Chapter 9
The Impact of Small Disjuncts on Classifier Learning

Gary M. Weiss

Abstract Many classifier induction systems express the induced classifier in terms of a disjunctive description. Small disjuncts are those that classify few training examples. These disjuncts are interesting because they are known to have a much higher error rate than large disjuncts and are responsible for many, if not most, of all classification errors. Previous research has investigated this phenomenon by performing ad hoc analyses of a small number of data sets. In this chapter we provide a much more systematic study of small disjuncts and analyze how they affect classifiers induced from 30 real-world data sets. A new metric, error concentration, is used to show that for these 30 data sets classification errors are often heavily concentrated toward the smaller disjuncts. Various factors, including pruning, training set size, noise, and class imbalance are then analyzed to determine how they affect small disjuncts and the distribution of errors across disjuncts. This analysis provides many insights into why some data sets are difficult to learn from and also provides a better understanding of classifier learning in general. We believe that such an understanding is critical to the development of improved classifier induction algorithms.

9.1 Introduction

It has long been observed that certain classification problems are quite difficult and that high levels of classification performance are not achievable in these cases. In certain circumstances entire classes of problems tend to be difficult, such as classification problems that deal with class imbalance [18]. These problems have often been studied in detail and sometimes methods have even been proposed for improving classification performance, but generally there is little explanation for why these techniques work and the research instead relies on empirical evaluations of the methods. As just one example, most of the research aimed at improving the performance

Gary M. Weiss
Fordham University, Bronx, NY 10458, USA, e-mail: gweiss@cis.fordham.edu

R. Stahlbock et al. (eds.), *Data Mining,* Annals of Information Systems 8,
DOI 10.1007/978-1-4419-1280-0_9,

of classifiers induced from imbalanced data sets provides little or no justification for the methods. In this chapter we focus on the role of small disjuncts in classifier learning and in so doing provide the terms and concepts necessary to provide these justifications. Additionally, we provide a number of conclusions about what makes classifier learning hard and under what circumstances.

Classifier induction programs often express the learned classifier as a disjunction. For example, such systems often express the classifier as a decision tree or a rule set, in which case each leaf in the decision tree or rule in the rule set corresponds to a disjunct. The *size* of a disjunct is defined as the number of training examples that the disjunct correctly classifies [9]. A number of empirical studies have shown that learned concepts include disjuncts that span a wide range of disjunct sizes and that small disjuncts – those disjuncts that correctly classify only a few training examples – collectively cover a significant percentage of the total test examples. These studies also show that small disjuncts have a much higher error rate than large disjuncts, a phenomenon sometimes referred to as the "problem with small disjuncts" and that these small disjuncts collectively contribute a significant portion of the total test errors.

One problem with past studies is that each study analyzes classifiers induced from only a few data sets. In particular, Holte et al. [9] analyze two data sets, Ali and Pazzani [1] one data set, Danyluk and Provost [8] one data set, Weiss [17] two data sets, Weiss and Hirsh [19] two data sets, and Carvalho and Freitas [3] two data sets. Because of the small number of data sets analyzed, and because there was no established way to measure the degree to which errors were concentrated toward the small disjuncts, these studies were not able to quantify the problem with small disjuncts. This chapter addresses these concerns. First, a new metric, error concentration, is introduced which quantifies, in a single number, the extent to which errors are concentrated toward the smaller disjuncts. This metric is then used to measure the error concentration of the classifiers induced from 30 data sets. Because we analyze a large number of data sets, we are able to draw general conclusions about the role that small disjuncts play in classifier learning.

Small disjuncts are of interest because they are responsible for many – if not most – of the errors that result when the induced classifier is applied to new (test) data. This in turn leads to two reasons for studying small disjuncts. First, we hope that what we learn about small disjuncts may enable us to build more effective classifier induction programs by addressing the problem with small disjuncts. Specifically, such learners would improve the classification performance of the examples covered by the small disjuncts without excessively degrading the accuracy of the examples covered by the larger disjuncts, such that the *overall* performance of the classifier is improved. Existing efforts to do just this, which are described in Section 9.9, have produced, at best, only marginal improvements. A better understanding of small disjuncts and their role in learning may be necessary before further advances are possible.

The second reason for studying small disjuncts is to provide a better understanding of small disjuncts and, by extension, of classifier learning in general. Most of the research on small disjuncts has not focused on this, which is the main focus of this

chapter. Essentially, small disjuncts are used as a lens through which to examine factors that are important to classifier learning, which is perhaps the most common data mining method. Pruning, training set size, noise, and class imbalance are each analyzed to see how they affect small disjuncts and the distribution of errors throughout the disjuncts – and, more generally, how this impacts classifier learning.

This chapter is an expanded version of an earlier paper [20]. It is organized as follows. In Section 9.2 we analyze the role of small disjuncts in classifier learning and introduce relevant metrics and terminology. Section 9.3 then describes the methodology used to conduct our experiments. Our experimental results and the analysis of these results are then presented in the next five sections. We provide a general analysis of the impact that small disjuncts have on learning in Section 9.4 and then, over the next four sections, we then analyze how each of the following factors interact with small disjuncts during the learning process: pruning (Section 9.5), training set size (Section 9.6), noise (Section 9.7), and class imbalance (Section 9.8). Related work is covered in Section 9.9 and our conclusions and future work are discussed in Section 9.10.

9.2 An Example: The Vote Data Set

In order to illustrate the problem with small disjuncts, the performance of a classifier induced by C4.5 [14] from the vote data set is shown in Fig. 9.1. This figure shows how the correctly and incorrectly classified test examples are distributed across the disjuncts in the induced classifier. The overall test set error rate for the classifier is 6.9%.

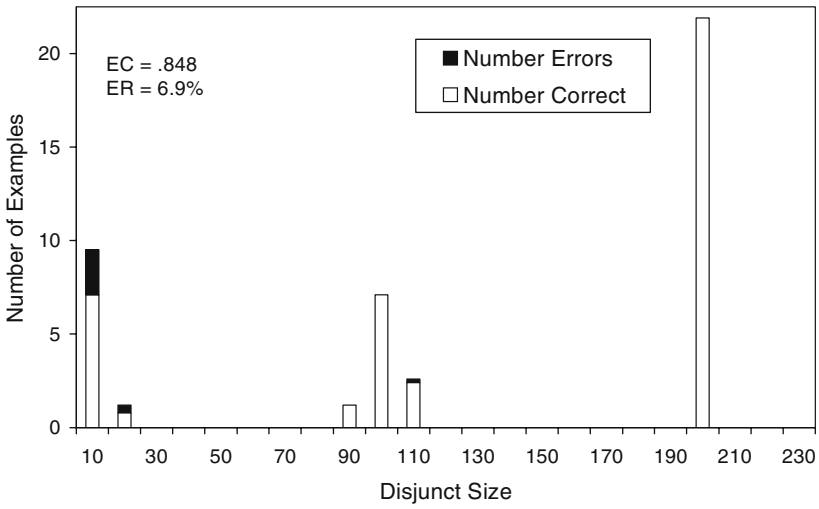

Fig. 9.1 Distribution of examples for vote data set

Each bar in the histogram in Fig. 9.1 covers 10 sizes of disjuncts. The leftmost bin shows that those disjuncts that correctly classify 0–9 training examples cover 9.5 test examples, of which 7.1 are classified correctly and 2.4 classified incorrectly (fractional values occur because the results are averaged over 10 cross-validated runs). Figure 9.1 clearly shows that the errors are concentrated toward the smaller disjuncts. Analysis at a finer level of granularity shows that the errors are skewed even more toward the small disjuncts –75% of the errors in the leftmost bin come from disjuncts of size 0 and 1. One may also be interested in the distribution of disjuncts by its size. The classifier associated with Fig. 9.1 is made up of 50 disjuncts, of which 45 are associated with the leftmost bin (i.e., have a disjunct size less than 10). Note that disjuncts of size 0 were formed because when the decision tree learner used to generate the classifier splits a node N using a feature f, the split will branch on all possible values of f – even if a feature value does not occur within the training data at N.

In order to more effectively show the extent to which errors are concentrated toward the small disjuncts, we plot the percentage of total test errors vs. the percentage of correctly classified test examples contributed by a set of disjuncts. The curve in Fig. 9.2 is generated by starting with the smallest disjunct from the classifier induced from the vote data set and then progressively adding larger disjuncts. This curve shows, for example, that disjuncts with size 0–4 cover 5.1% of the correctly classified test examples but 73% of the total test errors. The line $Y = X$ represents a classifier in which classification errors are distributed uniformly across the disjuncts, independent of the size of the disjunct. Since the "error concentration" curve in Fig. 9.2 falls above the line $Y = X$, the errors produced by this classifier are more concentrated toward the smaller disjuncts than to the larger disjuncts.

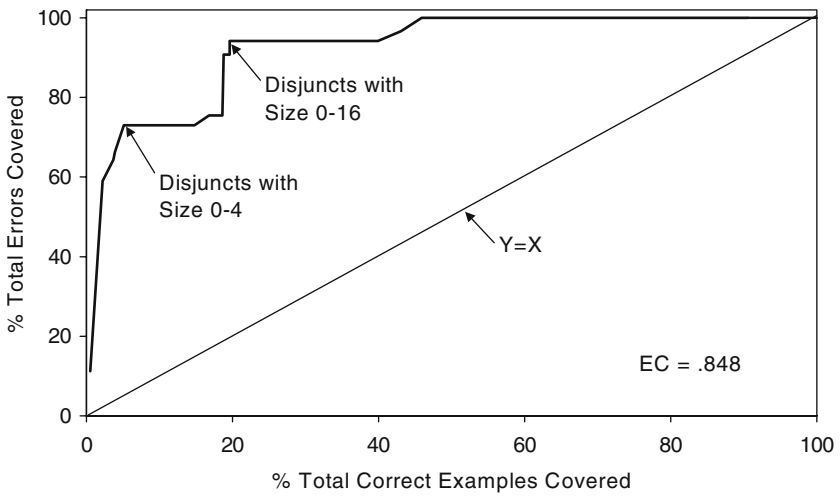

Fig. 9.2 Error concentration curve for the Vote data set

To make it easy to compare the degree to which errors are concentrated toward the smaller disjuncts for different classifiers, we introduce the *error concentration* (EC) metric. The error concentration of a classifier is defined as the fraction of the total area *above* the line $Y = X$ that falls below its error concentration curve. Using this scheme, the higher the error concentration, the more concentrated the errors are toward the smaller disjuncts. Error concentration may range from a value of +1, which indicates that all test errors are contributed by the smallest disjuncts, before even a single correctly classified test example is covered, to a value of –1, which indicates that all test errors are contributed by the largest disjuncts, after all correctly classified test examples are covered. Based on previous research, which indicates that small disjuncts have higher error rates than large disjuncts, one would expect the error concentration of most classifiers to be greater than 0. The error concentration for the classifier described in Fig. 9.2 is 0.848, indicating that the errors are highly concentrated toward the small disjuncts.

