CHAPTER 2

FOUNDATIONS OF IMBALANCED LEARNING

Gary M. Weiss

Fordham University

Abstract

Many important learning problems, from a wide variety of domains, involve learning from imbalanced data. Because this learning task is quite challenging, there has been a tremendous amount of research on this topic over the past fifteen years. However, much of this research has focused on methods for dealing with imbalanced data, without discussing exactly how or why such methods work—or what underlying issues they address. This is a significant oversight, which this chapter helps to address. This chapter begins by describing what is meant by imbalanced data, and by showing the effects of such data on learning. It then describes the fundamental learning issues that arise when learning from imbalanced data, and categorizes these issues as

either problem definition level issues, data level issues, or algorithm level issues. The chapter then describes the methods for addressing these issues and organizes these methods using the same three categories. As one example, the data-level issue of "absolute rarity" (i.e., not having sufficient numbers of minority-class examples to properly learn the decision boundaries for the minority class) can best be addressed using a data-level method that acquires additional minority-class training examples. But as we shall see in this chapter, sometimes such a direct solution is not available and less direct methods must be utlized. Common misconceptions are also discussed and explained. Overall, this chapter provides an understanding of the foundations of imbalanced learning by providing a clear description of the relevant issues, and a clear mapping from these *issues* to the *methods* that can be used to address them.

2.1 INTRODUCTION

Many of the machine learning and data mining problems that we study, whether they are in business, science, medicine, or engineering, involve some form of data imbalance. The imbalance is often an integral part of the problem and in virtually every case the less frequently occurring entity is the one that we are most interested in. For example, those working on fraud detection will focus on identifying the fraudulent transactions rather than the more common legitimate transactions [1], a telecommunications engineer will be far more interested in identifying equipment about to fail than equipment that will remain operational [2], and an industrial engineer will be more likely to focus on weld flaws than on welds that are completed satisfactorily [3].

In all of these situations it is far more important to accurately predict or identify the rarer case than the more common case, and this is reflected in the costs associated with errors in the predictions and classifications. For example, if we predict that telecommunication equipment is going to fail and it does not, we may incur some modest inconvenience and cost if the equipment is swapped out unnecessarily, but if we predict that equipment is not going to fail and it does, then we incur a much more significant cost when service is disrupted. In the case of medical diagnosis, the costs are even clearer: while a false-positive diagnosis may lead to a more expensive follow-up test and some patient anxiety, a false-negative diagnosis could result in death if a treatable condition is not identified.

This chapter covers the foundations of imbalanced learning. It begins by providing important background information and terminology and then describes the fundamental issues associated with learning from imbalanced data. This description provides the foundation for understanding the imbalanced learning problem. The chapter then categorizes the methods for handling class imbalance and maps each to the fundamental issue that each method addresses. This mapping is quite important since many research papers on imbalanced learning fail to provide a comprehensive description of how or why these methods work, and what underlying issue(s) they address. This chapter provides a good overview of the imbalanced learning problem and describes some of the key work in the area, but it is not intended to provide either a detailed description of the methods used for dealing with imbalanced data or a comprehensive literature survey. Details on many of the methods are provided in subsequent chapters in this book.

2.2 BACKGROUND

A full appreciation of the issues associated with imbalanced data requires some important background knowledge. In this section we look at what it means for a data set to be imbalanced, what impact class imbalance has on learning, the role of between-class imbalance and within-class imbalance, and how imbalance applies to unsupervised learning tasks.

2.2.1 What is an Imbalanced Data Set and what is its Impact on Learning?

We begin with a discussion of the most fundamental question: “What is meant by imbalanced data and imbalanced learning?” Initially we focus on classification problems and in this context learning from imbalanced data means learning from data in which the classes have unequal numbers of examples. But since virtually no datasets are perfectly balanced, this is not a very useful definition. There is no agreement, or standard, concerning the exact degree of class imbalance required for a data set to be considered truly “imbalanced.” But most practitioners would certainly agree that a data set where the most common class is less than twice as common as the rarest class would only be marginally unbalanced, that data sets with the imbalance ratio about 10:1 would be modestly imbalanced, and data sets with imbalance ratios above 1000:1 would be extremely unbalanced. But ultimately what we care about is how the imbalance impacts learning, and, in particular, the ability to learn the rare classes.

Learning performance provides us with an empirical—and objective—means for determining what should be considered an imbalanced data set. Figure 2.1, generated from data in an earlier study that analyzed twenty-six binary-class data sets [4], shows how class imbalance impacts minority-class classification performance. Specifically, it shows that the ratio between the minority class error rate and majority class error rate is greatest for the most highly imbalanced data sets and decreases as the amount of class imbalance decreases. Figure 2.1 clearly demonstrates that class imbalance leads to poorer performance when classifying minority-class examples, since the error rate ratios are above 1.0. This impact is actually quite severe, since data sets with class imbalance between 5:1 and 10:1 have a minority class error rate more than ten times that of the error rate on the majority class. The impact even appears quite significant for class imbalances between 1:1 and 3:1, which indicates that class imbalance is problematic in more situations than commonly acknowl-

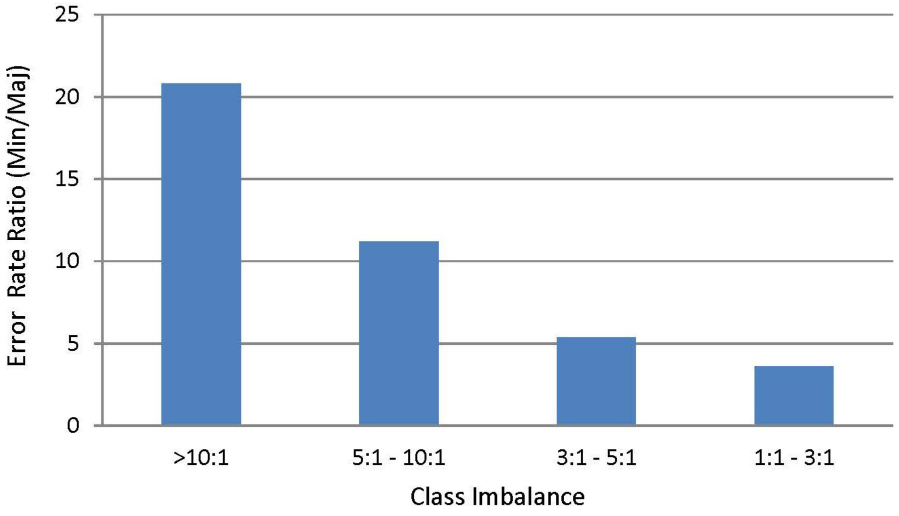

Figure 2.1 Impact of class imbalance on minority class performance

edged. This suggests that we should consider data sets with even moderate levels of class imbalance (e.g., 2:1) as "suffering" from class imbalance.

There are a few subtle points concerning class imbalance. First, class imbalance must be defined with respect to a particular data set or distribution. Since class labels are required in order to determine the degree of class imbalance, class imbalance is typically gauged with respect to the training distribution. If the training distribution is representative of the underlying distribution, as it is often assumed, then there is no problem; but if this is not the case, then we cannot conclude that the underlying distribution is imbalanced. But the situation can be complicated by the fact that when dealing with class imbalance, a common strategy is to artificially balance the training set. In this case, do we have class imbalance or not? The answer in this case is "yes"—we still do have class imbalance. That is, when discussing the problems associated with class imbalance we really care about the underlying distribution. Artificially balancing the training distribution may help with the effects of class imbalance, but does not remove the underlying problem.

A second point concerns the fact that while class imbalance literally refers to the relative proportions of examples belonging to each class, the absolute number of examples available for learning is clearly very important. Thus the class imbalance problem for a data set with 10,000 positive examples and 1,000,000 negative examples is clearly quite different from a data set with 10 positive examples and 1,000 negative examples—even though the class proportions are identical. These two problems can be referred to as problems with relative rarity and absolute rarity. A data set may suffer from neither of these problems, one of these problems, or both of these problems. We discuss the issue of absolute rarity in the context of class imbalance because highly imbalanced data sets very often have problems with absolute rarity.

2.2.2 Between-Class Imbalance, Rare Cases, and Small Disjuncts

Thus far we have been discussing class imbalance, or, as it has been termed, *between-class* imbalance. A second type of imbalance, which is not quite as well known or extensively studied, is *within-class* imbalance [5, 6]. Within-class imbalance is the result of rare cases [7] in the true, but generally unknown, classification concept to be learned. More specifically, rare cases correspond to sub-concepts in the induced classifier that cover relatively few cases. For example, in a medical dataset containing patient data where each patient is labeled as “sick” or “healthy”, a rare case might correspond to those sick patients suffering from botulism, a relatively rare illness. In this domain within-class imbalance occurs within the “sick” class because of the presence of much more general cases, such as those corresponding to the common cold. Just as the minority class in an imbalanced data set is very hard to learn well, the rare cases are also hard to learn—even if they are part of the majority class. This difficulty is much harder to measure than the difficulty with learning the rare class, since rare cases can only be defined with respect to the classification concept, which, for real-world problems, is unknown, and can only be approximated. However, the difficulty of learning rare cases can be measured using artificial datasets that are generated directly from a predefined concept. Figure 2.2 shows the results generated from the raw data from an early study on rare cases [7].

