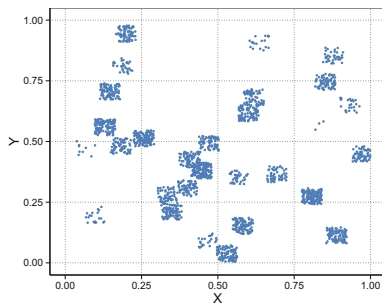


(d) Ensemble/Base Performance (1-dimensional Histogram - 40 Buckets)

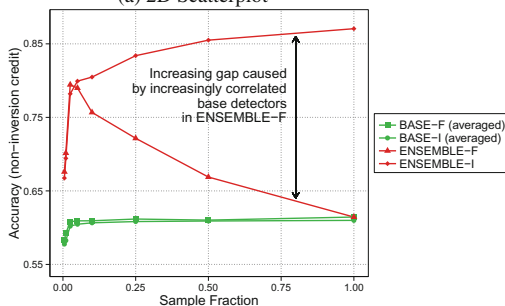
**Fig. 2.6** Effectiveness of base and ensemble on locally uniform data sets (Sampling “rates” for infinite data set are defined with respect to finite base data set  $\mathcal{D}$ )



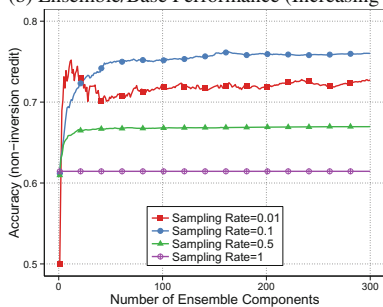
**Fig. 2.7** Effectiveness of base and ensemble on locally uniform data sets (Sampling “rates” for infinite data set are defined with respect to finite base data set  $\mathcal{D}$ )



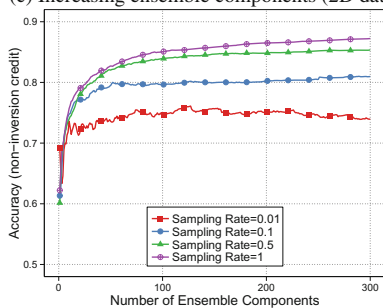
(a) 2D Scatterplot



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



(c) Increasing ensemble components (2D data set- ENSEMBLE-F)



(d) Increasing ensemble components (2D data set- ENSEMBLE-I)

the non-inversion credit with increasing subsample size are illustrated in Fig. 2.7b. As in the case of the 1-dimensional 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 Fig. 2.7c, d. The former (Fig. 2.7c) is for the case of the *ENSEMBLE-F* method with the 2d-distribution, whereas the latter is for the case of *ENSEMBLE-I* method with the 2d-distribution. 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 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.

#### 2.4.2 Experimental Examples of Bias-Variance Decomposition

The synthetic nature of the data sets allow us a way to show the bias-variance decomposition experimentally. In real settings, one can never construct ensemble performances like *ENSEMBLE-I* with a single instance of a finite data set. Although some sampling methods exist [20] to *estimate* the bias and variance experimentally for real data sets, we argue that such methods are too unreliable/approximate to provide any meaningful insights. This is in part because the base distribution of a real data set is unavailable and there are correlations among the results from different samples. However, for synthetic data sets, where one has access to the base distribution, it is possible to simulate the bias and variance performance very closely.

In the previous section, we used rank-centric measures for test the performance of *ENSEMBLE-I* and *ENSEMBLE-F*. In this section, we will study the more conventional *MSE* measure because it is relevant to the bias-variance decomposition. In order to compute the *MSE*, we do need to standardize both the ground-truth and the predicted scores for comparability. Therefore, just before the computation of the *MSE*, the ground-truth scores and the predicted scores (both for the base and the

ensembles) are standardized to zero mean and unit variance. In the following, we will assume that the size of the test data  $\mathcal{D}$  is  $n_0$ .

It is important to remember that the bias-variance decomposition depends on the *choice of the random process* over which the expected value  $E[MSE]$  is computed. Therefore, we will consider the following two ways of defining the random process:

1. *Random Process A (Data-centric)*: The base distribution is available to the analyst. We can describe the random process for *BASE-I* as follows. For a given test data set  $\mathcal{D}$  of size  $n_0$ , we repeatedly draw samples of size  $f \cdot n_0$  from the base distribution in order to compute the scores.