9.3 Description of Experiments

The majority of results presented in this chapter are based on an analysis of 30 data sets, of which 19 were obtained from the UCI repository [2] and 11, identified with a "+," were obtained from researchers at AT&T [6, 7]. These data sets are summarized in Table 9.1.

Table 9.1 Description of 30 data sets

No.	Data set	Size	No.	Data set	Size
1	adult	21,280	16	market1+	3180
2	bands	538	17	market2+	11,000
3	blackjack+	15,000	18	move+	3028
4	breast-wisc	699	19	network1+	3577
5	bridges	101	20	network2+	3826
6	coding	20,000	21	ocr+	2688
7	crx	690	22	promoters	106
8	german	1000	23	sonar	208
9	heart-hungarian	293	24	soybean-large	682
10	hepatitis	155	25	splice-junction	3175
11	horse-colic	300	26	ticket1+	556
12	hypothyroid	3771	27	ticket2+	556
13	kr-vs-kp	3196	28	ticket3+	556
14	labor	57	29	vote	435
15	liver	345	30	weather+	5597

Numerous experiments are run on these data sets to assess the impact that small disjuncts have on learning. The majority of the experimental results presented in this chapter are based on C4.5 [14], a popular program for inducing decision trees. C4.5 was modified by the author to collect a variety of information related to disjunct

size. Note that disjunct size is defined based on the number of examples covered by the training data but, as is typical in data mining, the classification results are measured based on the performance on the test data. Many experiments were repeated using Ripper [6], a program for inducing rule sets, to ensure the generality of our results. Because Ripper exports detailed information about the performance of individual rules, internal modifications to the program were not required in order to track the statistics related to disjunct size. All experiments for both learners employ 10-fold cross-validation and all results are based on the averages over these 10 runs. Pruning tends to eliminate most small disjuncts and, for this reason, research on small disjuncts generally disables pruning [8, 9, 17, 19]. If this were not done, then pruning would mask the problem with small disjuncts. While this means that the analyzed classifiers are not the same as the ones that would be generated using the learners in their standard configurations, these results are nonetheless important, since the performance of the unpruned classifiers constrains the performance of the pruned classifiers. However, in this chapter both unpruned and pruned classifiers are analyzed, for both C4.5 and Ripper. This makes it possible to analyze the effect that pruning has on small disjuncts and to evaluate pruning as a strategy for addressing the problem with small disjuncts. As the results for pruning in Section 9.5 will show, the problem with small disjuncts is still evident after pruning, although to a lesser extent.

All results, other than those described in Section 9.5, are based on the use of C4.5 and Ripper with their pruning strategies disabled. For C4.5, when pruning is disabled the –m 1 option is also used, to ensure that C4.5 does not stop splitting a node before the node contains examples belonging to a single class (the default is –m 2). Ripper is configured to produce unordered rules so that it does not produce a single default rule to cover the majority class.

9.4 The Problem with Small Disjuncts

Previous research claims that errors tend to be concentrated most heavily in the smaller disjuncts [1, 3, 8, 9, 15, 17, 19]. In this section we provide the most comprehensive analysis of this claim to date, by measuring the degree to which errors are concentrated toward the smaller disjuncts for the 30 data sets listed in Table 9.1, for classifiers induced by C4.5 and Ripper.

The experimental results for C4.5 and Ripper, in order of decreasing error concentration, are displayed in Tables 9.2 and 9.3, respectively. In addition to specifying the error concentration, these tables also list the error rate of the induced classifier, the size of the data set, and the size of the largest disjunct in the induced classifier. They also specify the percentage of the total test errors that are contributed by the smallest disjuncts that collectively cover 10% of the correctly classified test examples and then the percentage of the total correctly classified examples that are covered by the smallest disjuncts that collectively cover half of the total errors.

Table 9.2 Error concentration results for C4.5

EC rank	Data set rame	Error rate	Data set size	Largest disjunct	% Errs at 10% correct	% Correct at 50% errors	Error conc.
1	kr-vs-kp	0.3	3196	669	75.0	1.1	0.874
2	hypothyroid	0.5	3771	2697	85.2	0.8	0.852
3	vote	6.9	435	197	73.0	1.9	0.848
4	splice-junction	5.8	3175	287	76.5	4.0	0.818
5	ticket2	5.8	556	319	76.1	2.7	0.758
6	ticket1	2.2	556	366	54.8	4.4	0.752
7	ticket3	3.6	556	339	60.5	4.6	0.744
8	soybean-large	9.1	682	56	53.8	9.3	0.742
9	breast-wisc	5.0	699	332	47.3	10.7	0.662
10	ocr	2.2	2688	1186	52.1	8.9	0.558
11	hepatitis	22.1	155	49	30.1	17.2	0.508
12	horse-colic	16.3	300	75	31.5	18.2	0.504
13	crx	19.0	690	58	32.4	14.3	0.502
14	bridges	15.8	101	33	15.0	23.2	0.452
15	heart-hungar.	24.5	293	69	31.7	21.9	0.450
16	market1	23.6	3180	181	29.7	21.1	0.440
17	adult	16.3	21,280	1441	28.7	21.8	0.424
18	weather	33.2	5597	151	25.6	22.4	0.416
19	network2	23.9	3826	618	31.2	24.2	0.384
20	promoters	24.3	106	20	32.8	20.6	0.376
21	network1	24.1	3577	528	26.1	24.1	0.358
22	german	31.7	1000	56	17.8	29.4	0.356
23	coding	25.5	20,000	195	22.5	30.9	0.294
24	move	23.5	3028	35	17.0	30.8	0.284
25	sonar	28.4	208	50	15.9	32.9	0.226
26	bands	29.0	538	50	65.2	54.1	0.178
27	liver	34.5	345	44	13.7	40.3	0.120
28	blackjack	27.8	15,000	1989	18.6	39.3	0.108
29	labor	20.7	57	19	33.7	49.1	0.102
30	market2	46.3	11,000	264	10.3	45.5	0.040

As an example of how to interpret the results in these tables, consider the entry for the kr-vs-kp data set in Table 9.2. The error concentration for the classifier induced from this data set is 0.874. Furthermore, the smallest disjuncts that collectively cover 10% of the correctly classified test examples contribute 75% of the total test errors, while the smallest disjuncts that contribute half of the total errors cover only 1.1% of the total correctly classified examples. These measurements provide a concrete indication of just how concentrated the errors are toward the smaller disjuncts.

The results for C4.5 and Ripper show that although the error concentration values are, as expected, almost always positive, the values vary widely, indicating that the induced classifiers suffer from the problem of small disjuncts to varying degrees. The classifiers induced using Ripper have a slightly smaller average error concentration than those induced using C4.5 (0.445 vs. 0.471), indicating that the classifiers induced by Ripper have the errors spread slightly more uniformly across the disjuncts. Overall, Ripper and C4.5 tend to generate classifiers with similar error concentration values. This can be seen by comparing the EC rank in Table 9.3

Table 9.3 Error concentration results for Ripper

EC rank	C4.5 rank	Data set name	Error rate	Data set size	Largest disjunct	% Errors 10% correct	% Correct 50% Errs	Error conc.
1	2	hypothyroid	1.2	3771	2696	96.0	0.1	0.898
2	1	kr-vs-kp	0.8	3196	669	92.9	2.2	0.840
3	6	ticket1	3.5	556	367	69.4	1.6	0.802
4	7	ticket3	4.5	556	333	61.4	5.6	0.790
5	5	ticket2	6.8	556	261	71.0	3.2	0.782
6	3	vote	6.0	435	197	75.8	3.0	0.756
7	4	splice-junction	6.1	3175	422	62.3	7.9	0.678
8	9	breast-wisc	5.3	699	355	68.0	3.6	0.660
9	8	soybean-large	11.3	682	61	69.3	4.8	0.638
10	10	ocr	2.6	2688	804	50.5	10.0	0.560
11	17	adult	19.7	21,280	1488	36.9	15.0	0.516
12	16	market1	25.0	3180	243	32.2	16.9	0.470
13	12	horse-colic	22.0	300	73	20.7	23.9	0.444
14	13	crx	17.0	690	120	32.5	19.7	0.424
15	15	heart-hungar.	23.9	293	67	25.8	24.8	0.390
16	26	bands	21.9	538	62	25.6	29.2	0.380
17	25	sonar	31.0	208	47	32.6	23.9	0.376
18	23	coding	28.2	20,000	206	22.6	29.2	0.374
19	18	weather	30.2	5597	201	23.8	24.8	0.356
20	24	move	32.1	3028	45	25.9	25.6	0.342
21	14	bridges	14.5	101	39	41.7	35.5	0.334
22	20	promoters	19.8	106	24	20.0	20.0	0.326
23	11	hepatitis	20.3	155	60	19.3	20.8	0.302
24	22	german	30.8	1000	99	12.1	35.0	0.300
25	19	network2	23.1	3826	77	25.6	22.9	0.242
26	27	liver	34.0	345	28	28.2	32.0	0.198
27	28	blackjack	30.2	15,000	1427	12.3	42.3	0.0108
28	21	network1	23.4	3577	79	18.9	46.0	0.090
29	29	labor	24.5	57	21	0.0	18.3	0.006
30	30	market2	48.8	11,000	55	10.4	49.8	0.018

for Ripper (column 1) with the EC rank for C4.5 (column 2), which is displayed graphically in the scatter plot in Fig. 9.3, where each point represents the error concentration for a single data set. Since the points in Fig. 9.3 are clustered around the line $Y = X$, both learners tend to produce classifiers with similar error concentrations and hence tend to suffer from the problem with small disjuncts to similar degrees. The agreement is especially close for the most interesting cases, where the error concentrations are large – the largest 10 error concentration values in Fig. 9.3, for both C4.5 and Ripper, are generated by the same 10 data sets.