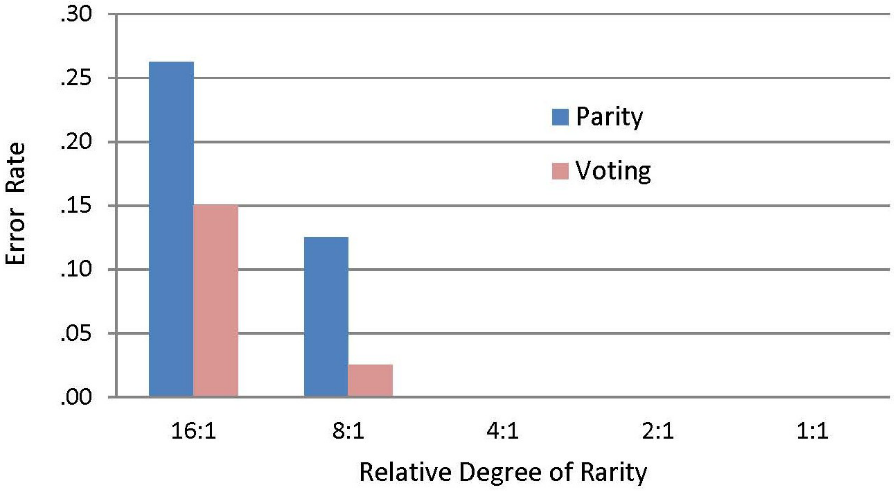

Figure 2.2 Impact of within-class imbalance on rare cases

Figure 2.2 shows the error rate for the cases, or subconcepts, within the parity and voting data sets, based on how rare the case is relative to the most general case in the classification concept associated with the data set. For example, a relative degree of rarity of 16:1 means that the rare case is 16 times as rare as the most common case, while a value of 1:1 corresponds to the most common case. For the two datasets shown in Figure 2.2 we clearly see that the rare cases (i.e., those with a higher relative degree of rarity) have a much higher error rate than the common cases, where, for this particular set of experiments, the more common cases are learned perfectly and have no errors. The concepts associated with the two data sets can be learned perfectly (i.e., there is no noise) and the errors were introduced by limiting the size of the training set.

Rare cases are difficult to analyze because one does not know the true concept and hence cannot identify the rare cases. This inability to identify these rare cases impacts the ability to develop strategies for dealing with them. But rare cases will manifest themselves in the learned concept, which is an approximation of the true concept. Many classifiers, such as decision tree and rule-based learners, form disjunctive concepts, and for these learners the rare cases will form small disjuncts—the disjuncts in the learned classifier that cover few training examples [8]. The relationship between the rare and common cases in the true (but generally unknown) concept, and the disjuncts in the induced classifier, is depicted in Figure 2.3.

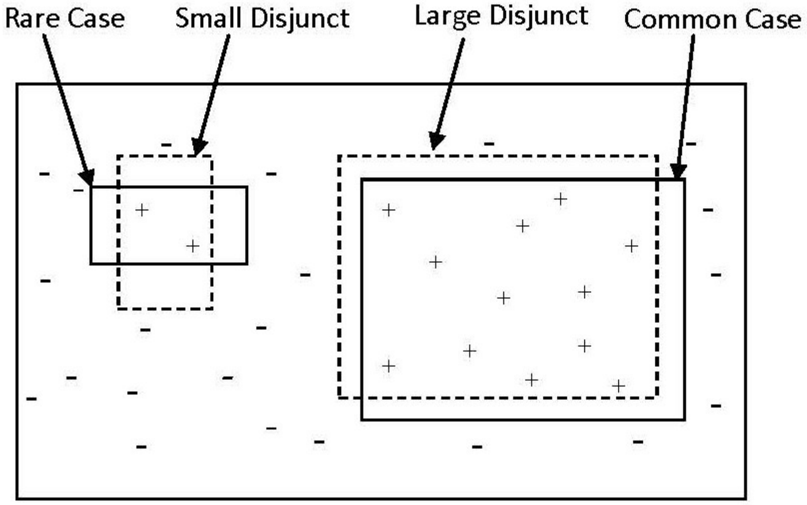

Figure 2.3 Relationship between rare/common cases and small/large disjuncts

Figure 2.3 shows a concept made up of two positively-labeled cases, one a rare case and one a common case, and the small disjunct and large disjunct that the classifier forms to cover them. Any examples located within the solid boundaries corresponding to these two cases should be labeled as positive and data points outside of these boundaries should be labeled as negative. The training examples are shown using the plus (“+”) and minus (“-”) symbols. Note that the classifier will have misclassification errors on future test examples, since the boundaries for the rare and common cases do not match the decision boundaries, represented by the dashed rectangles, which are formed by the classifier. Because approximately 50% of the decision boundary for the small disjunct falls outside of the rare case, we expect this small disjunct to have an error rate near 50%. Applying similar reasoning, the error rate of the large disjunct in this case will only be about 10%. Because the uncertainty in this noise-free case mainly manifests itself near the decision boundaries, in such cases we generally expect the small disjuncts to have a higher error rate, since a higher proportion of its “area” is near the decision boundary of the case to be learned. The difference between the induced decision boundaries and the actual decision boundaries in this case is mainly due to an insufficient number of training examples, although the bias of the learner also plays a role. In real-world situations, other factors, such as noise, will also have an effect.

The pattern of small disjuncts having a much higher error rates than large disjuncts, suggested by Figure 2.3, has been observed in practice in numerous studies [7, 8, 9, 10, 11, 12, 13]. This pattern is shown in Figure 2.4 for the classifier induced by C4.5 from the move data set [13]. Pruning was disabled in this case since pruning has been shown to obscure the effect of small disjuncts on learning [12]. The disjunct size, specified on the x-axis, is determined by the number of training examples correctly classified by the disjunct (i.e., leaf node). The impact of the error prone small disjuncts on learning is actually much greater than suggested by Figure 2.4, since the disjuncts of size 0-3, which correspond to the left-most bar in the figure, cover about 50% of the total examples and 70% of the errors.

In summary, we see that both rare classes and rare cases are difficult to learn and both lead to difficulties when learning from imbalanced data. When we discuss the foundational issues associated with learning from imbalanced data, we will see that these two difficulties are connected, in that rare classes are disproprotionately made up of rare cases.

2.2.3 Imbalanced Data for Unsupervised Learning Tasks

Virtually all work that focuses explicitly on imbalanced data focuses on imbalanced data for classification. While classification is a key supervised learning

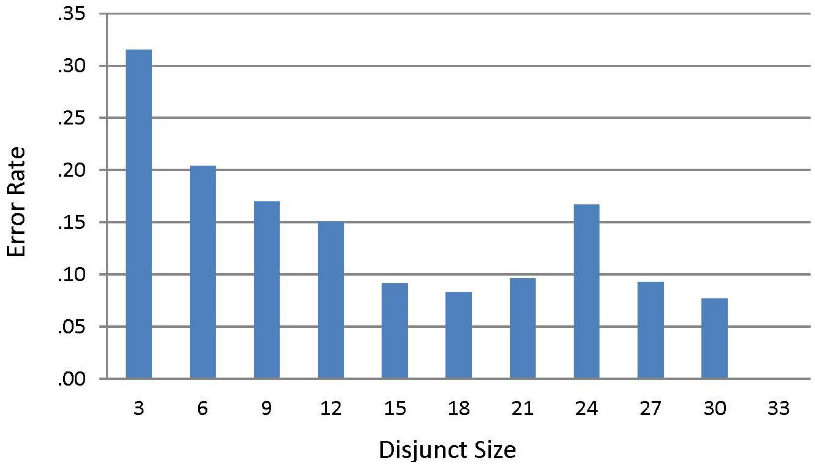

Figure 2.4 Impact of disjunct size on classifier performance (move data set)

task, imbalanced data can affect unsupervised learning tasks as well, such as clustering and association rule mining. There has been very little work on the effect of imbalanced data with respect to clustering, largely because it is difficult to quantify "imbalance" in such cases (in many ways this parallels the issues with identifying rare cases). But certainly if there are meaningful clusters containing relatively few examples, existing clustering methods will have trouble identifying them. There has been more work in the area of association rule mining, especially with regard to market basket analysis, which looks at how the items purchased by a customer are related. Some groupings of items, such as *peanut butter* and *jelly*, occur frequently and can be considered common cases. Other associations may be extremely rare, but represent highly profitable sales. For example *cooking pan* and *spatula* will be an extremely rare association in a supermarket, not because the items are unlikely to be purchased together, but because neither item is frequently purchased in a supermarket [14]. Association rule mining algorithms should ideally be able to identify such associations.