We can describe the random process of *BASE-F* as follows. The training data of size  $n_0$  is drawn from the base distribution and then a fraction  $f$  sample of size  $f \cdot n_0$  is subsampled from it to create the training data. Note that this process is equivalent to drawing a sample of size  $f \cdot n_0$  from the base-distribution, although there are overlaps among the samples drawn for a particular run of subsampling. However, the bias and variance are computed not over a particular run of subsampling but *over all possible draws* of the base data of size  $n_0$ . This is an important point, because it makes the bias-variance decomposition of *BASE-F* and *BASE-I* very similar. It is important to note that the variance of the prediction of a given test point needs to include the variance caused by initially selecting a particular base data set of size  $n_0$  from the distribution for subsampling. For example, if one drew a subsample of fraction  $f = 1$ , then the same subsample will be drawn every time from a particular base data set, but the variance of the prediction of a test point by *BASE-F* will still be non-zero because it includes the variance of drawing a training data set of size  $n_0$  from the base distribution. In this context, the variance of predicting each test point, when computed over a very large number of instantiations of the base data set (followed by subsampling in each case), is not very different between *BASE-F* and *BASE-I*. As we will see later, a portion of this variance of *BASE-F* is always irreducible in practice, because of correlations among subsampled base detectors. This irreducible variance is an artifact of the fact that an analyst has access to only a *single finite instance* of this data set, and there are fundamental limitations to the variance reduction process with this finite resource.

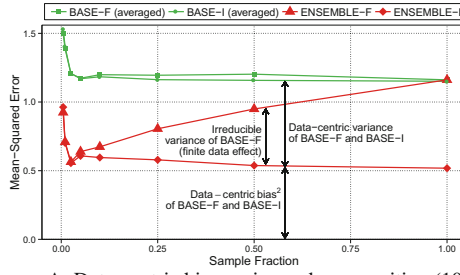
2. *Random Process B (Model-Centric)*: In this case, it is not assumed that the base distribution is available. Rather a single finite data set is available, and the random process of *BASE-F* is simply that of subsampling this finite data set. Note that the subsampling is now part of the detector itself, and therefore we have a randomized detector with a particular bias and variance on a finite data set. The expected values of the MSE, bias and variance are computed with respect to the random process implied by the stochastic behavior of this *detector*. This is the reason that this way of defining the bias-variance trade-off is referred to as model-centric. Note that although both *BASE-I* and *BASE-F* can be captured by random process A, only *BASE-F* can be captured by random process B. Therefore, the former approach is more general. However, the model-centric approach of bias-variance decomposition is more valuable in some ensemble settings where one cannot

relate the variance reduction directly with statistical variations in the training data, but with the randomized variations in the detector. In some ensembles (such as the example discussed in this sections), one can use both decomposition, but the data-centric decomposition provides better insights.

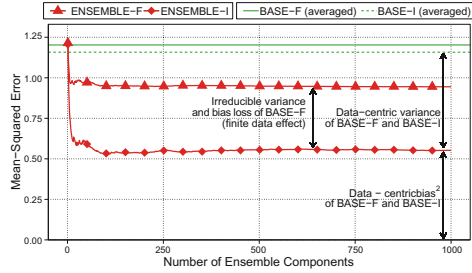
Although we do not distinguish between training and test points in the simulation of *BASE-F*, the predictions are still (roughly) similar to the case when training and test data were different, because distance-based detectors exclude the test point at hand while computing the nearest neighbors for the prediction. It does cause a small difference in training data size of a single point; this effect is negligible. Note that if we draw a different data set  $\mathcal{D}'$ , which is of the same size as the test data  $\mathcal{D}$  (for creating the subsampled training data sets), we will get roughly similar results. This fact is validated by the similar performance of *BASE-F* and *BASE-I* in Fig. 2.6.

In the following, we will run the process of 1000 trials; our basic assumption is that 1000 trials are sufficient to stabilize the ensemble performance from a practical point of view. Therefore, we can roughly estimate these results to be reflective of an infinite number of trials, which are required for accurately computing quantities like bias and variance.