With respect to classification accuracy, the two learners perform similarly, although C4.5 performs slightly better (it outperforms Ripper on 18 of the 30 data sets, with an average error rate of 18.4% vs. 19.0%). However, as will be shown in the next section, when pruning is used Ripper slightly outperforms C4.5.

The results in Tables 9.2 and 9.3 indicate that, for both C4.5 and Ripper, there is a relationship between the error rate and error concentration of the induced classifiers.

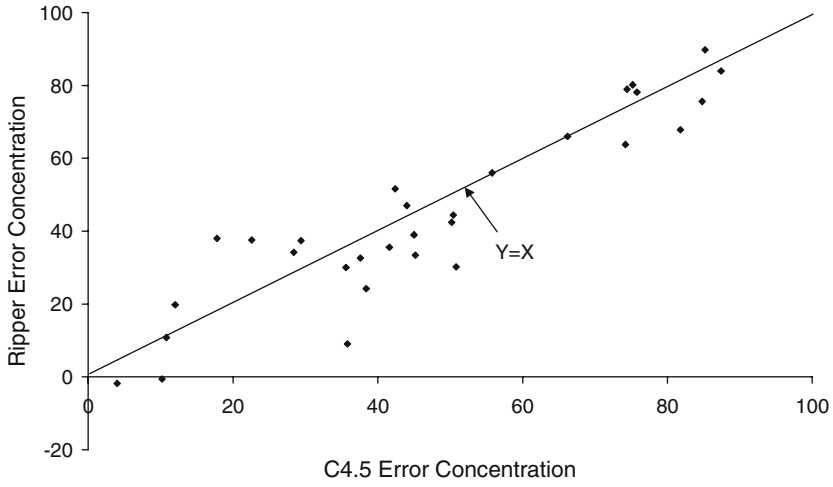

Fig. 9.3 Comparison of C4.5 and Ripper error concentration values

These results show that, for the 30 data sets, when the induced classifier has an error rate less than 12%, then the error concentration is always greater than 0.50. Based on the error rate and error concentration values, the induced classifiers seem to fit naturally into the following three categories:

1. High EC/moderate ER data sets 1–10 for C4.5 and Ripper
2. Medium EC/high ER data sets 11–22 for C4.5; 11–24 for Ripper
3. Low EC/high ER data sets 23–30 for C4.5; 25–30 for Ripper

It is interesting to note that for those data sets in the high-EC/moderate-ER category, the largest disjunct generally covers a very large portion of the total training examples. As an example, consider the hypothyroid data set. Of the 3394 examples (90% of the total data) used for training, nearly 2700 of these examples, or 79%, are covered by the largest disjunct induced by C4.5 and Ripper. To see that these large disjuncts are extremely accurate, consider the vote data set, which falls within the same category. The distribution of errors for the vote data set was shown previously in Fig. 9.1. The data used to generate this figure indicates that the largest disjunct, which covers 23% of the total training examples, does not contribute a single error when used to classify the test data. These observations lead us to speculate that concepts that can be learned well (i.e., have low error rates) are often made up of very general cases that lead to highly accurate large disjunct – and therefore to classifiers with very high error concentrations. Concepts that are difficult to learn, on the other hand, either are not made up of very general cases or, due to limitations with the expressive power of the learner, these general cases cannot be represented using large disjuncts. This leads to classifiers without very large, highly accurate disjuncts and with many small disjuncts. These classifiers tend to have much smaller error concentrations.

9.5 The Effect of Pruning on Small Disjuncts

The results in the previous section, consistent with previous research on small disjuncts, were generated using C4.5 and Ripper with their pruning strategies disabled. Pruning is generally not used when studying small disjuncts because of the belief that it disproportionately eliminates small disjuncts from the induced classifier and thereby obscures the very phenomenon we wish to study. However, because pruning is employed by many learning systems, it is worthwhile to understand how it affects small disjuncts and the distribution of errors across disjuncts – as well as how effective it is at addressing the problem with small disjuncts. In this section we investigate the effect of pruning on the distribution of errors across the disjuncts in the induced classifier. We begin with an illustrative example. Figure 9.4 shows the distribution of errors for the classifier induced from the vote data set using C4.5 with pruning. This distribution can be compared to the corresponding distribution in Fig. 9.1 that was generated using C4.5 without pruning, to show the effect that pruning has on the distribution of errors.

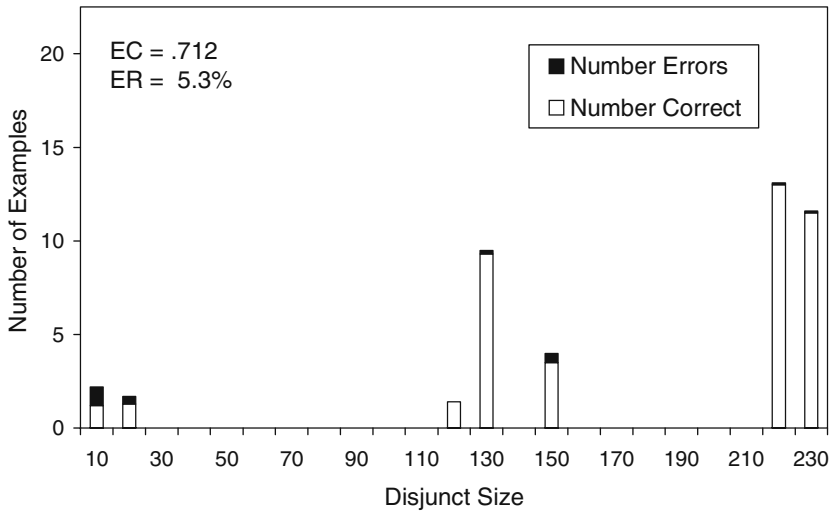

Fig. 9.4 Distribution of examples with pruning for the vote data set

A comparison of Figs. 9.4 with 9.1 shows that with pruning the errors are less concentrated in the small disjuncts. This is also confirmed by the error concentration value, which is reduced from 0.848 to 0.712. It is also apparent that with pruning far fewer examples are classified by disjuncts with size 0–9 and 10–19. The underlying data indicates that without pruning the induced classifiers typically (i.e., over the 10 runs) contain 48 disjuncts, of which 45 are of size 10 or less, while with pruning only 10 disjuncts remain, of which 7 have size 10 or less. So, in this case pruning eliminates 38 of the 45 disjuncts with size 10 or less. This confirms the assumption

that pruning eliminates many, if not most, small disjuncts. The emancipated examples – those that would have been classified by the eliminated disjuncts – are now classified by larger disjuncts. It should be noted, however, that even with pruning the error concentration is still quite positive (0.712), indicating that the errors still tend to be concentrated toward the small disjuncts. In this case pruning also causes the overall error rate of the classifier to decrease from 6.9 to 5.3%.

The performance of the classifiers induced from the 30 data sets, using C4.5 and Ripper with their default pruning strategies, is presented in Tables 9.4 and 9.5, respectively. The induced classifiers are again placed into three categories, although in this case the patterns that were previously observed are not nearly as evident. In particular, with pruning some classifiers continue to have low error rates but no longer have large error concentrations (e.g., ocr, soybean-lg, and ticket3 for C4.5 only). In these cases pruning has caused the rarely occurring classification errors to be distributed much more uniformly throughout the disjuncts.

Table 9.4 Error concentration results for C4.5 with pruning

EC rank	Data set	Error rate	Data set size	Largest disjunct	% Errors 10% correct	% Correct 50% errors	Error conc.
1	hypothyroid	0.5	3771	2732	90.7	0.7	0.818
2	ticket1	1.6	556	410	46.7	10.3	0.730
3	vote	5.3	435	221	68.7	2.9	0.712
4	breast-wisc	4.9	699	345	49.6	10.0	0.688
5	kr-vs-kp	0.6	3196	669	35.4	15.6	0.658
6	splice-junction	4.2	3175	479	41.6	25.9	0.566
7	crx	15.1	690	267	45.2	11.5	0.516
8	ticket2	4.9	556	442	48.1	12.8	0.474
9	weather	31.1	5597	573	26.2	22.2	0.442
10	adult	14.1	21,280	5018	36.6	17.6	0.424
11	german	28.4	1000	313	29.6	21.9	0.404
12	soybean-large	8.2	682	61	48.0	14.4	0.394
13	network2	22.2	3826	1685	30.8	21.2	0.362
14	ocr	2.7	2688	1350	40.4	34.3	0.348
15	market1	20.9	3180	830	28.4	23.6	0.336
16	network1	22.4	3577	1470	24.4	27.2	0.318
17	ticket3	2.7	556	431	37.0	20.9	0.310
18	horse-colic	14.7	300	137	35.8	19.3	0.272
19	coding	27.7	20,000	415	17.2	34.9	0.216
20	sonar	28.4	208	50	15.1	34.6	0.202
21	heart-hung.	21.4	293	132	19.9	31.8	0.198
22	hepatitis	18.2	155	89	24.2	26.3	0.168
23	liver	35.4	345	59	17.6	34.8	0.162
24	promoters	24.4	106	26	17.2	37.0	0.128
25	move	23.9	3028	216	14.4	42.9	0.094
26	blackjack	27.6	15,000	3053	16.9	44.7	0.092
27	labor	22.3	57	24	14.3	40.5	0.082
28	bridges	15.8	101	67	14.9	50.1	0.064
29	market2	45.1	11,000	426	12.2	44.7	0.060
30	bands	30.1	538	279	0.8	58.3	0.184