2.3 FOUNDATIONAL ISSUES

Now that we have established the necessary background and terminology, and demonstrated some of the problems associated with class imbalance, we are ready to identify and discuss the specific issues and problems associated with learning from imbalanced data. These issues can be divided into three major categories/levels: problem definition issues, data issues, and algorithm issues. Each of these categories is briefly introduced and then described in detail in subsequent subsections.

Problem definition issues occur when one has insufficient information to properly define the learning problem. This includes the situation when there is no objective way to evaluate the learned knowledge, in which case one cannot learn an optimal classifier. Unfortunately, issues at the problem definition level are commonplace. Data issues concern the actual data that is available for learning and includes the problem of absolute rarity, where there are insufficient examples associated with one or more classes to effectively learn the class. Finally, algorithm issues occur when there are inadequacies in a learning algorithm that make it perform poorly for imbalanced data. A simple example involves applying an algorithm designed to optimize accuracy to an imbalanced learning problem where it is more important to classify minority-class examples correctly than to classify majority-class examples correctly.

2.3.1 Problem Definition Level Issues

A key task in any problem solving activity is to *understand* the problem. As just one example, it is critically important for computer programmers to understand their customer's requirements before designing, and then implementing, a software solution. Similarly, in data mining it is critical for the data mining practitioner to understand the problem and the user requirements. For classification tasks, this includes understanding how the performance of the generated classifier will be judged. Without such an understanding it will be impossible to design an optimal or near-optimal classifier. While this need for evaluation information applies to all data mining problems, it is particularly important for problems with class imbalance. In these cases, as noted earlier, the costs of errors are often asymmetric and quite skewed, which violates the default assumption of most classifier induction algorithms, which is that errors have uniform cost and thus accuracy should be optimized. The impact of using accuracy as an evaluation metric in the presence of class imbalance is well known—in most cases poor minority class performance is traded off for improved majority class performance. This makes sense from an optimization standpoint, since overall accuracy is the weighted average of the accuracies associated with each class, where the weights are based on the proportion of training examples belonging to each class. This effect was clearly evident in Figure 2.1, which showed that the minority-class examples have a much lower accuracy than majority-class examples. What was not shown in Figure 2.1, but is shown by the underlying data [4], is that minority class predictions occur much less frequently than majority-class predictions, even after factoring in the degree of class imbalance.

Accurate classifier evaluation information, if it exists, should be passed to the classifier induction algorithm. This can be done in many forms, one of the simplest forms being a cost matrix. If this information is available, then it is the algorithm's responsibility to utilize this information appropriately; if the algorithm cannot do this, then there is an algorithm-level issue. Fortunately, over the past decade most classification algorithms have increased in sophistication so that they can handle evaluation criteria beyond accuracy, such as class-based misclassification costs and even costs that vary per example.

The problem definition issue also extends to unsupervised learning problems. Association rule mining systems do not have very good ways to evaluate the value of an association rule. But unlike the case of classification, since no single quantitative measure of quality is generated, this issue is probably better understood and acknowledged. Association rules are usually tagged with support and confidence values, but many rules with either high support or confidence values—or even both—will be uninteresting and potentially of little value. The lift of an association rule is a somewhat more useful measurement, but still does not consider the context in which the association will be used (lift measures how much more likely the antecedent and consequent of the rule are to occur together than if they where statistically independent). But as with classification tasks, imbalanced data causes further problems for the metrics most commonly used for association rule mining. As mentioned earlier, association rules that involve rare items are not likely to be generated, even if the rare items, when they do occur, often occur together (e.g., *cooking pan* and *spatula* in supermarket sales). This is a problem because such associations between rare items are more likely to be profitable because higher profit margins are generally associated with rare items.

2.3.2 Data Level Issues

The most fundamental data level issue is the lack of training data that often accompanies imbalanced data, which was previously referred to as an issue of *absolute rarity*. Absolute rarity does not only occur when there is imbalanced data, but is very often present when there are extreme degrees of imbalance—such as a class ratio of one to one million. In these cases the number of examples associated with the rare class, or rare case, is small in an absolute sense. There is no predetermined threshold for determining absolute rarity and any such threshold would have to be domain specific and would be determined based on factors such as the dimensionality of the instance space, the distribution of the feature values within this instance space, and, for classification tasks, the complexity of the concept to be learned.

Figure 2.5 visually demonstrates the problems that can result from an "absolute" lack of data. The figure shows a simple concept, identified by the solid rectangle; examples within this rectangle belong to the positive class and examples outside of this rectangle belong to the negative class. The decision boundary induced by the classifier from the labeled training data is indicated by the dashed rectangle. Figure 2.5a and 2.5b shows the same concept but with Figure 2.5b having approximately half as many training examples as in

Figure 2.5a. As one would expect, we see that the induced classifier more closely approximates the true decision boundary in Figure 2.5a, due to the availability of additional training data.

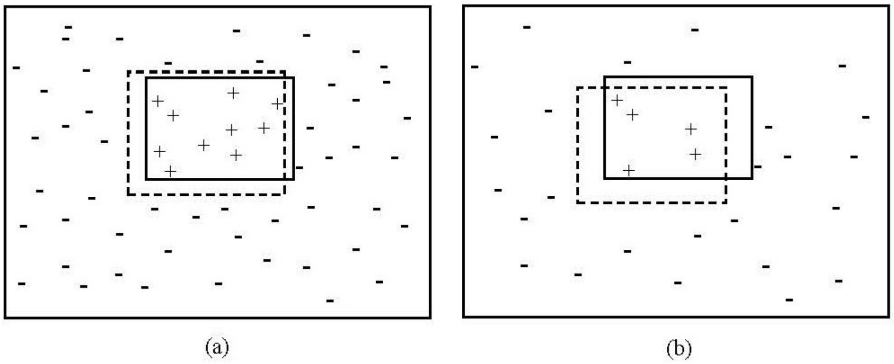

Figure 2.5 The impact of absolute rarity on classifier performance

Having a small amount of training data will generally have a much larger impact on the classification of the minority-class (i.e., positive) examples. In particular, it appears that about 90% of the space associated with the positive class (in the solid rectangle) is covered by the learned classifier in Figure 2.5a, while only about 70% of it is covered in Figure 2.5b. One paper summarized this effect as follows: "A second reason why minority-class examples are misclassified more often than majority-class examples is that fewer minority-class examples are likely to be sampled from the distribution *D*. Therefore, the training data are less likely to include (enough) instances of all of the minority-class subconcepts in the concept space, and the learner may not have the opportunity to represent all truly positive regions. Because of this, some minority-class test examples will be mistakenly classified as belonging to the majority class." [4, page 325].

Absolute rarity also applies to rare cases, which may not contain sufficiently many training example to be learned accurately. One study that used very simple artificially generated data sets found that once the training set dropped below a certain size, the error rate for the rare cases rose while the error rate for the general cases remained at zero. This occurred because with the reduced amount of training data, the common cases were still sampled sufficiently to be learned, but some of the rare cases were missed entirely [7]. The same study showed, more generally, that rare cases have a much higher misclassification rate than common cases. We refer to this as the *problem with rare cases*. This research also demonstrated something that had previously been assumed—that rare cases cause small disjuncts in the learned classifier. The *problem with small disjuncts*, observed in many empirical studies, is that they (i.e., small disjuncts) generally have a much higher error rate than large disjuncts [7, 8, 9, 10, 11, 12]. This phenomenon is again the result of a lack of data. The most thorough empirical study of small disjuncts analyzed thirty real-world data sets and showed that, for the classifiers induced from these data sets, the vast majority of errors are concentrated in the smaller disjuncts [12].

These results suggest that absolute rarity poses a very serious problem for learning. But the problem could also be that small disjuncts sometimes do not represent rare, or exceptional, cases, but instead represent noise. The underlying problem, then, is that there is no easy way to distinguish between those small disjuncts that represent rare/exceptional cases, which should be kept, and those that represent noise, which should be discarded (i.e., pruned).

We have seen that rare cases are difficult to learn due to a lack of training examples. It is generally assumed that rare classes are difficult to learn for similar reasons. But in theory it could be that rare classes are not disproportionately made up of rare cases, when compared to the makeup of common classes. But one study showed that this is most likely not the case since, across twenty-six data sets, the disjuncts labeled with the minority class were much smaller than the disjuncts with majority-class labels [4]. Thus, rare classes tend to be made up of more rare cases (on the assumption that rare cases form small disjuncts) and since these are harder to learn than common cases, the minority class will tend to be harder to learn than the majority class. This effect is therefore due to an absolute lack of training examples for the minority class.

Another factor that may exacerbate any issues that already exist with imbalanced data is *noise*. While noisy data is a general issue for learning, its impact is magnified when there is imbalanced data. In fact, we expect noise to have a greater impact on rare cases than on common cases. To see this, consider Figure 2.6. Figure 2.6a includes no noisy data while Figure 2.6b includes a few noisy examples. In this case a decision tree classifier is used which is configured to require at least two examples at the terminal nodes as a mean of overfitting avoidance. We see that in Figure 2.6b, when one of the two training examples in the rare positive case is erroneously labeled as belonging to the negative class, the classifier misses the rare case completely, since two positive training examples are required to generate a leaf node. The less rare positive case, however, is not significantly affected since most of the examples in the induced disjunct are still positive and the two erroneously labeled training examples are not sufficient to alter the decision boundaries. Thus, noise will have a more significant impact on the rare cases than on the common cases. Another way to look at things is that it will be hard to distinguish between rare cases and noisy data points. Pruning, which is often used to combat noise, will remove the rare cases and the noisy cases together.