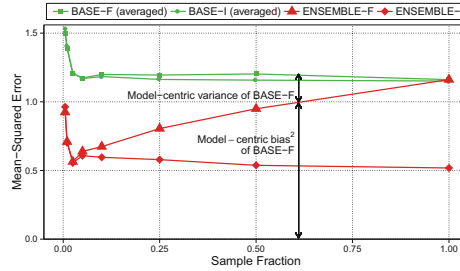
We show the performance of both the base and ensemble performance in Fig. 2.8. In this case, we present the results for the 2-dimensional distribution discussed earlier in this section. Note that Fig. 2.8a, b are respectively identical to Fig. 2.8c, d. However, they are annotated differently. In Fig. 2.8a, b, the annotation is designed to show the bias-variance decomposition according to the data-centric setting. The first observation is that the data-centric bias of *BASE-I* is simply the MSE of *ENSEMBLE-I* after a large number of trials. This is because *ENSEMBLE-I* has zero variance after a large number of subsamples, and all the *MSE* is simply the bias of its base detector. It is, however, less obvious why the bias of *BASE-F* and *BASE-I* should be the same. Note that the random process defining *BASE-F* includes the variance of the initial step of choosing the base data of size  $n_0$ , even though we see only one instance of this finite data set. After this variance is included, the variances of *BASE-F* and *BASE-I* are equivalent to sampling a training data of size  $f \cdot n_0$  directly from the base distribution. Furthermore, the expected *MSE* of *BASE-F* is also the same as that of *BASE-I*, even though there are minor differences in the *particular* case of Fig. 2.8 due to statistical fluctuations. In fact, the *MSE* of *BASE-I* in Fig. 2.8 reflects the *expected MSE* of *BASE-F* more closely than the specific instantiation of *BASE-F* in the figure. This is because *BASE-F* is constructed using a subset of the test points as the training data, whereas the data-centric random process assumes that the base detectors are constructed using a different sample from the same distribution (as in *BASE-I*). Therefore, the data-centric bias, variance, and *MSE* of both *BASE-F* and *BASE-I* are identical, and are defined completely by the *BASE-I* and *ENSEMBLE-I* simulations. Interestingly, both the *BASE-F* and *ENSEMBLE-F* simulations are completely irrelevant for defining the data-centric bias and variance of *BASE-F*. However, the *ENSEMBLE-F* plot is still interesting, in that it shows how much of the variance one can heuristically reduce, while working within the limitations of a finite resource (data set). Because of the finiteness of the data set, there are correlations among



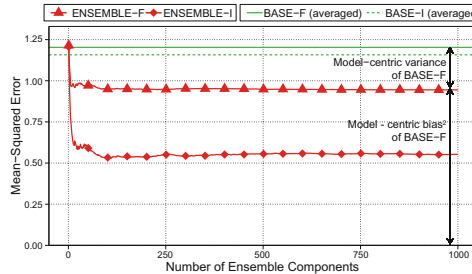
(a) Random Process A: Data-centric bias-variance decomposition (1000 components)



(b) Random Process A: Data-centric bias-variance decomposition (sampling rate 0:5)



(c) Random Process B: Model-centric bias-variance decomposition (1000 components)



(d) Random Process B: Model-centric bias-variance decomposition (sampling rate 0.5)

**Fig. 2.8** Two different ways of performing the bias-variance decomposition of the same MSE (data-centric view and model-centric view). One can define the bias-variance decomposition in any number of ways by choosing the appropriate random process over which the expectation is computed. An important assumption is that 1000 base detectors are sufficient for stabilization of ensemble performance, and therefore we are treating the (ensemble) variance to be negligible at that point

the base detectors of *ENSEMBLE-F*, and therefore only a portion of the variance can be reduced. The portion of the irreducible variance because of intra-detector correlations are annotated in Fig. 2.8a, b.