Table 9.5 Error concentration results for Ripper with pruning

EC rank	C4.5 rank	Data set	Error rate	Data set size	Largest disjunct	% Errors 10% correct	% Correct 50% errs	Error conc.
1	1	hypothyroid	0.9	3771	2732	97.2	0.6	0.930
2	5	kr-vs-kp	0.8	3196	669	56.8	5.4	0.746
3	2	ticket1	1.6	556	410	41.5	11.9	0.740
4	6	splice-junction	5.8	3175	552	46.9	10.7	0.690
5	3	vote	4.1	435	221	62.5	2.8	0.648
6	8	ticket2	4.5	556	405	73.3	7.8	0.574
7	17	ticket3	4.0	556	412	71.3	9.0	0.516
8	14	ocr	2.7	2688	854	29.4	24.5	0.306
9	20	sonar	29.7	208	59	23.1	25.4	0.282
10	30	bands	26.0	538	118	22.1	24.0	0.218
11	9	weather	26.9	5597	1148	18.8	35.4	0.198
12	23	liver	32.1	345	69	13.6	34.7	0.146
13	12	soybean-large	9.8	682	66	17.8	47.4	0.128
14	11	german	29.4	1000	390	14.7	32.4	0.128
15	4	breast-wisc	4.4	699	370	14.4	31.4	0.124
16	15	market1	21.3	3180	998	19.0	43.4	0.114
17	7	crx	15.1	690	272	16.4	39.1	0.108
18	13	network2	22.6	3826	1861	15.3	39.5	0.090
19	16	network1	23.3	3577	1765	16.0	42.0	0.090
20	18	horse-colic	15.7	300	141	13.8	36.6	0.086
21	21	hungar-heart	18.8	293	138	17.9	42.6	0.072
22	19	coding	28.3	20,000	894	12.7	46.5	0.052
23	26	blackjack	28.1	15,000	4893	16.8	45.3	0.040
24	22	hepatitis	22.3	155	93	25.5	57.2	0.004
25	29	market2	40.9	11,000	2457	7.7	50.2	0.016
26	28	bridges	18.3	101	71	19.1	55.0	0.024
27	25	move	24.1	3028	320	10.9	63.1	0.094
28	10	adult	15.2	21,280	9293	9.8	67.9	0.146
29	27	labor	18.2	57	25	0.0	70.9	0.228
30	24	promoters	11.9	106	32	0.0	54.1	0.324

The results in Tables 9.4 and 9.5, when compared to the results in Tables 9.2 and 9.3, show that pruning tends to reduce the error concentration of most classifiers. This is shown graphically by the scatter plot in Fig. 9.5. Since most of the points fall below the line $Y = X$, we conclude that for both C4.5 and Ripper, pruning, as expected, tends to reduce error concentration. However, Fig. 9.5 makes it clear that pruning has a more dramatic impact on the error concentration for classifiers induced using Ripper than those induced using C4.5. Pruning causes the error concentration to decrease for 23 of the 30 data sets for C4.5 and for 26 of the 30 data sets for Ripper. More significant, however, is the magnitude of the changes in error concentration. On average, pruning causes the error concentration for classifiers induced using C4.5 to drop from 0.471 to 0.375, while the corresponding drop when using Ripper is from 0.445 to 0.206. These results indicate that the pruned classifiers produced by Ripper have the errors much less concentrated toward the small disjuncts than those produced by C4.5. Given that Ripper is generally known

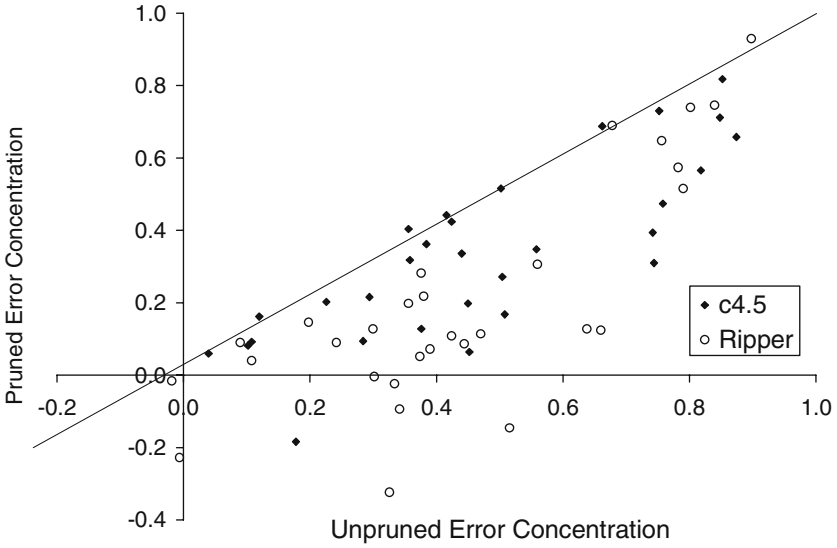

Fig. 9.5 Effect of pruning on error concentration

to produce very simple rule sets, this larger decrease in error concentration is likely due to the fact that Ripper has a more aggressive pruning strategy than C4.5.

The results in Tables 9.4 and 9.5 and in Fig. 9.5 indicate that, even with pruning, the "problem with small disjuncts" is still quite evident for both C4.5 and Ripper. For both learners the error concentration, averaged over the 30 data sets, is still decidedly positive. Furthermore, even with pruning both learners produce many classifiers with error concentrations greater than 0.50. However, it is certainly worth noting that with pruning, seven of the classifiers induced by Ripper have *negative* error concentrations. Comparing the error concentration values for Ripper with and without pruning reveals one particularly interesting example. For the adult data set, pruning causes the error concentration to drop from 0.516 to –0.146. This large change likely indicates that many error-prone small disjuncts are eliminated. This is supported by the fact that the size of the largest disjunct in the induced classifier changes from 1488 without pruning to 9293 with pruning. Thus, pruning seems to have an enormous effect on this Ripper classifier.

The effect that pruning has on error rate is shown graphically in Fig. 9.6 for both C4.5 and Ripper. Because most of the points in Fig. 9.6 fall below the line $Y = X$, we conclude that pruning tends to reduce the error rate for both C4.5 and Ripper. However, the figure also makes it clear that pruning improves the performance of Ripper more than it improves the performance of C4.5. In particular, for C4.5 pruning causes the error rate to drop for 19 of the 30 data sets while for Ripper pruning causes the error rate to drop for 24 of the 30 data sets. Over the 30 data sets pruning causes C4.5's error rate to drop from 18.4 to 17.5% and Ripper's error rate to drop from 19.0 to 16.9%.

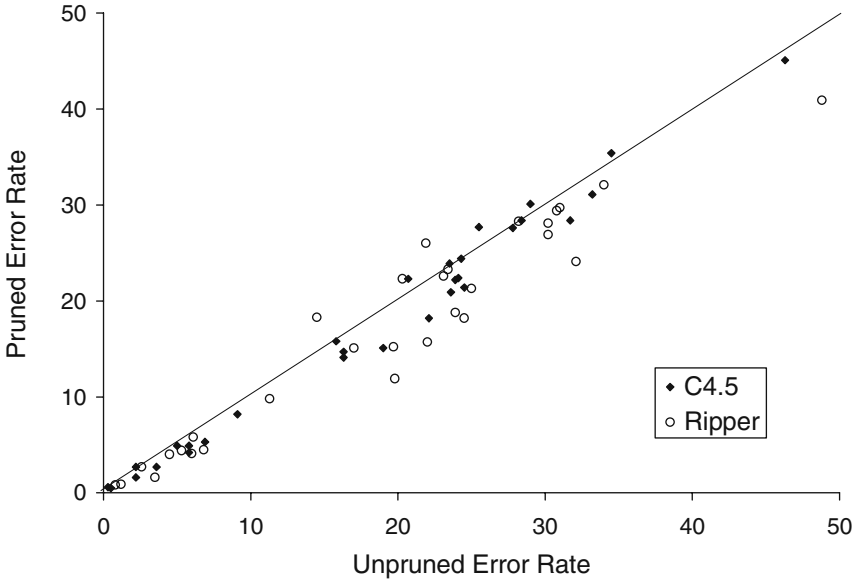

Fig. 9.6 Effect of pruning on error rate

Given that pruning tends to affect small disjuncts more than large disjuncts, an interesting question is whether pruning is more effective at reducing error rate when the errors in the unpruned classifier are most highly concentrated in the small disjuncts. Figure 9.7 addresses this by plotting the absolute reduction in error rate due to pruning vs. the error concentration rank of the unpruned classifier. The data sets with high and medium error concentrations show a fairly consistent reduction in error rate.[1] Finally, the classifiers in the low-EC/high-ER category show a net *increase* in error rate. These results suggest that pruning is most beneficial when the errors are most highly concentrated in the small disjuncts – and may actually hurt when this is not the case. The results for Ripper show a somewhat similar pattern, although the unpruned classifiers with low error concentrations do consistently show some reduction in error rate when pruning is used.

The results in this section show that pruned classifiers generally have lower error rates and lower error concentrations than their unpruned counterparts. Our analysis shows us that for the vote data set this change is due to the fact that pruning eliminates most small disjuncts. A similar analysis, performed for other data sets in this study, shows a similar pattern – pruning eliminates mostsmall disjuncts. In

[1] Note that although the classifiers in the medium-EC/high-ER category show a greater absolute reduction in error rate than those in the high-EC/moderate-ER group, this corresponds to a smaller relative reduction in error rate, due to the differences in the error rate of the unpruned classifiers.

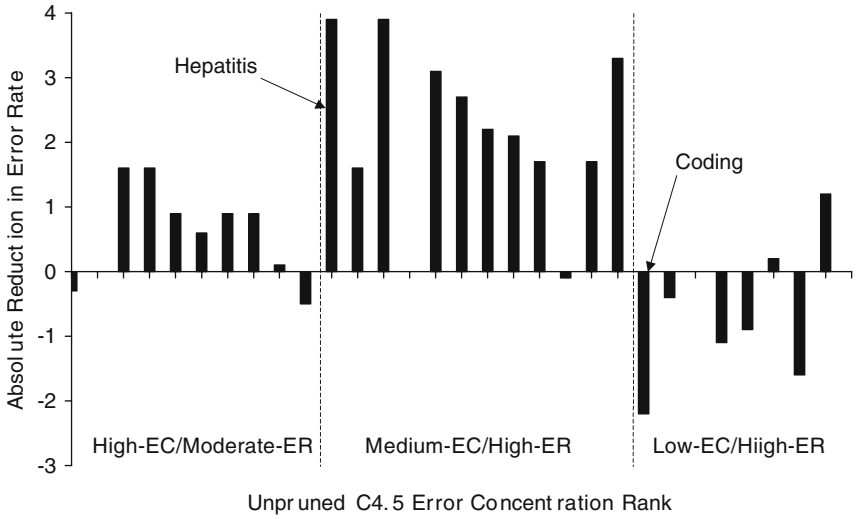

Fig. 9.7 Improvement in error rate versus error concentration rank

summary, pruning is a strategy for dealing with the "problem of small disjuncts." Pruning eliminates many small disjuncts and the emancipated examples that would have been classified by the eliminated disjuncts are then classified by other, typically much larger, disjuncts. The result of pruning is that there is a decrease in the average error rate of the induced classifiers and the remaining errors are more uniformly distributed across the disjuncts.