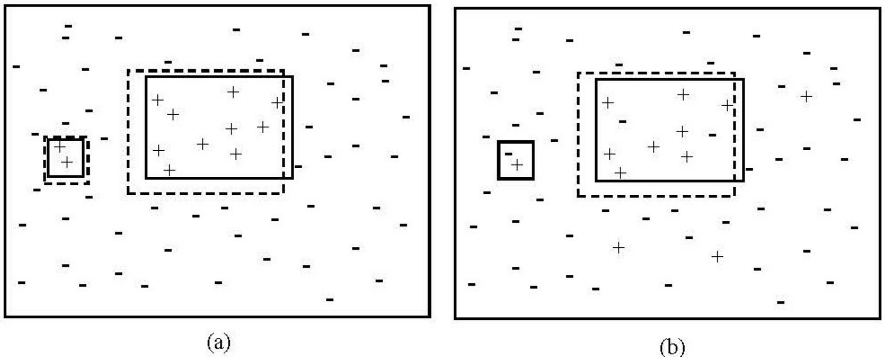

Figure 2.6 The effect of noise on rare cases

It is worth noting that while this section highlights the problem with absolute rarity, it does not highlight the problem with relative rarity. This is because we view relative rarity as an issue associated with the algorithm level. The reason is that class imbalance, which generally focuses on the relative differences in class proportions, is not fundamentally a problem at the data level—it is simply a property of the data distribution. We maintain that the problems associated with class imbalance and relative rarity are due to the lack of a proper problem formulation (with accurate evaluation criteria) or with algorithmic limitations with existing learning methods. The key point is that relative rarity/class imbalance is a problem only because learning algorithms cannot effectively handle such data. This is a very fundamental point, but one that is not often acknowledged.

2.3.3 Algorithm Level Issues

There are a variety of algorithm-level issues that impact the ability to learn from imbalanced data. One such issue is the inability of some algorithms to optimize learning for the target evaluation criteria. While this is a general issue with learning, it affects imbalanced data to a much greater extent than balanced data since in the imbalanced case the evaluation criteria typically diverge much further from the standard evaluation metric—accuracy. In fact, most algorithms are still designed and tested much more thoroughly for accuracy optimization than for the optimization of other evaluation metrics. This issue is impacted by the metrics used to guide the heuristic search process. For example, decision trees are generally formed in a top down manner and the tree construction process focuses on selecting the best test condition to expand the extremities of the tree. The quality of the test condition (i.e., the condition used to split the data at the node) is usually determined by the "purity" of a split, which is often computed as the weighted average of the purity values of each branch, where the weights are determined by the fraction of examples that follow that branch. These metrics, such as information gain, prefer test conditions that result in a balanced tree, where purity is increased for most of the examples, in contrast to test conditions that yield high purity for a relatively small subset of the data but low purity for the rest [15]. The problem with this is that a single high purity branch that covers only a few examples may identify a rare case. Thus, such search heuristics are biased against identifying highly accurate rare cases, which will also impact their performance on rare classes (which as discussed earlier are often comprised of rare cases).

The bias of a learning algorithm, which is required if the algorithm is to generalize from the data, can also cause problems when learning from imbalanced data. Most learners utilize a bias that encourages generalization and simple models to avoid the possibility of overfitting the data. But studies have shown that such biases work well for large disjuncts but not for small disjuncts [8], leading to the observed problem with small disjuncts (these biases tend to make the small disjuncts overly general). Inductive bias also plays a role with respect to rare classes. Many learners prefer the more common classes in the presence of uncertainty (i.e., they will be biased in favor of the class priors). As a simple example, imagine a decision tree learner that branches on all possible feature values when splitting a node in the tree. If one of the resulting branches covers no training examples, then there is no evidence on which to base a classification. Most decision-tree learners will predict the most frequently occurring class in this situation, biasing the results against rarer classes.

The algorithm-level issues discussed thus far concern the use of search heuristics and inductive biases that favor the common classes and cases over the rare classes and cases. But the algorithm-level issues do not just involve favoritism. It is fundamentally more difficult for an algorithm to identify rare patterns than to identify relatively common patterns. There may be quite a few instances of the rare pattern, but the sheer volume of examples belonging to the more common patterns will obscure the relatively rare patterns. This is perhaps best illustrated with a variation of a common idiom in English: finding relatively rare patterns is "like finding needles in a haystack." The problem in this case is not so much that there are few needles, but rather that there is so much more hay.

The problem with identifying relatively rare patterns is partly due to the fact that these patterns are not easily located using the greedy search heuristics that are in common use. Greedy search heuristics have a problem with relative rarity because the rare patterns may depend on the conjunction of many conditions, and therefore examining any single condition in isolation may not provide much information, or guidance. While this may also be true of common objects, with rare objects the impact is greater because the common objects may obscure the true signal. As a specific example of this general problem, consider the association rule mining problem described earlier, where we want to be able to detect the association between *cooking pan* and *spatula*. The problem is that both items are rarely purchased in a supermarket, so that even if the two are often purchased together when either one is purchased, this association may not be found. To find this association, the minimum support threshold for the algorithm would need to be set quite low. However, if this is done, there will be a combinatorial explosion because frequently occurring items will be associated with one another in an enormous number of ways. This association rule mining problem has been called the *rare item problem* [14] and it is an analog of the problem of identifying rare cases in classification problems. The fact that these random co-occurrences will swamp the meaningful associations between rare items is one example of the problem with relative rarity.

Another algorithm-level problem is associated with the divide-and-conquer approach that is used by many classification algorithms, including decision tree algorithms. Such algorithms repeatedly partition the instance space (and the examples that belong to these spaces) into smaller and smaller pieces. This process leads to data fragmentation [16], which is a significant problem when trying to identify rare patterns in the data, because there is less data in each partition from which to identify the rare patterns. Repeated partitioning can lead to the problem of absolute rarity within an individual partition, even if the original data set only exhibits the problem of relative rarity. Data mining algorithms that do not employ a divide-and-conquer approach therefore tend to be more appropriate when mining rare classes/cases.

2.4 METHODS FOR ADDRESSING IMBALANCED DATA

This section describes methods that address the issues with learning from imbalanced data that were identified in the previous section. These methods are organized based on whether they operate at the problem definition, data, or algorithm level. As methods are introduced the underlying issues that they address are highlighted. While this section covers most of the major methods that have been developed to handle imbalanced data, the list of methods is not exhaustive.

2.4.1 Problem Definition Level Methods

There are a number of methods for dealing with imbalanced data that operate at the problem definition level. Some of these methods are relatively straightforward in that they directly address foundational issues that operate at this same level. But due to the inherent difficulty of learning from imbalanced data, some methods have been introduced that simplify the problem in order to produce more reasonable results. Finally, it is important to note that in many cases there simply is insufficient information to properly define the problem and in these cases the best option is to utilize a method that moderates the impact of this lack of knowledge.

2.4.1.1 Use Appropriate Evaluation Metrics It is always preferable to use evaluation metrics that properly factor in how the mined knowledge will be used. Such metrics are essential when learning from imbalanced data since they will properly value the minority class. These metrics can be contrasted with accuracy, which places more weight on the common classes and assigns value to each class proportional to its frequency in the training set. The proper solution is to use meaningful and appropriate evaluation metrics and for imbalanced data this typically translates into providing accurate cost information to the learning algorithms (which should then utilize cost-sensitive learning to produce an appropriate classifier).

Unfortunately, it is not always possible to acquire the base information necessary to design good evaluation metrics that properly value the minority class. The next best solution is to provide evaluation metrics that are robust given this lack of knowledge, where "robust" means that the metrics yield good results over a wide variety of assumptions. If these metrics are to be useful for learning from imbalanced data sets, they will tend to value the minority class much more than accuracy, which is now widely recognized as a poor metric when learning from imbalanced data. This recognition has led to the ascension of new metrics to replace accuracy for learning from unbalanced data.

A variety of metrics are routinely used when learning from imbalanced data when accurate evaluation information is not available. The most common metric involves ROC analysis and AUC, the area under the ROC curve [17, 18]. ROC analysis can sometimes identify optimal models and discard suboptimal ones independent from the cost context or the class distribution (i.e., if one ROC curve dominates another), although in practice ROC curves tend to intersect so that there is no one dominant model. ROC analysis does not have any bias towards models that perform well on the majority class at the expense of the majority class—a property that is quite attractive when dealing with imbalanced data. AUC summarizes this information into a single number, which facilitates model comparison when there is not a dominating ROC curve. Recently there has been some criticism concerning the use of ROC analysis for model comparison [19], but nonetheless this measure is still the most common metric used for learning from imbalanced data.