The bias-variance decomposition with random process B is shown in Fig. 2.8c, d. This model-centric approach can no longer decompose the error of *BASE-I* since it starts with a finite data set, and does not assume that the training data sets are drawn from a base distribution. Therefore, even though the plots of *BASE-I* and *ENSEMBLE-I* are shown in the figure, they are irrelevant from a bias-variance decomposition perspective. The draws of data from an infinite data set are relevant to the model-centric random process B. After a large number of trials *ENSEMBLE-F* is assumed to have zero variance, and therefore the variance of base *BASE-F* is simply the difference between *BASE-F* and *ENSEMBLE-F*. The remaining part of the *MSE* is the (model-centric) bias. It is interesting to note that the model-centric view does not provide a particularly satisfactory explanation of variance reduction in this particular setting of subsampling. However, as discussed in the next chapter, there are indeed several settings in which the model-centric view provides better insights. For example, methods like feature bagging can be better explained from a model-centric view of variance reduction. Therefore, an appropriate form of the bias-variance trade-off can be used to justify different types of ensemble methods.

## 2.5 Conclusions

This chapter introduces the theory of outlier ensembles. The ideas in this chapter show that the theoretical underpinnings of ensemble analysis for classification are not very different from those in outlier detection. As a result, many of the existing ensemble schemes for classification can be generalized directly to outlier detection. In this chapter, we provide both a data-centric and a model-centric view of outlier ensembles. These ideas can be used to explain both data-centric and model-centric outlier ensembles. By designing ensemble methods to reduce either the bias or the variance or both, one can design more accurate outlier detection methods. Bias-reduction methods are generally harder to adapt from classification to outlier detection because of the fact the accuracy needs to be computed in intermediate steps in most such methods. Accuracy computation requires the knowledge of labels that are not available in unsupervised settings. On the other hand, variance-reduction methods can be adapted more easily from classification to outlier detection. The theoretical ideas discussed in this chapter set the stage to view the rich literature in classification as a reservoir of ideas, which can be adapted in various ways to the outlier detection domain.

### Exercises

1. Consider a randomized outlier detection algorithm,  $g(\bar{X}, \mathcal{D})$ , which is almost ideal in the sense that it correctly learns the function  $f(\bar{X})$  most of the time. The value of  $f(\bar{X})$  is known to be finite. At the same time, because of a small

bug in the program, the randomized detector  $g(\bar{X}, \mathcal{D})$  outputs an  $\infty$  score about 0.00001% of the time. Furthermore, every test point is equally likely to receive such a score, although this situation occurs only 0.00001% of the time for any particular test instance. What is the model-centric bias of the bug-infested base detector  $g(\bar{X}, \mathcal{D})$ ?

2. Would you recommend running the randomized base detector of Exercise 1 multiple times, and averaging the predictions of the test instance? How about using the median?
3. Does the data-centric variance of an average  $k$ -nearest neighbor outlier detector increase or decrease with  $k$ ? What about the bias?

## References

1. C. C. Aggarwal. Outlier Ensembles: Position Paper, *ACM SIGKDD Explorations*, 14(2), pp. 49–58, December, 2012.
2. C. C. Aggarwal. Outlier Analysis, Second Edition, *Springer*, 2017.
3. C. C. Aggarwal and P. S. Yu. Outlier Detection in Graph Streams. *IEEE ICDE Conference*, 2011.
4. C. C. Aggarwal and S. Sathe. Theoretical Foundations and Algorithms for Outlier Ensembles, *ACM SIGKDD Explorations*, 17(1), June 2015.
5. L. Brieman. Bagging Predictors. *Machine Learning*, 24(2), pp. 123–140, 1996.
6. L. Brieman. Random Forests. *Journal Machine Learning archive*, 45(1), pp. 5–32, 2001.
7. G. Brown, J. Wyatt, R. Harris, and X. Yao. Diversity creation methods: a survey and categorisation. *Information Fusion*, 6:5(20), 2005.
8. R. Bryll, R. Gutierrez-Osuna, and F. Quek. Attribute Bagging: Improving Accuracy of Classifier Ensembles by using Random Feature Subsets. *Pattern Recognition*, 36(6), pp. 1291–1302, 2003.
9. P. Buhlmann, B. Yu. Analyzing bagging. *Annals of Statistics*, pp. 927–961, 2002.
10. P. Buhlmann. Bagging, Subbagging and Bragging for Improving Some Prediction Algorithms. *Recent advances and trends in nonparametric statistics*, Elsevier, 2003.
11. A. Buja, W. Stuetzle. Observations on bagging. *Statistica Sinica*, 16(2), 323, 2006.
12. M. Denil, D. Matheson, and N. De Freitas. Narrowing the Gap: Random Forests In Theory and in Practice. *ICML Conference*, pp. 665–673, 2014.
13. T. Dietterich. Ensemble Methods in Machine Learning. *First International Workshop on Multiple Classifier Systems*, 2000.
14. Y. Freund and R. Schapire. A Decision-theoretic Generalization of Online Learning and Application to Boosting. *Computational Learning Theory*, 1995.
15. Y. Freund and R. Schapire. Experiments with a New Boosting Algorithm. *ICML Conference*, pp. 148–156, 1996.
16. J. Friedman. On Bias, Variance, 0/1loss, and the Curse-of-Dimensionality. *Data Mining and Knowledge Discovery*, 1(1), pp. 55–77, 1997.
17. S. Geman, E. Bienenstock, and R. Doursat. Neural Networks and the Bias/Variance Dilemma. *Neural computation*, 4(1), pp. 1–58, 1992.
18. T. K. Ho. Random decision forests. *Third International Conference on Document Analysis and Recognition*, 1995. Extended version appears in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(8), pp. 832–844, 1998.
19. T. K. Ho. Nearest Neighbors in Random Subspaces. *Lecture Notes in Computer Science*, Vol. 1451, pp. 640–648, *Proceedings of the Joint IAPR Workshops SSPR'98 and SPR'98*, 1998. <http://link.springer.com/chapter/10.1007/BFb0033288>