One can gauge the effectiveness of pruning as a strategy for addressing the problem with small disjuncts by comparing it to an "ideal" strategy that causes the error rate of the small disjuncts to equal the error rate of the larger disjuncts. Table 9.6 shows the average error rates of the classifiers induced by C4.5 for the 30 data sets, without pruning, with pruning, and with two variants of this idealized strategy. The error rates for the idealized strategies are determined by first identifying the smallest disjuncts that collectively cover 10% (20%) of the training examples and then calculating the error rate of the classifier as if the error rate of these small disjuncts equaled the error rate of the examples classified by all of the other disjuncts.

Table 9.6 Comparison of pruning to idealized strategy

	Strategy			
	No Pruning (%)	Pruning (%)	Idealized (10%)	Idealized (20%)
Average error rate	18.4	17.5	15.2%	13.5%
Relative improvement		4.9	17.4%	26.6%

The results in Table 9.6 show that the idealized strategy yields much more dramatic improvements in error rate than pruning, even when it is only applied to the disjuncts that cover 10% of the training examples. This indicates that pruning is not very effective at addressing the problem with small disjuncts and provides a strong motivation for finding better strategies for handling small disjuncts (several such strategies are discussed in Section 9.9). Note, however, that we are not suggesting that the performance of the idealized strategies can necessarily ever be realized.

For many real-world problems, it is more important to classify a reduced set of examples with high precision than in finding the classifier with the best overall accuracy. For example, if the task is to identify customers likely to buy a product in response to a direct marketing campaign, it may be impossible to utilize all classifications – budgetary concerns may permit one to only contact the 10,000 people most likely to make a purchase. Given that our results indicate that pruning *decreases* the precision of the larger, more precise disjuncts (compare Figs. 9.1 and 9.4), this suggests that pruning may be harmful in such cases – even though pruning leads to an overall increase in the accuracy of the induced classifier. To investigate this further, classifiers were generated by starting with the largest disjunct and then progressively adding smaller disjuncts. A classification decision is made only if an example is covered by one of the added disjuncts; otherwise no classification is made. The error rate (i.e., precision) of the resulting classifiers, generated with and without pruning, is shown in Table 9.7, as is the difference in error rates. A negative difference indicates that pruning leads to an improvement (i.e., a reduction) in error rate, while a positive difference indicates that pruning leads to an increase in error rate. Results are reported for classifiers with disjuncts that collectively cover 10, 30, 50, 70, and 100% of the training examples.

The last row in Table 9.7 shows the error rates averaged over the 30 data sets. These results clearly show that, over the 30 data sets, pruning only helps for the last column – when all disjuncts are included in the evaluated classifier. Note that these results, which correspond to the accuracy results presented earlier, are typically the only results that are described. This leads to an overly optimistic view of pruning, since in other cases pruning results in a *higher* overall error rate. As a concrete example, consider the case where we use only the disjuncts that collectively cover 50% of the training examples. In this case C4.5 with pruning generates classifiers with an average error rate of 12.9% whereas C4.5 without pruning generates classifiers with an average error rate of 11.4%. Looking at the individual results for this situation, pruning does worse for 17 of the data sets, better for 9 of the data sets, and the same for 4 of the data sets. However, the magnitude of the differences is much greater in the cases where pruning performs worse.

The results from the last row of Table 9.7 are displayed graphically in Fig. 9.8, which plots the error rates, with and without pruning, averaged over the 30 data sets. Note, however, that unlike the results in Table 9.7, Fig. 9.8 shows classifier performance at each 10% increment.

Figure 9.8 clearly demonstrates that under most circumstances pruning does *not* produce the best results. While it produces marginally better results when predictive

Table 9.7 Effect of pruning when classification based only on largest disjuncts

Data set name	Error rate with pruning (Yes) and without pruning (No)											
	10% covered			30% covered			70% covered			100% covered		
Pruning used:	Yes	No	Δ	Yes	No	Δ	Yes	No	Δ	Yes	No	Δ
kr-vs-kp	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.1	0.6	0.3	0.3
hypothyroid	0.1	0.3	–0.2	0.2	0.1	0.1	0.1	0.0	0.0	0.5	0.5	0.0
vote	3.1	0.0	3.1	1.0	0.0	1.0	2.3	0.7	1.6	5.3	6.9	–1.6
splice-junction	0.3	0.9	–0.6	0.2	0.3	–0.1	2.4	0.6	1.8	4.2	5.8	–1.6
ticket2	0.3	0.0	0.3	2.7	0.8	1.9	2.5	1.0	1.5	4.9	5.8	–0.9
ticket1	0.1	2.1	–1.9	0.3	0.6	–0.3	0.3	0.3	0.0	1.6	2.2	–0.5
ticket3	2.1	2.0	0.1	1.7	1.2	0.5	1.5	0.5	1.0	2.7	3.6	–0.9
soybean-large	1.5	0.0	1.5	5.4	1.0	4.4	4.7	1.3	3.5	8.2	9.1	–0.9
breast-wisc	1.5	1.1	0.4	1.0	1.0	0.0	1.0	1.4	–0.4	4.9	5.0	–0.1
ocr	1.5	1.8	–0.3	1.9	0.8	1.1	1.9	1.0	0.9	2.7	2.2	0.5
hepatitis	5.4	6.7	–1.3	15.0	2.2	12.9	12.8	12.1	0.6	18.2	22.1	–3.9
horse-colic	20.2	1.8	18.4	14.6	4.6	10.0	10.7	10.6	0.1	14.7	16.3	–1.7
crx	7.0	7.3	–0.3	7.9	6.5	1.4	7.8	9.3	–1.6	15.1	19.0	–3.9
bridges	10.0	0.0	10.0	17.5	0.0	17.5	14.9	9.4	5.4	15.8	15.8	0.0
heart-hung.	15.4	6.2	9.2	18.4	11.4	7.0	16.0	16.4	–0.4	21.4	24.5	–3.1
market1	16.6	2.2	14.4	12.2	7.8	4.4	14.5	15.9	–1.4	20.9	23.6	–2.6
adult	3.9	0.5	3.4	3.6	4.9	–1.3	8.3	10.6	–2.3	14.1	16.3	–2.2
weather	5.4	8.6	–3.2	10.6	14.0	–3.4	22.7	24.6	–1.9	31.1	33.2	–2.1
network2	10.8	9.1	1.7	12.5	10.7	1.8	15.1	17.2	–2.1	22.2	23.9	–1.8
promoters	10.2	19.3	–9.1	10.9	10.4	0.4	19.6	16.8	2.8	24.4	24.3	0.1
network1	15.3	7.4	7.9	13.1	11.8	1.3	16.7	17.3	–0.6	22.4	24.1	–1.7
german	10.0	4.9	5.1	11.1	12.5	–1.4	20.4	25.7	–5.3	28.4	31.7	–3.3
coding	19.8	8.5	11.3	18.7	14.3	4.4	23.6	20.6	3.1	27.7	25.5	2.2
move	24.6	9.0	15.6	19.2	12.1	7.1	22.6	18.7	3.8	23.9	23.5	0.3
sonar	27.6	27.6	0.0	23.7	23.7	0.0	24.4	24.3	0.1	28.4	28.4	0.0
bands	13.1	0.0	13.1	34.3	16.3	18.0	33.8	26.6	7.2	30.1	29.0	1.1
liver	27.5	36.2	–8.8	32.4	28.1	4.3	30.7	31.8	–1.2	35.4	34.5	0.9
blackjack	25.3	26.1	–0.8	25.1	25.8	–0.8	26.1	24.4	1.7	27.6	27.8	–0.2
labor	25.0	25.0	0.0	17.5	24.8	–7.3	24.4	17.5	6.9	22.3	20.7	1.6
market2	44.1	45.5	–1.4	43.1	44.3	–1.2	43.3	45.3	–2.0	45.1	46.3	–1.2
Average	11.6	8.7	2.9	12.5	9.7	2.8	14.2	13.4	0.8	17.5	18.4	–0.9

accuracy is the evaluation metric (i.e., all examples must be classified), it produces much poorer results when one can be very selective about the classification "rules" that are used. These results confirm the hypothesis that when pruning eliminates some small disjuncts, the emancipated examples cause the error rate of the more accurate large disjuncts to decrease. The overall error rate is reduced only because the error rate for the emancipated examples is lower than their original error rate. Thus, pruning redistributes the errors such that the errors are more uniformly distributed than without pruning. This is exactly what one does not want to happen when one can be selective about which examples to classify (or which classifications to act upon). We find the fact that pruning improves only classifier performance when disjuncts covering more than 80% of the training examples are used to be quite compelling.

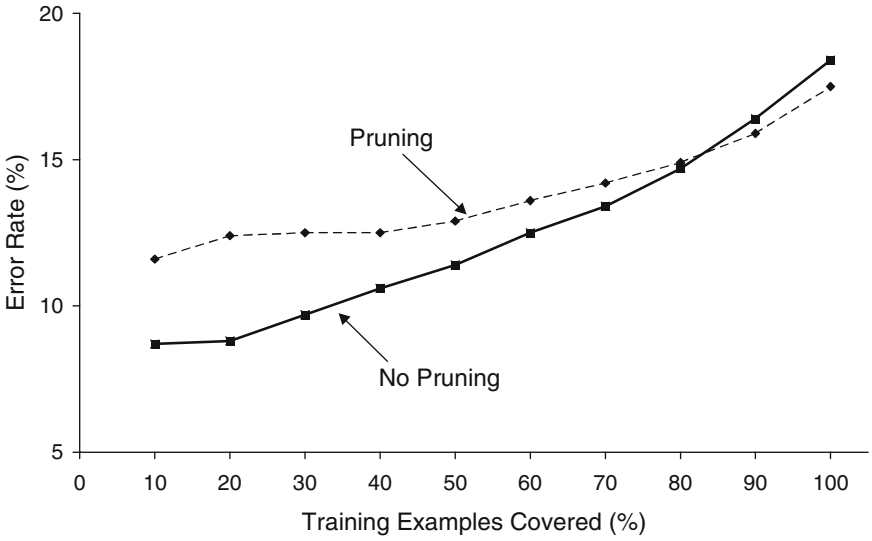

Fig. 9.8 Averaged error rate based on classifiers built from the largest disjuncts

9.6 The Effect of Training Set Size on Small Disjuncts

The amount of training data available for learning has several well-known effects. Namely, increasing the amount of training data will tend to increase the accuracy of the classifier and increase the number of "rules," as additional training data permits the existing rules to be refined. In this section we analyze the effect that training set size has on small disjuncts and error concentration.