Other common metrics used for imbalanced learning are based upon precision and recall. The precision of classification rules is essentially the accuracy associated with those rules, while the recall of a set of rules (or a classifier) is the percentage of examples of a designated class that are correctly predicted. For imbalanced learning, recall is typically used to measure the coverage of the minority class. Thus, precision and recall make it possible to assess the performance of a classifier on the minority class. Typically one generates precision and recall curves by considering alternative classifiers. Just like AUC is used for model comparison for ROC analysis, there are metrics that combine precision and recall into a single number to facilitate comparisons between models. These include the geometric mean (the square root of precision times recall) and the F-measure [20]. The F-measure is parameterized and can be adjusted to specify the relative importance of precision versus recall, but the F1-measure, which weights precision and recall equally, is the variant most often used when learning from imbalanced data.

It is also important to use appropriate evaluation metrics for unsupervised learning tasks that must handle imbalanced data. As described earlier, association rule mining treats all items equally even though rare items are often more important than common ones. Various evaluation metrics have been proposed to deal with this imbalance and algorithms have been developed to mine association rules that satisfy these metrics. One simple metric assigns uniform weights to each item to represent its importance, perhaps its per-unit profit [21]. A slightly more sophisticated metric allows this weight to vary based on the transaction it appears in, which can be used to reflect the quantity of the item [22, 23]. But such measures still cannot represent simple metrics like total profit. Utility mining [24, 25] provides this capability by allowing one to specify a uniform weight to represent per-item profit and a transaction weight to represent a quantity value. Objective oriented association rule mining [26] methods, which make it possible to measure how well an association rule meets a user's objective, can be used to find association rules in a medical dataset where only treatments that have minimal side effects and minimum levels of effectiveness are considered.

2.4.1.2 Redefine the Problem One way to deal with a difficult problem is to convert it into a simpler problem. The fact that the problem is not an equivalent problem may be outweighed by the improvement in results. This topic has received very little attention in the research community, most likely because it is not viewed as a research-oriented solution and is highly domain specific. Nonetheless, this is a valid approach that should be considered. One relatively general method for redefining a learning problem with imbalanced data is to focus on a subdomain, or partition of the data, where the degree of imbalance is lessened. As long as this subdomain or partition is easily identified, this is a viable strategy. It may also be a more reasonable strategy than removing the imbalance artificially via sampling. As a simple example, in medical diagnosis one could restrict the population to people over ninety years of age, especially if the targeted disease tends to be more common in the aged. Even if the disease occurs much more rarely in the young, using the entire population for the study could complicate matters if the people under ninety, due to their much larger numbers, collectively contribute more examples of the disease. Thus the strategy is to find a subdomain where the data is less imbalanced, but where the subdomain is still of sufficient interest. Other alternative strategies might be to group similar rare classes together and then simplify the problem by predicting only this "super-class."

2.4.2 Data Level Methods

The main data level issue identified earlier involves absolute rarity and a lack of sufficient examples belonging to rare classes and, in some cases, to the rare cases that may reside in either a rare or common class. This is a very difficult issue to address, but methods for doing this are described in this section. This section also describes methods for dealing with relative rarity (the standard class imbalance problem), even though, as we shall discuss, we believe that issues with relative rarity are best addressed at the algorithms level.

2.4.2.1 Active Learning and Other Information Acquisition Strategies The most direct way of addressing the issue of absolute rarity is to acquire additional labeled training data. Randomly acquiring additional labeled training data will be helpful and there are heuristic methods to determine if the projected improvement in classification performance warrants the cost of obtaining more training data—and how many additional training examples should be acquired

[27]. But a more efficient strategy is to preferentially acquire data from the rare classes or rare cases. Unfortunately, this cannot easily be done directly since one cannot identify examples belonging to rare classes and rare cases with certainty. But there is an expectation that active learning strategies will tend to preferentially sample such examples. For example, uncertainty sampling methods [28] are likely to focus more attention on rare cases, which will generally yield less certain predictions due to the smaller number of training examples to generalize from. Put another way, since small disjuncts have a much higher error rate than large disjuncts, it seems clear that active learning methods would focus on obtaining examples belonging to those disjuncts. Other work on active learning has further demonstrated that active learning methods are capable of preferentially sampling the rare classes by focusing the learning on the instances around the classification boundary [29]. This general information acquisition strategy is supported by the empirical evidence that shows that balanced class distributions generally yield better performance than unbalanced ones [4].

Active learning and other simpler information acquisition strategies can also assist with the relative rarity problem, since such strategies, which acquire examples belonging to the rarer classes and rarer cases, address the relative rarity problem while addressing the absolute rarity problem. Note that this is true even if uncertainty sampling methods tend to acquire examples belonging to rare cases, since prior work has shown that rare cases tend to be more associated with the rarer classes [4]. In fact, this method for dealing with relative rarity is to be preferred to the sampling methods addressed next, since those methods do not obtain new knowledge (i.e., valid new training examples).

2.4.2.2 Sampling Methods Sampling methods are a very popular method for dealing with imbalanced data. These methods are primarily employed to address the problem with relative rarity but do not address the issue of absolute rarity. This is because, with the exception of some methods that utilize some intelligence to generate new examples, these methods do not attack the underlying issue with absolute rarity—a lack of examples belonging to the rare classes and rare cases. But, as will be discussed in Section 2.4.3, our view is also that sampling methods do not address the underlying problem with relative rarity either. Rather, sampling masks the underlying problem by artificially balancing the data, without solving the basic underlying issue. The proper solution is at the algorithm level and requires algorithms that are designed to handle imbalanced data.

The most basic sampling methods are random undersampling and random oversampling. Random undersampling randomly eliminates majority-class examples from the training data while random oversampling randomly duplicates minority-class training examples. Both of these sampling techniques decrease the degree of class imbalance. But since no new information is introduced, any underlying issues with absolute rarity are not addressed. Some studies have shown random oversampling to be ineffective at improving recognition of the minority class [30, 31] while another study has shown that random undersampling is ineffective [32]. These two sampling methods also have significant drawbacks. Undersampling discards potentially useful majority-class examples, while oversampling increases the time required to train a classifier and also leads to overfitting that occurs to cover the duplicated training examples [31, 33].

More advanced sampling methods use some intelligence when removing or adding examples. This can minimize the drawbacks that were just described and, in the case of intelligently adding examples, has the potential to address the underlying issue of absolute rarity. One undersampling strategy only removes majority-class examples that are redundant with other examples or border regions with minority-class examples, figuring that they may be the result of noise [34]. SMOTE, on the other hand, oversamples the data by introducing new, non-replicated minority-class examples from the line segments that join the 5 minority-class nearest neighbors [33]. This tends to expand the decision boundaries associated with the small disjuncts/rare cases, as opposed to the overfitting associated with random oversampling. Another approach is to identify a good class distribution for learning and then generate samples with that distribution. Once this is done multiple training sets with the desired class distribution can be formed using all minority-class examples and a subset of the majority-class examples. This can be done so that each majority-class example is guaranteed to occur in at least one training set, so no data is wasted. The learning algorithm is then applied to each training set and meta-learning is used to form a composite learner from the resulting classifiers. This approach can be used with any learning method and it was applied to four different learning algorithms [1]. The same basic approach for partitioning the data and learning multiple classifiers has also been used with support vector machines and an SVM ensemble has outperformed both undersampling and oversampling [35].

All of these more sophisticated methods attempt to reduce some of the drawbacks associated with the simple random sampling methods. But for the most part it seems unlikely that they introduce any new knowledge and hence they do not appear to truly address any of the underlying issues previously identified. Rather, they at best compensate for learning algorithms that are not well suited to dealing with class imbalance. This point is made quite clearly in the description of the SMOTE method, when it is mentioned that the introduction of the new examples effectively serves to change the *bias* of the learner, forcing a more general bias, but only for the minority class. Theoretically such a modification to the bias could be implemented at the algorithm level. As discussed later, there has been research at the algorithm level in modifying the bias of a learner to better handle imbalanced data.

The sampling methods just described are designed to reduce between-class imbalance. Although research indicates that reducing between-class imbalance will tend to also reduce within-class imbalances [4], it is worth considering whether sampling methods can be used in a more direct manner to reduce within-class imbalances—and if this is beneficial. This question has been studied using artificial domains and the results indicate that it is not sufficient to eliminate between-class imbalances (i.e., rare classes) in order to learn complex concepts that contain within-class imbalances (i.e., rare cases) [5]. Only when the within-class imbalances are also eliminated can the concept be learned well. This suggests that sampling should be used to improve the performance associated with rare cases. Unfortunately, there are problems with implementing the strategy for real-world domains where one cannot identify the rare cases. The closest we can get to this approach is to assume that rare cases correspond to small disjuncts in the induced classifier and then sample based on disjunct size, with the goal of equalizing the sizes of the disjuncts in the induced classifier.