20. R. Kohavi and D.H. Wolpert. Bias plus variance decomposition for zero-one loss functions, *ICML Conference*, 1996.
21. E. Kong and T. Dietterich. Error-Correcting Output Coding Corrects Bias and Variance. *Proceedings of the Twelfth International Conference on Machine Learning*, pp. 313–321, 1995.
22. A. Lazarevic, and V. Kumar. Feature Bagging for Outlier Detection, *ACM KDD Conference*, 2005.
23. F. T. Liu, K. M. Ting, and Z.-H. Zhou. Isolation Forest. *ICDM Conference*, 2008. Extended version appears in: *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 6(1), 3, 2012.
24. R. Michalski, I. Mozetic, J. Hong and N. Lavrac. The Multi-Purpose Incremental Learning System AQ15 and its Testing Applications to Three Medical Domains, *Proceedings of the Fifth National Conference on Artificial Intelligence*, pp. 1041–1045, 1986.
25. S. Rayana, L. Akoglu. Less is More: Building Selective Anomaly Ensembles with Application to Event Detection in Temporal Graphs. *SDM Conference*, 2015.
26. S. Rayana, L. Akoglu. Less is More: Building Selective Anomaly Ensembles. *ACM Transactions on Knowledge Discovery and Data Mining*, to appear, 2016.
27. L. Rokach. Pattern classification using ensemble methods, *World Scientific Publishing Company*, 2010.
28. M. Salehi, C. Leckie, M. Moshtaghi, and T. Vaithianathan. A Relevance Weighted Ensemble Model for Anomaly Detection in Switching Data Streams. *Advances in Knowledge Discovery and Data Mining*, pp. 461–473, 2014.
29. G. Seni and J. Elder. Ensemble Methods in Data Mining: Improving Accuracy through Combining Predictions. *Synthesis Lectures in Data Mining and Knowledge Discovery*, Morgan and Claypool, 2010.
30. R. Tibshirani. Bias, Variance, and Prediction Error for Classification Rules, *Technical Report, Statistics Department, University of Toronto*, 1996.
31. G. Valentini and T. Dietterich. Bias-variance Analysis of Support Vector Machines for the Development of SVM-based Ensemble Methods. *Journal of Machine Learning Research*, 5, pp. 725–774, 2004.
32. A. Zimek, M. Gaudet, R. Campello, J. Sander. Subsampling for efficient and effective unsupervised outlier detection ensembles, *KDD Conference*, 2013.
33. Z.-H. Zhou. Ensemble Methods: Foundations and Algorithms. *Chapman and Hall/CRC Press*, 2012.



Outlier Ensembles

An Introduction

Aggarwal, C.C.; Sathe, S.

2017, XVI, 276 p. 55 illus., 9 illus. in color., Hardcover

ISBN: 978-3-319-54764-0