Figure 9.9 returns to the vote data set example, but this time shows the distribution of examples and errors when the training set is limited to use only 10% of the total data. These results can be compared with those in Fig. 9.1, which are based upon 90% of the data being used for training. Thus, the results in Fig. 9.9 are based on 1/9th the training data used in Fig. 9.1. Note that the size of the bins, and consequently the scale of the *x*-axis, has been reduced in Fig. 9.9.

A comparison of the relative distribution of errors between Figs. 9.9 and 9.1 shows that errors are more concentrated toward the smaller disjuncts in Fig. 9.1, which has a higher error concentration (0.848 vs. 0.628). This indicates that increasing the amount of training data increases the degree to which the errors are concentrated toward the small disjuncts. Like the results in Fig. 9.1, the results in Fig. 9.9 show that there are three groupings of disjuncts, which one might be tempted to refer to as small, medium, and large disjuncts. The size of the disjuncts within each group differs between the two figures, due to the different number of training examples used to generate each classifier (note the change in scale of the *x*-axis). It is informative to compare the error concentrations for classifiers induced

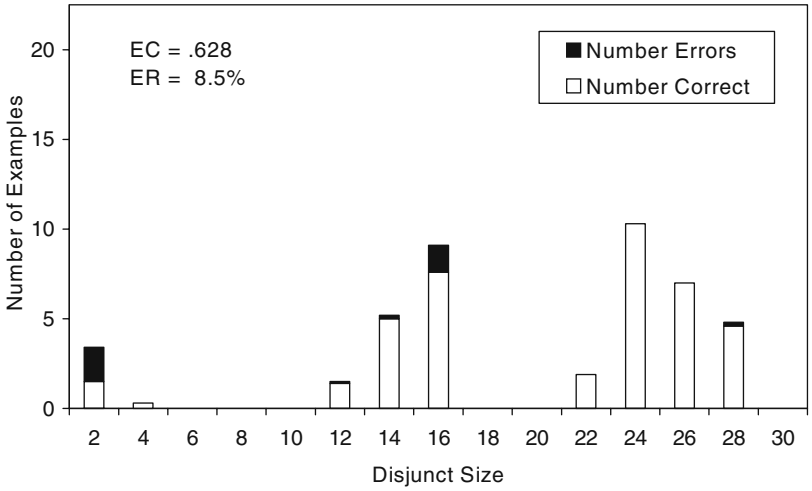

Fig. 9.9 Distribution of examples for the vote data set (using 1/9 of the normal training data)

using different training set sizes because error concentration is a relative measure – it measures the distribution of errors within the classifier relative to the disjuncts within the classifier and relative to the total number of errors produced by the classifier (which will be less when more training data is available). Summary statistics for all 30 data sets are shown in Table 9.8.

Table 9.8 shows the error rate and error concentration for the classifiers induced from each of the 30 data sets using three different training set sizes. The last two columns highlight the impact of training set size, by showing the change in error concentration and error rate that occurs when the training set size is increased by a factor of 9. As expected, the error rate tends to decrease with additional training data while the error concentration, consistent with the results associated with the vote data set, shows a consistent increase – for 27 of the 30 data sets the error concentration increases when the amount of training data is increased by a factor of 9.

The observation that an increase in training data leads to an increase in error concentration can be explained by analyzing how an increase in training data affects the classifier that is learned. As more training data becomes available, the induced classifier is better able to sample, and learn, the general cases that exist within the concept. This causes the classifier to form highly accurate large disjuncts. As an example, note that the largest disjunct in Fig. 9.1 does not cover a single error and that the medium-sized disjuncts, with sizes between 80 and 109, cover only a few errors. Their counterparts in Fig. 9.9, with size between 20 and 27 and 10 and 15, have a higher error rate. Thus, an increase in training data leads to more accurate large disjuncts and a higher error concentration. The small disjuncts that are formed using the increased amount of training data may correspond to rare cases within the concept that previously were not sampled sufficiently to be learned.

Table 9.8 The effect of training set size on error concentration

	Amount of total data used for training						Δ from	
	10%		50%		90%		10 to 90%	
Data Set	ER	EC	ER	EC	ER	EC	ER	EC
kr-vs-kp	3.9	0.742	0.7	0.884	0.3	0.874	–3.6	0.132
hypothyroid	1.3	0.910	0.6	0.838	0.5	0.852	–0.8	–0.058
vote	9.0	0.626	6.7	0.762	6.9	0.848	–2.1	0.222
splice-junction	8.5	0.760	6.3	0.806	5.8	0.818	–2.7	0.058
ticket2	7.0	0.364	5.7	0.788	5.8	0.758	–1.2	0.394
ticket1	2.9	0.476	3.2	0.852	2.2	0.752	–0.7	0.276
ticket3	9.5	0.672	4.1	0.512	3.6	0.744	–5.9	0.072
soybean-large	31.9	0.484	13.8	0.660	9.1	0.742	–22.8	0.258
breast-wisc	9.2	0.366	5.4	0.650	5.0	0.662	–4.2	0.296
ocr	8.9	0.506	2.9	0.502	2.2	0.558	–6.7	0.052
hepatitis	22.2	0.318	22.5	0.526	22.1	0.508	–0.1	0.190
horse-colic	23.3	0.452	18.7	0.534	16.3	0.504	–7.0	0.052
crx	20.6	0.460	19.1	0.426	19.0	0.502	–1.6	0.042
bridges	16.8	0.100	14.6	0.270	15.8	0.452	–1.0	0.352
heart-hungarian	23.7	0.216	22.1	0.416	24.5	0.450	0.8	0.234
market1	26.9	0.322	23.9	0.422	23.6	0.440	–3.3	0.118
adult	18.6	0.486	17.2	0.452	16.3	0.424	–2.3	–0.062
weather	34.0	0.340	32.7	0.380	33.2	0.416	–0.8	0.076
network2	27.8	0.354	24.9	0.342	23.9	0.384	–3.9	0.030
promoters	36.0	0.108	22.4	0.206	24.3	0.376	–11.7	0.268
network1	28.6	0.314	25.1	0.354	24.1	0.358	–4.5	0.044
german	34.3	0.248	33.3	0.334	31.7	0.356	–2.6	0.108
coding	38.4	0.214	30.6	0.280	25.5	0.294	–12.9	0.080
move	33.7	0.158	25.9	0.268	23.5	0.284	–10.2	0.126
sonar	40.4	0.028	27.3	0.292	28.4	0.226	–12.0	0.198
bands	36.8	0.100	30.7	0.152	29.0	0.178	–7.8	0.078
liver	40.5	0.030	36.4	0.054	34.5	0.120	–6.0	0.090
blackjack	29.4	0.100	27.9	0.094	27.8	0.108	–1.6	0.008
labor	30.3	0.114	17.0	0.044	20.7	0.102	–9.6	–0.012
market2	47.3	0.032	45.7	0.028	46.3	0.040	–1.0	0.008
Average	23.4	0.347	18.9	0.438	18.4	0.471	–5.0	0.124

In this section we noted that additional training data reduces the error rate of the induced classifier and increases its error concentration. These results help to explain the pattern, described in Section 9.4, that classifiers with low error rates tend to have higher error concentrations than those with high error rates. That is, if we imagine that additional training data were made available to those data sets where the associated classifier has a high error rate, we would expect the error rate to decline and the error concentration to increase. This would tend to move classifiers into the high-EC/moderate-ER category. Thus, to a large extent, the pattern that was established in Section 9.4 between error rate and error concentration reflects the degree to which a concept has been learned – concepts that have been well-learned tend to have very large disjuncts which are extremely accurate and hence have low error concentrations.

9.7 The Effect of Noise on Small Disjuncts

Noise plays an important role in classifier learning. Both the structure and performance of a classifier will be affected by noisy data. In particular, noisy data may cause many erroneous small disjuncts to be induced. Danyluk and Provost [8] speculated that the classifiers they induced from (systematic) noisy data performed poorly because of an inability to distinguish between these erroneous consistencies and correct ones. Weiss [17] and Weiss and Hirsh [19] explored this hypothesis using, respectively, two artificial data sets and two real-world data sets and showed that noise can make rare cases (i.e., true exceptions) in the true, unknown, concept difficult to learn. The research presented in this section further investigates the role of noise in learning, and, in particular, shows how noisy data affects induced classifiers and the distribution of the errors across the disjuncts within these classifiers.

The experiments described in this section involve applying random class noise and random attribute noise to the data. The following experimental scenarios are explored:

Scenario 1: Random class noise applied to the training data
Scenario 2: Random attribute noise applied to the training data
Scenario 3: Random attribute noise applied to both training and test data

Class noise is applied only to the training set since the uncorrupted class label in the test set is required to properly measure classifier performance [12]. The second scenario, in which random attribute noise is applied only to the training set, permits us to measure the sensitivity of the learner to noise (if attribute noise were applied to the test set then even if the correct concept were learned there would be classification errors). The third scenario, in which attribute noise is applied to both the training and test sets, corresponds to the real-world situation where errors in measurement affect all examples. A level of $n\%$ random class noise means that for $n\%$ of the examples the class label is replaced by a randomly selected class value, including possibly the original value. Attribute noise is defined similarly, except that for numerical attributes a random value is selected between the minimum and maximum values that occur within the data set. Note that only when the noise level reaches 100% is all information contained within the original data lost.

The vote data set is used to illustrate the effect that noise has on the distribution of examples, by disjunct size. The results are shown in Fig. 9.10a–f, with the graphs in the left column corresponding to the case when there is no pruning and the graphs in the right column corresponding to the case when pruning is employed. Figure 9.10a, which is an exact copy of Fig. 9.1, and Fig. 9.10b, which is an exact copy of Fig. 9.4, show the results without any noise and are provided for comparison purposes. Figures 9.10c and 9.10d correspond to the case where 10% attribute noise is applied to the training data and Figs. 9.10e and 9.10f to the case where 10% class noise is applied to the train-ing data.