2.4.3 Algorithm Level Methods

A number of algorithm level methods have been developed to handle imbalanced data. The majority of these techniques involve using search methods that are well suited for identifying rare patterns in data when common patterns abound.

2.4.3.1 Search Methods that Avoid Greed and Recursive Partitioning Greedy search methods and search methods than use a divide and conquer approach to recursively partition the search space have difficulty finding rare patterns, for the reasons provided in Section 2.3.3. Thus learning methods that avoid, or minimize these two approaches, will tend to perform better when there is imbalanced data. The advances in computational power that have occurred since many of the basic learning methods were introduced make it more practical to utilize less greedy search heuristics. Perhaps the best example of this is genetic algorithms, which are global search techniques that work with populations of candidate solutions rather than a single solution and employ stochastic operators to guide the search process [36]. These methods tend to be far less greedy than many popular learning methods and these characteristics permit genetic algorithms to cope well with attribute interactions [36, 37] and avoid getting stuck in local maxima, which together make genetic algorithms very suitable for dealing with rarity. In addition, genetic algorithms also do not rely on a divide and conquer approach that leads to the data fragmentation problem. Several systems have relied on the power of genetic algorithms to handle rarity. Timeweaver [38] uses a genetic algorithm to predict very rare events while Carvalho and Freitas [39, 40] use a genetic algorithm to discover "small disjunct rules." Certainly other search methods are less greedy than decision trees and also do not suffer from the data fragmentation problems. However, no truly comprehensive study has examined a large variety of different search methods over a large variety of imbalanced data sets, so definitive conclusions cannot be drawn. Such studies would be useful and it is interesting to note that these types of large scale empirical studies have been conducted to compare the effectiveness of sampling methods—which have garnered much more focused attention from the imbalanced data community.

2.4.3.2 Search Methods that Use Metrics Designed to Handle Imbalanced Data

One problem level method for handling class imbalance involves using evaluation metrics that properly value the learned/mined knowledge. However, evaluation metrics also play a role at the algorithm level, to guide the heuristic search process. Some metrics have been developed to improve this search process when dealing with imbalanced data—most notably metrics based on precision and recall. Search methods that focus on simultaneously maximizing precision and recall may fail due to the difficulty of optimizing these competing values, so some systems adopt more sophisticated approaches. Timeweaver [38], a genetic algorithm-based classification system, periodically modifies the parameter to the F-measure that controls the relative importance of precision and recall in the fitness function, so that a diverse set of classification rules is evolved, with some rules having high precision and others high recall. The expectation is that this will eventually lead to rules with both high precision and recall. A second approach optimizes recall in the first phase of the search process and precision in the second phase, by eliminating false positives covered by the rules [41]. Returning to the needle and haystack analogy, this approach identifies regions likely to contain needles in the first phase and then discards strands of hay within these regions in the second phase.

2.4.3.3 Inductive Biases Better Suited for Imbalanced Data Most inductive learning systems heavily favor generality over specialization. While an inductive bias that favors generality is appropriate for learning common cases, it is not appropriate for rare cases and may even cause rare cases to be totally ignored. There have been several attempts to improve the performance of data mining systems with respect to rarity by choosing a more appropriate bias. The simplest approach involves modifying existing systems to eliminate some small disjuncts based on tests of statistical significance or using error estimation techniques—often as part of an overfitting avoidance strategy. The hope is that these will remove only improperly learned disjuncts but such methods will also remove those disjuncts formed to cover rare cases. The basic problem is that the significance of small disjuncts cannot be reliably estimated and consequently significant small disjuncts may be eliminated along with the insignificant ones. Error estimation techniques are also unreliable when there are only a few examples, and hence they suffer from the same basic problem. These approaches work well for large disjuncts because in these cases statistical significance and error rate estimation techniques yield relatively reliable estimates—something they do not do for small disjuncts.

More sophisticated approaches have been developed but the impact of these strategies on rare cases cannot be measured directly, since the rare cases in the true concept are generally not known. Furthermore, in early work on this topic the focus was on the performance of small disjuncts, so it is difficult to assess the impact of these strategies on class imbalance. In one study the learner's maximum generality bias was replaced with a maximum specificity bias for the small disjuncts, which improved the performance of the small disjuncts but degraded the performance of the larger disjuncts and the overall accuracy [8]. Another study also utilized a maximum specificity bias but took steps to ensure that this did not impact the performance of the large disjuncts, by using a different learning method classify them [11]. A similar hybrid approach was also used in one additional study [39, 40].

Others advocate the use of instance-based learning for domains with many rare cases/small disjuncts due to the highly specific bias associated with this learning method [10]. In such methods all training examples are generally stored in memory and utilized, as compared to other approaches where examples when they fall below some utility threshold are ignored (e.g., due to pruning). In summary, there have been several attempts to select an inductive bias that will perform better in the presence of small disjuncts, which are assumed to represent rare cases. But these methods have shown only mixed success and, most significantly, this work has not directly examined class imbalance; these methods may assist with class imbalance since rare classes are believed to be formed disproportionately from rare cases. Such approaches, which have not garnered much attention in the past decade, are quite relevant and should be reexamined in the more modern context of class imbalance.

2.4.3.4 Algorithms that Implicitly or Explicitly Favor Rare Classes and Cases

Some algorithms preferentially favor the rare classes or cases and hence tend to perform well on classifying rare classes and cases. Cost-sensitive learning algorithms are one of the most popular such algorithms for handling imbalanced data. While the assignment of costs in response to the problem characteristics is done at the problem level, cost-sensitive learning must ultimately be implemented at the algorithm level. There are several algorithmic methods for implementing cost sensitive learning, including weighting the training examples in a cost proportionate manner [42] and building the cost-sensitivity directly into the learning algorithm [43]. These iterative algorithms place different weights on the training distribution after each iteration and increase (decrease) the weights associated with the incorrectly (correctly) classified examples. Because rare classes/cases are more error-prone than common classes/cases [4, 38] it is reasonable to believe that boosting will improve their classification performance. Note that because boosting effectively alters the distribution of the training data, one could consider it a type of advanced adaptive sampling technique. AdaBoost's weight-update rule has also been made cost-sensitive, so that misclassified examples belonging to rare classes are assigned higher weights than those belonging to common classes. The resulting system, Adacost [44], has been empirically shown to produce lower cumulative misclassification costs than AdaBoost and thus, like other cost-sensitive learning methods, can be used to address the problem with rare classes.

Boosting algorithms have also been developed to directly address the problem with rare classes. RareBoost [45] scales false-positive examples in proportion to how well they are distinguished from true-positive examples and scales false-positive examples in proportion to how well they are distinguished from true-negative examples. A second algorithm that uses boosting to address the problems with rare classes is SMOTEBoost [46]. This algorithm recognizes that boosting may suffer from the same problems as oversampling (e.g., overfitting), since boosting will tend to weight examples belonging to the rare classes more than those belonging to the common classes—effectively duplicating some of the examples belonging to the rare classes. Instead of changing the distribution of training data by updating the weights associated with each example, SMOTEBoost alters the distribution by adding new minority-class examples using the SMOTE algorithm [33].

2.4.3.5 Learn only the Rare Class The problem of relative rarity often causes the rare classes to be ignored by classifiers. One method for addressing this data level problem is to employ an algorithm that only learns classification rules for the rare class, since this will prevent the more common classes from overwhelming the rarer classes. There are two main variations to this approach. The recognition-based approach learns only from examples associated with the rare class, thus recognizing patterns shared by the training examples, rather than discriminating between examples belonging to different class. Several systems have used such recognition-based methods to learn rare classes [47, 48].

The other approach, which is more common and supported by several learning algorithms, learns from examples belonging to all classes but first learns rules to cover the rare classes [15, 49, 50]. Note that this approach avoids most of the problems with data fragmentation, since examples belonging to the rare classes will not be allocated to the rules associated with the common classes, before any rules are formed that cover the rare classes. Such methods are also free to focus only on the performance of the rules associated with the rare class and not worry about how this affects the overall performance of the classifier [15, 50]. Probably the most popular such algorithm is the Ripper algorithm [49], which builds rules using a separate-and-conquer approach. Ripper normally generates rules for each class from the rarest class to the most common class. At each stage it grows rules for the one targeted class by adding conditions until no examples are covered that belong to the other classes. This leads to highly specialized rules, which are good for covering rare cases. Ripper then covers the most common class using a default rule that is used when no other rule is applicable.

2.4.3.6 Algorithms for Mining Rare Items Association rule mining is a well understood area. However, when metrics other than support and confidence are used to identify itemsets or their association rules, algorithmic changes are required. In Section 2.4.1 we briefly discussed a variety of metrics for finding association rules when additional metrics are added to support and confidence. We did not describe the corresponding changes to the association rule mining algorithms, but they are described in detail in the relevant papers [21, 22, 23, 24, 25, 26].