A comparison of Fig. 9.10a,c and e shows that both attribute and class noise cause more test examples to be covered by small disjuncts, although this shift is

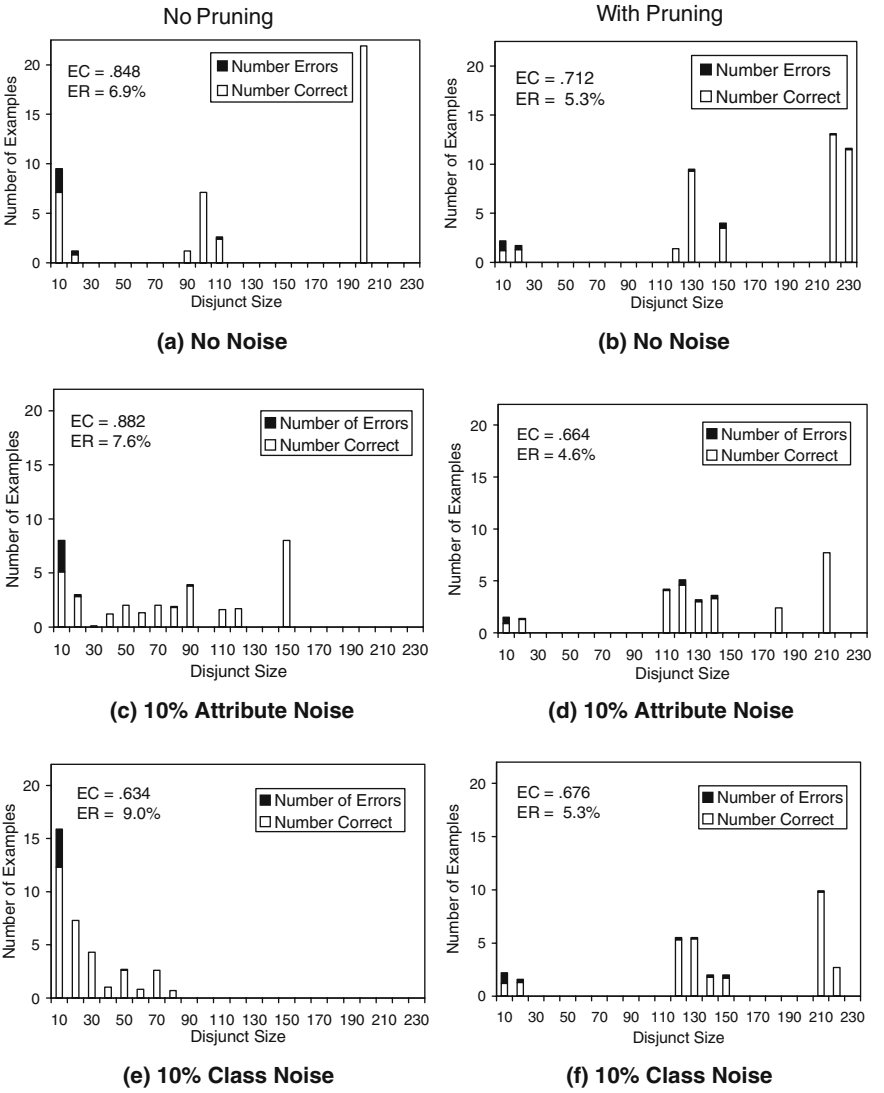

Fig. 9.10 The effect that noise has on the distribution of examples, by disjunct size

more dramatic for class noise than for attribute noise. The underlying data indicates that this shift occurs because noisy data causes more small disjuncts to be formed. This comparison also shows that the error concentration remains fairly stable when attribute noise is added but decreases significantly when class noise is added.

A careful examination of Fig. 9.10 makes it clear that pruning reduces the shift in distribution of (correctly and incorrectly) examples that is observed when pruning is not used. A comparison of the error rates for classifiers with and without pruning also shows that pruning is able to combat the effect of noise on the ability of the

classifier to learn the concept. Surprisingly, when pruning is used, classifier accuracy for the vote data set actually improves when 10% attribute noise is added – the error rate decreases from 5.3 to 4.6%. This phenomenon, which is discussed in more detail shortly, is actually observed for many of the 30 data sets, but only when low (e.g., 10%) levels of attribute noise are added. The error concentration results also indicate that even with pruning, noise causes the errors to be distributed more uniformly throughout the disjuncts than when no noise is applied.

The results presented in the remainder of this section are based on averages over 27 of the 30 data sets listed in Table 9.1 (the coding, ocr, and bands data sets were omitted due to difficulties applying our noise model to these data sets). The next three figures show, respectively, how noise affects the number of leaves, the error rate, and the error concentration of the induced classifiers. Measurements are taken at the following noise levels: 0, 5, 10, 20, 30, 40, and 50%. The curves in these figures are labeled to identify the type of noise that is applied, whether it is applied to the training set or training and test sets, and whether pruning is used. The labels are interpreted as follows: the "Class" and "Attribute" prefix indicate the type of noise, the "-Both" term, if included, indicates that the noise is applied to the training and test sets rather than to just the training set, and the "-Prune" suffix is used to indicate that the results are with pruning.

Figure 9.11 shows that without pruning the number of leaves in the induced decision tree increases dramatically with increasing levels of noise, but that pruning effectively eliminates this increase. The effect that noise has on error rate is shown in Fig. 9.12. Error rate increases with increasing levels of noise, with one exception. When attribute noise is applied to only the training data and pruning is used, the error rate decreases slightly from 17.7% with 5% noise to 17.5% with 10% noise.

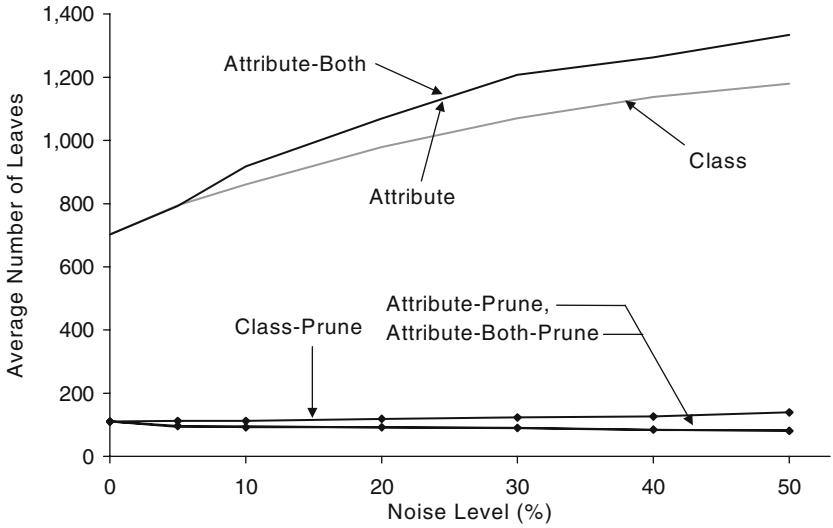

Fig. 9.11 The effect of noise on classifier complexity

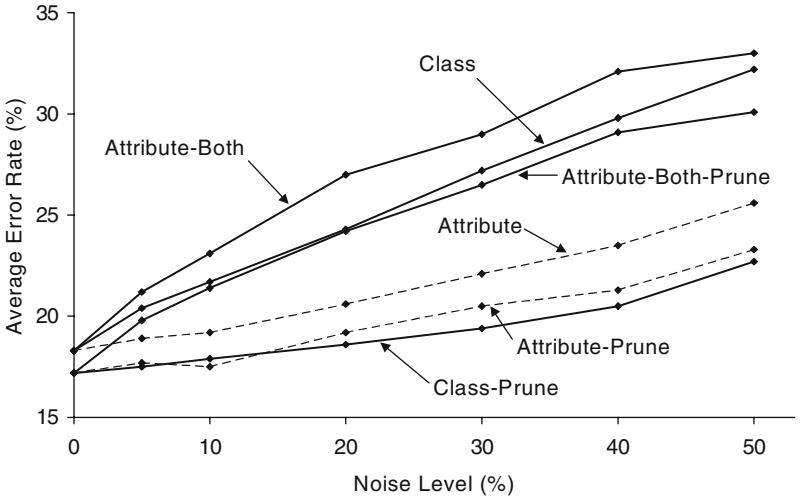

Fig. 9.12 The effect of noise on error rate

This decrease is no anomaly, since it occurs for many of the data sets analyzed. We believe the decrease in error rate may be due to the fact that attribute noise leads to more aggressive pruning (most of the data sets that show the decrease in error rate have high overall error rates, which perhaps are more likely to benefit from aggressive pruning). Figure 9.12 also shows that pruning is far more effective at handling class noise than attribute noise.

Figure 9.13 shows the effect of noise on error concentration. When pruning is not employed, increasing levels of noise lead to decreases in error concentration,

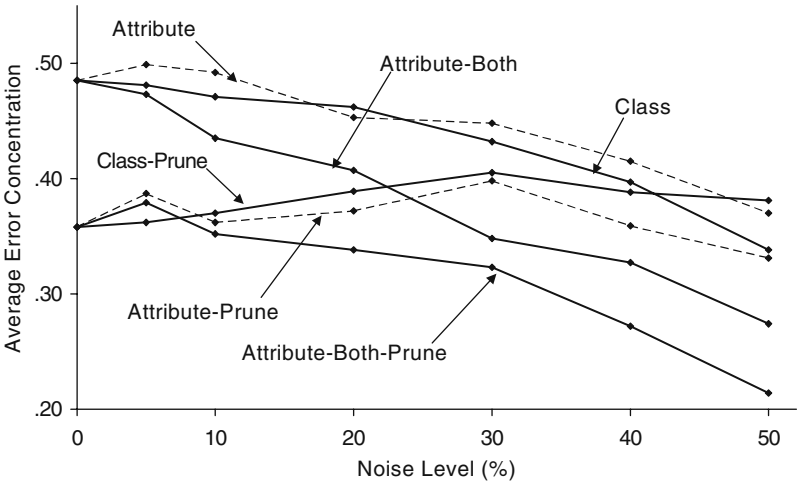

Fig. 9.13 The effect of noise on error concentration The effect of class distribution on error concentration

indicating that errors become more uniformly distributed based on disjunct size. This helps explain why we find a low-ER/high-EC group of classifiers and a high-ER/medium-EC group of classifiers: adding noise to classifiers in the former increases their error rate and decreases their error concentration, making them look more like classifiers in the latter group. The results in Fig. 9.13 also show, however, that when there is noise only in the training set, then pruning causes the error concentration to remain relatively constant (this is especially true for class noise).