There is also an algorithmic solution to the rare item problem, in which significant associations between rarely occurring items may be missed, because the minimum support value, *minsup*, cannot be set too low, because a very low value would cause a combinatorial explosion of associations. This problem can be solved by specifying multiple minimum levels of support to reflect the frequencies of the associated items in the distribution [14]. Specifically, the user can specify a different *minsup* value for each item. The minimum support for an association rule is then the lowest *minsup* value amongst the items in the rule. Association rule mining systems are tractable mainly because of the downward closure property of support: if a set of items satisfies *minsup* then so do all of its subsets. While this downward closure property does not hold with multiple minimum levels of support, the standard Apriori algorithm for association rule mining can be modified to satisfy the sorted closure property for multiple minimum levels of support [14]. The use of multiple minimum levels of support then becomes tractable. Empirical results indicate that the new algorithm is able to find meaningful associations involving rare items without producing a huge number of meaningless rules involving common items.

2.5 MAPPING FOUNDATIONAL ISSUES TO SOLUTIONS

This section briefly summarizes the foundational problems with imbalanced data described in Section 2.3 and how they can be addressed by the various methods described in Section 2.4. This section is organized using the three basic categories identified earlier in this chapter: problem definition level, data level, and algorithm level.

The problem definition level issues arise because researchers and practitioners often do not have all of the necessary information about a problem to solve it optimally. Most frequently this involves not possessing the necessary metrics to accurately assess the utility of the mined knowledge. The solution to this problem is simple, although often not achievable: obtain the requisite knowledge and from this generate the metrics necessary to properly evaluate the mined knowledge. Because this is not often possible, one must take the next best course of action—use the best available metric or one that is at least "robust" such that it will lead to good, albeit suboptimal solutions, given incomplete knowledge and hence inexact assumptions. In dealing with imbalanced data this often means using ROC analysis when the necessary evaluation information is missing. One alternate solution that was briefly discussed involves redefining the problem to a simpler problem for which more exact evaluation information is available. Fortunately the state of the art in data mining technology has advanced to the point where in most cases if we do have the precise evaluation information, we can utilize it; in the past data mining algorithms were often not sufficiently sophisticated to incorporate such knowledge.

Data level issues also arise when learning from imbalanced data. These issues mainly relate to absolute rarity. Absolute rarity occurs when one or more classes do not have sufficient numbers of examples to adequately learn the decision boundaries associated with that class. Absolute rarity has a much bigger impact on the rare classes than on common classes. Absolute rarity also applies to rare cases, which may occur for either rare classes or common classes, but are disproportionately associated with rare classes. The ideal and most straightforward approach to handling absolute rarity, in either of its two main forms, is to acquire additional training examples. This can often be done most efficiently via active learning and other information acquisition strategies.

It is important to understand that we do not view class imbalance, which results from a relative difference in frequency between the classes, as a problem at the data level—the problem only exists because most algorithms do not respond well to such imbalances. The straightforward method for dealing with class imbalance is via sampling, a method that operates at the data level. But this method for dealing with class imbalance has many problems, as we discussed previously (e.g., undersampling involves discarding potentially useful data) and is far from ideal. A much better solution would be to develop algorithms that can handle the class imbalance. At the current moment sampling methods do perform competitively and therefore cannot be ignored, but it is important to recognize that such methods will always have limited value and that algorithmic solutions can potentially be more effective. We discuss these methods next (e.g., one-class learning) because we view them as addressing foundational algorithmic issues.

Algorithm level issues mainly involve the ability to find subtle patterns in data that may be obscured due to imbalanced data and class imbalance in particular (i.e., relative rarity). Finding patterns, such as those that identify examples belonging to a very rare class, is a very difficult task. To accomplish this task it is important to have an appropriate search algorithm, a good evaluation metric to guide the heuristic search process, and an appropriate inductive bias. It is also important to deal with issues such as data fragmentation, which can be especially problematic for imbalanced data. The most common mechanism for dealing with this algorithm level problem is to use sampling, a data level method, to reduce the degree of class imbalance. But for reasons outlined earlier, this strategy does not address the foundational underlying issue—although it does provide some benefit. The strategies that function at the algorithm level include: using a non-greedy search algorithm and one that does not repeatedly partition the search space, using search heuristics that are guided by metrics that are appropriate for imbalanced data, using inductive biases that are appropriate for imbalanced data, and using algorithms that explicitly or implicitly focus on the rare classes or rare cases, or only learn the rare class.

2.6 MISCONCEPTIONS ABOUT SAMPLING METHODS

Sampling methods are the most common methods for dealing with imbalanced data, but yet there are widespread misconceptions related to these methods. The most basic misconception concerns the notion that sampling methods are *equivalent* to certain other methods for dealing with class imbalance. In particular, Breiman [51] establishes the connection between the distribution of training-set examples, the costs of mistakes on each class, and the placement of the decision threshold. Thus, for example, one can make false negatives twice as costly as false positives by assigning appropriate costs or by increasing the ratio of positive to negative examples in the training set by a factor of two or by setting the probability threshold for determining the class label to two-thirds rather than one-half. Unfortunately, as implemented in real-world situations, these equivalences do *not* hold.

As a concrete example, suppose a training set has 10,000 examples and a class distribution of 100:1, so that there are only 100 positive examples. One way to improve the identification of the rare class is to impose a greater cost for false negatives than for false positives. A cost ratio of 100:1 is theoretically equivalent to modifying the training distribution so that it is balanced, with a 1:1 class ratio. To generate such a balanced distribution in practice, one would typically oversample the minority class, undersample the majority class, or do both. But if one undersamples the majority class, then potentially valuable data is thrown away and if one oversamples the minority class, then one is making exact copies of examples, which can lead to overfitting. For the equivalence to hold, one should randomly select *new* minority class examples from the original distribution, which would include examples that are not already available for training. But this is almost never feasible. Even generating new, synthetic, minority-class examples violates the equivalence, since these examples will, at best, only be a better *approximation* of the true distribution. Thus sampling methods are not equivalent in practice to other methods for dealing with imbalanced data and they have drawbacks that other methods, such as cost-sensitive learning, do not have, if implemented properly.

Another significant concern with sampling is that its impact is often not fully understood—or even considered. Increasing the proportion of examples belonging to the rare class has two distinct effects. First, it will help address the problems with relative rarity, and, if the examples are new examples, will also address the problem with absolute rarity by injecting new knowledge. However, if no corrective action is taken, it will also have a second effect—it will impose non-uniform error costs, causing the learner to be biased in favor of predicting the rare class. In many situations this second effect is desired and is the actually the main reason for altering the class distribution of the training data. But in other cases, namely when new examples are added (e.g., via active learning), this effect is not desirable. That is, in these other cases the intent is to improve performance with respect to the rare class by having *more data* available for that class, not by biasing the data mining algorithm toward that class. In these cases this bias should be removed.

The bias introduced toward predicting the oversampled class can be removed using the equivalences noted earlier to account for the differences between the training distribution and the underlying distribution [4, 43]. For example, the bias can be removed by adjusting the decision thresholds, as was done in one study which demonstrated the positive impact of removing this unintended bias [4]. That study showed that adding new examples to alter the class distribution of the training data, so that it deviates from the natural, underlying, distribution, improved classifier performance. However, classifier performance was improved even more when the bias just described was removed by adjusting the decision thresholds within the classifier. Other research studies that investigate the use of sampling to handle rare cases and class imbalance almost never remove this bias—and worse yet, do not even discuss the implications of this decision. This issue must be considered much more carefully in future studies.

2.7 RECOMMEDATIONS AND GUIDELINES

This chapter categorized some of the major issues with imbalanced data and then described the methods most appropriate for handling each type of issue. Thus one recommendation is to try to use those methods for handling imbalanced data that are most appropriate for dealing with the underlying issue. This usually means utilizing methods at the same level as the issue, when possible. But often the ideal method is not feasible—like using active learning to obtain more training data when there is an issue of absolute rarity. Thus, one must often resort to sampling, but in such cases one should be aware of the drawbacks associated with these methods and avoid the common misconceptions associated with these methods. Unfortunately, it is not easy to effectively deal with imbalanced data because of the fundamental issues that are involved—which is probably why even after more than a decade of intense scrutiny, the research community still has much work remaining to come up with effective methods for dealing with these problems. Even methods that had become accepted, such as the use of AUC to generate robust classifiers when good evaluation metrics are not available, are now coming into question [19]. Nonetheless, there has been progress, and certainly there is a much better appreciation of the problem than in the past.

REFERENCES

1. P. Chan and S. Stolfo, "Toward scalable learning with non-uniform class and cost distributions: a case study in credit card fraud detection," in *Proc. of the Fourth International Conference on Knowledge Discovery and Data Mining*, pp. 164–168, 2001.

2. G. Weiss and H. Hirsh, "Learning to predict rare events in event sequences," in *Proc. of the Fourth International Conference on Knowledge Discovery and Data Mining*, pp. 359–363, 1998.

3. T. Liao, "Classification of weld flaws with imbalanced data," *Expert Systems with Applications: An International Journal*, vol. 35, no. 3, pp. 1041–1052, 2008.

4. G. Weiss and F. Provost, "Learning when training data are costly: the effect of class distribution on tree induction," *Journal of Artificial Intelligence Research*, vol. 19, pp. 315–354, 2003.

5. N. Japkowicz, “Concept learning in the presence of between-class and within-class imbalances,” in *Proc. of the Fourteenth Conference of the Canadian Society for Computational Studies of Intelligence*, pp. 67–77, 2001.

6. N. Japkowicz, “Class imbalances versus small dijsuncts,” *ACM SIGKDD Explorations*, vol. 6, no. 1, pp. 40–49, 2004.

7. G. Weiss, “Learning with rare cases and small disjuncts,” in *Proc. of the Twelfth International Conference on Machine Learning*, pp. 558–565, 1995.

8. R. Holte, L. Acker, and B. Porter, “Concept learning and the problem of small disjuncts,” in *Proc. of the Eleventh International Joint Conference on Artificial Intelligence*, pp. 813–818, 1989.

9. K. Ali and M. Pazzani, “HYDRA-MM: learning multiple descriptions to improve classification accuracy,” *International Journal of Artificial Intelligence Tools*, vol. 4, pp. 97–122, 1995.

10. A. van den Bosch, T. Weijters, H. J. van den Herik, and W. Daelemans, “When small disjuncts abound, try lazy learning: A case study,” in *Proc. of the Seventh Belgian-Dutch Conference on Machine Learning*, pp. 109–118, 1997.

11. K. Ting, “The problem of small disjuncts: its remedy in decision trees,” in *Proc. of the Tenth Canadian Conference on Artificial Intelligence*, pp. 91–97, 1994.

12. G. Weiss and H. Hirsh, “A quantitative study of small disjuncts,” in *Proc. of the Seventeenth National Conference on Artificial Intelligence*, pp. 665–670, AAAI Press, 2000.

13. G. Weiss and H. Hirsh, “A quantitative study of small disjuncts: experiments and results,” Tech. Rep. ML-TR-42, Rutgers University, 2000.

14. B. Liu, W. Hsu, and Y. Ma, “Mining association rules with multiple minimum supports,” in *Proc. of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 337–341, 1999.

15. P. Riddle, R. Segal, and O. Etzioni, “Representation design and brute-force induction in a boeing manufacturing design,” *Applied Artificial Intelligence*, vol. 8, pp. 125–147, 1994.

16. J. Friedman, R. Kohavi, and Y. Yun, “Lazy decision trees,” in *Proc. of the Thirteenth National Conference on Artificial Intelligence*, pp. 717–724, 1996.

17. A. Bradley, “The use of the area under the ROC curve in the evaluation of machine learning algorithms,” *Pattern Recognition*, vol. 30, no. 7, pp. 1145–1159, 1997.

18. F. Provost and T. Fawcett, “Robust classification for imprecise environments,” *Machine Learning*, vol. 42, pp. 203–231, 2001.

19. D. Hand, “Measuring classifier performance: A coherent alternative to the area under the ROC curve,” *Machine Learning*, vol. 77, pp. 103–123, 2009.

20. C. van Rijsbergen, *Information Retrieval.* London: Butterworths, 1979.

21. C. Cai, A. Fu, C. Cheng, and W. Kwong, “Mining association rules with weighted items,” in *Proc. of Database Engineering and Applications Symposium*, pp. 68–77, 1998.

22. C. Carter, H. Hamilton, and J. Cercone, “Share based measures for itemsets,” in *Principles of Data Mining and Knowledge Discovery, Lecture Notes in Computer Science, 1263*, pp. 14–24, 1997.

23. W. Wang, J. Yang, and P. Yu, “Efficient mining of weighted association rules (WAR),” in *Proc. of the 6th ACM International Conference on Knowledge Discovery and Data Mining*, pp. 270–274, 2000.

24. H. Yao, H. Hamilton, and C. Butz, “A foundational approach to mining itemset utilities from databases,” in *Proc. of the 4th SIAM International Conference on Data Mining*, pp. 482–496, 2004.

25. J. Yao and J. Hamilton, “Mining itemset utilities from transaction databases,” *Data and Knowledge Engineering*, vol. 59, no. 3, pp. 603–626, 2006.

26. Y. Shen, Q. Yang, and Z. Zhang, “Objective-oriented utility-based association mining,” in *Proc. of the 2002 IEEE International Conference on Data Mining*, pp. 426–433, 2002.

27. G. Weiss and Y. Tian, “Maximizing classifier utility when there are data acquisition and modeling costs,” *Data Mining and Knowledge Discovery*, vol. 17, no. 2, pp. 253–282, 2008.

28. D. Lewis and J. Catlett, “Heterogeneous uncertainty sampling for supervised learning,” in *Proc. of the Eleventh International Conference on Machine Learning*, pp. 148–156, 1994.

29. S. Ertekin, J. Huang, and C. Giles, “Active learning for class imbalance problem,” in *Proc. of the 30th International Conference on Research and Development in Information Retrieval*, 2007.

30. C. Ling and C. Li, “Data mining for direct marketing problems and solutions,” in *Proc. of the Fourth International Conference on Knowledge Discovery and Data Mining*, pp. 73–79, 1998.

31. C. Drummond and R. Holte, “C4.5, class imbalance, and cost sensitivity: why under-sampling beats over-sampling,” in *ICML Workshop on Learning from Imbalanced Data Sets II*, 2003.

32. N. Japkowicz and S. Stephen, “The class imbalance problem: a systematic study,” *Intelligent Data Analysis*, vol. 6, no. 5, pp. 429–450, 2002.

33. N. Chawla, K. Bowyer, L. Hall, and W. Kegelmeyer, “SMOTE: Synthetic minority over-sampling technique,” *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.

34. M. Kubat and S. Matwin, “Addressing the curse of imbalanced training sets: one-sided selection,” in *Proc. of the Fourteenth International Conference on Machine Learning*, pp. 179–186, Morgan Kaufmann, 1997.

35. R. Yan, Y. Liu, R. Jin, and A. Hauptmann, “On predicting rare classes with SVM ensembles in scene classification,” in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing*, 2003.

36. D. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning.* Addison-Wesley, 1989.

37. A. Freitas, “Evolutionary computation,” in *Handbook of Data Mining and Knowledge Discovery*, pp. 698–706, Oxford University Press, 2002.

38. G. Weiss, “Timeweaver: a genetic algorithm for identifying predictive patterns in sequences of events,” in *Proc. of the Genetic and Evolutionary Computation Conference*, pp. 718–725, 1999.

39. D. Carvalho and A. A. Freitas, “A genetic algorithm for discovering small-disjunct rules in data mining,” *Applied Soft Computing*, vol. 2, no. 2, pp. 75–88, 2002.

40. D. Carvalho and A. A. Freitas, “New results for a hybrid decision tree/genetic algorithm for data mining,” in *Proc. of the Fourth International Conference on Recent Advances in Soft Computing*, pp. 260–265, 2002.

41. M. Joshi, R. Agarwal, and V. Kumar, “Mining needles in a haystack: classifying rare classes via two-phase rule induction,” in *SIGMOD ’01 Conference on Management of Data*, pp. 91–102, 2001.

42. B. Zadrozny, J. Langford, and N. Abe, “Cost-sensitive learning by cost-proportionate example weighting,” in *Proc. of the Third IEEE International Conference on Data Mining*, pp. 435–442, 2003.

43. C. Elkan, “The foundations of cost-sensitive learning,” in *Proc. of the Seventeenth International Conference on Machine Learning*, pp. 239–246, 2001.

44. W. Fan, S. Stolfo, J. Zhang, and P. Chan, “Adacost: misclassification cost-sensitive boosting,” in *Proc. of the Sixteenth International Conference on Machine Learning*, pp. 99–105, 1999.

45. M. Joshi, V. Kumar, and R. C. Agarwal, “Evaluating boosting algorithms to classify rare cases: comparison and improvements,” in *First IEEE International Conference on Data Mining*, pp. 257–264, 2001.

46. N. Chawla, A. Lazarevic, L. Hall, and K. Bowyer, “SMOTEBoost: Improving prediction of the minority class in boosting,” in *Proc. of Principles of Knowledge Discovery in Databases*, pp. 107–119, 2003.

47. N. Japkowicz, C. Myers, and M. Gluck, “A novelty detection approach to classification,” in *Proc. of the Fourteenth Joint Conference on Artificial Intelligence*, pp. 518–523, 1995.

48. B. Raskutti and A. Kowalczyk, “Extreme re-balancing for SVMs: a case study,” in *ICML Workshop on Learning from Imbalanced Data Sets II*, 2003.

49. W. Cohen, “Fast effective rule induction,” in *Proc. of the Twelfth International Conference on Machine Learning*, pp. 115–123, 1995.

50. M. Kubat, R. Holte, and S. Matwin, “Learning when negative examples abound,” in *Machine Learning: ECML-97, Lecture Notes in Artificial Intelligence 1224*, pp. 146–153, Springer, 1997.

51. L. Breiman, J. Friedman, R. Olshen, and C. Stone, *Classification and Regressions Trees.* Boca Raton, Fl: Chapman and Hall/CRC Press, 1984.