The results in this section demonstrate that pruning enables the learner to combat noisy training data. Specifically, pruning removes many of the disjuncts that are caused by the noise (Fig. 9.11) and this yields a much smaller increase in error rate than if pruning were not employed (Fig. 9.12). Because pruning eliminates many of the erroneous small disjuncts, the errors are not nearly as concentrated in the small disjuncts (Fig. 9.13). We believe that the increase in error rate that comes from noisy training data when pruning is employed is at least partly due to the inability of the learner to distinguish between true exceptions and noise.

The detailed results associated with the individual data sets show that for class noise there is a trend for data sets with high error concentrations to experience a greater increase in error rate from class noise. What is much more apparent, however, is that many classifiers with low error concentrations are *extremely* tolerant of class noise, whereas none of the classifiers with high error concentrations exhibit this tolerance. For example, the blackjack and labor data sets, both of which have low error concentrations, are so tolerant of noise that when 50% random class noise is added to the training set, the error rate on the induced classifier on the test data increases by less than 1%. These results are consistent with the belief that noise makes learning difficult because it makes of an inability to distinguish between true exceptions and noise. Even without the addition of noise, none of the concepts can be induced perfectly (i.e., they have nonzero error rate). The classifiers with a high error concentration already show an inability to properly learn the rare cases in the concept (which show up as small disjuncts) – the addition of noise simply worsens the situation. Those concepts with very general cases that can be learned well without noise (leading to highly accurate large disjuncts and low error concentrations) are less susceptible to noise. For example, corrupting the class labels for a few examples belonging to a very large disjunct is unlikely to change the class label learned for that disjunct.

9.8 The Effect of Class Imbalance on Small Disjuncts

A data set exhibits class imbalance if the number of examples belonging to each class is unequal. A great deal of recent research, some of which is described in Section 9.9, has studied the problem of learning classifiers from imbalanced data, since this has long been recognized as commonly occurring and difficult data mining problem. However, with few exceptions [11, 22], this research has not examined the role of small disjuncts when learning from imbalanced data.

The study by Weiss and Provost [22] showed that examples truly belonging to the minority class are misclassified much more often than examples belonging to the majority class and that examples labeled by the classifier as belonging to the minority class (i.e., minority-class predictions) have much higher error rates than those labeled with the majority class. That study further showed that the minority-labeled disjuncts tend to cover fewer training examples than the majority-labeled disjuncts. This result is not surprising given that the minority class has, by definition, fewer training examples than the majority class.[2] The study concluded that part of the reason that minority-class predictions are more error prone than majority-class predictions is because the minority-class predictions have a lower average disjunct size and hence suffer more from the problem with small disjuncts. The work by Jo and Japkowicz [11] is discussed in Section 9.9.

In this section we extend the research by Weiss and Provost [22] to consider whether there is a causal link between class imbalance and the problem with small disjuncts in the opposite direction. That is, we consider whether class imbalance causes small disjuncts to have a higher error rate than large disjuncts, or, more generally, whether an increase in class imbalance will cause an increase in error concentration. Before evaluating this hypothesis empirically, it is useful to speculate why such a causal link might exist. Weiss and Provost suggested that one reason that minority-class predictions are more error prone than the majority-class predictions is because, by definition, there are more majority-class test examples than minority-class test examples. To see why this is so, imagine a data set for which there are nine majority-class examples for every one minority-class example. If one *randomly* generates a classifier and *randomly* labels each disjunct (e.g., leaf), then the minority-labeled disjuncts will have an expected error rate of 90% while the majority-labeled disjuncts will have an expected error rate of only 10%. Thus, this test-distribution effect favors majority-class predictions. Given that Weiss and Provost showed that small disjuncts are disproportionately likely to be labeled with the minority class, one would therefore expect this test-distribution effect to favor the larger disjuncts over the smaller disjuncts.

We evaluate this hypothesis by altering the class distribution of data sets and then measuring the error concentration associated with the induced classifiers. For simplicity, we look at only two class distributions for each data set: the naturally occurring class distribution and a perfectly balanced class distribution, in which each class is represented in equal proportions. By comparing the error concentrations for these two class distributions, we can also determine how much of the "problem with small disjuncts" is due to class imbalance in the data set.

We form data sets with the natural and balanced class distributions using the methodology described by Weiss and Provost [22]. This methodology employs stratified sampling, without replacement, to form the desired class distribution from the

[2] The detailed results show that the induced classifiers have more majority-labeled disjuncts than minority-labeled disjuncts, but the ratio of majority-labeled disjuncts to minority-labeled disjuncts is smaller than the ratio of majority-class examples to minority-class examples. Thus the majority-class disjuncts cover more examples than the minority-class examples.

original data set. The number of examples selected for training is the same for the natural and balanced versions of each data set, to ensure that any differences in performance are due solely to the difference in class distribution (the actual number of training examples that are used is reduced from what is available, to ensure that the balanced class distribution can be formed without duplicating any examples). Because this methodology reduces the number of training examples, we exclude the small data sets when studying class imbalance, so that all classifiers are induced from using a "reasonable" number of examples. The data sets employed in this section include the larger data sets from Table 9.1 plus some additional data sets. These data sets, listed in Table 9.9, are identical to the ones studied by Weiss and Provost [22]. They include 20 data sets from the UCI repository, 5 data sets, identified with a "+," from previously published work by researchers at AT&T [7] and one new data set, the phone data set, generated by the author. The data sets are listed in order of decreasing class imbalance (the percentage of minority-class examples in each data set is included). In order to simplify the presentation and analysis of the results, data sets with more than two classes were mapped into two classes by designating the least frequently occurring class as the minority class and mapping the remaining classes into a new, majority class. Each data set that originally started with more than two classes is identified with an asterisk (*).

Table 9.9 Description of data sets for class imbalance experiments

No.	Data set	Min. (%)	Size	No.	Data set	Min. (%)	Size
1	letter-a*	3.9	20,000	14	network2	27.9	3826
2	pendigits*	8.3	13,821	15	yeast*	28.9	1484
3	abalone*	8.7	4177	16	network1+	29.2	3577
4	sick-euthyroid	9.3	3163	17	car*	30.0	1728
5	connect-4*	9.5	11,258	18	german	30.0	1.000
6	optdigits*	9.9	5620	19	breast-wisc	34.5	699
7	covertype*	14.8	581,102	20	blackjack+	35.6	15,000
8	solar-flare*	15.7	1389	21	weather+	40.1	5597
9	phone	18.2	652,557	22	bands	42.2	538
10	letter-vowel*	19.4	20,000	23	market1+	43.0	3181
11	contraceptive*	22.6	1473	24	crx	44.5	690
12	adult	23.9	48,842	25	kr-vs-kp	47.8	3196
13	splice-junction*	24.1	3175	26	move+	49.4	3029

Figure 9.14 shows the error concentration for the classifiers induced by C4.5 from the natural and balanced versions of the data sets listed in Table 9.9. Since the error concentrations are all greater than zero when there is no class imbalance, we conclude that even with a balanced data set errors tend to be concentrated toward the smaller disjuncts. However, by comparing the error concentrations associated with the classifiers induced from the balanced and natural class distributions, we see that when there is class imbalance, with few exceptions, the error concentration increases. The differences tend to be larger when the data set has greater class imbalance (the leftmost data set has the most natural class imbalance and the class imbalance decreases from left to right).

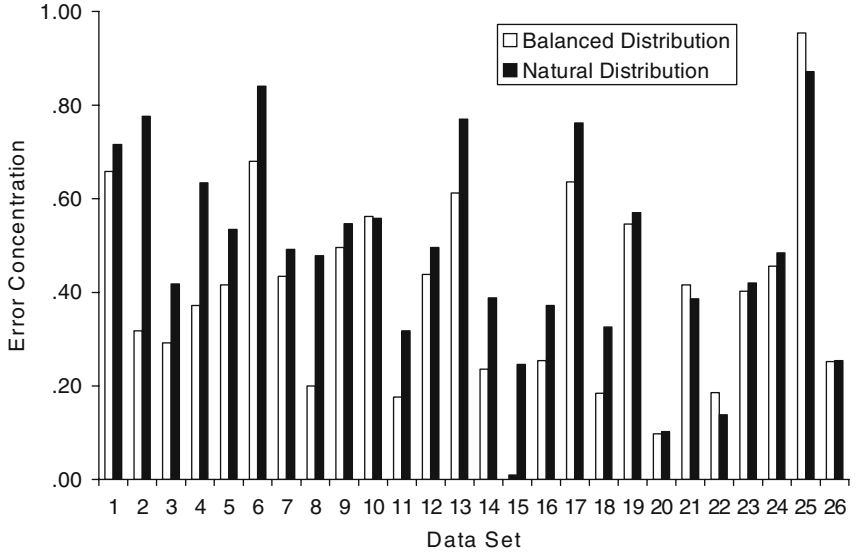

Fig. 9.14 The effect of class distribution on error concentration

If we look at the average error concentration for the classifiers induced from the natural and balanced versions of the 26 data sets, we see that the balanced versions have an average error concentration of 0.396 while the natural versions have an average error concentration of 0.496. This corresponds to a 20% reduction in error concentration when class imbalance is removed. If we restrict our attention to the first 18 data sets, which contain at most 30% minority-class examples, then the differences in error concentration are 28% (0.387 for the balanced data sets versus 0.537 for the data sets with the natural class distributions). We therefore conclude that for data sets with class imbalance, part of the reason why small disjuncts have a higher error rate than the large disjuncts is due to the fact that minority-class predictions are more likely to be erroneous due to the test-distribution effect described earlier. This is empirical evidence that class imbalance is partly responsible for the problem with small disjuncts. This also indicates that if one artificially modifies the class distribution of the training data to be more balanced, then the error concentration will decrease. This observation may help explain why, as noted by Weiss and Provost [22], classifiers built using balanced class distributions tend to be quite robust.

9.9 Related Work

Research on small disjuncts can be placed into the following three categories, which we use to organize our discussion of related work. These three categories are based on whether the purpose of the research is to:

1. characterize and/or measure the role of small disjuncts in learning,
2. provide a better understanding of small disjuncts (e.g., why they are more error prone than large disjuncts), or
3. design better classifiers that address the problem with small disjuncts.

Most previous research on small disjuncts only incidentally tried to characterize or measure the role of small disjuncts in learning and only analyzed one or two data sets [1, 3, 8, 9, 17, 19]. This made it impossible to form any general conclusions. We addressed this problem by analyzing 30 data sets.

Some research has focused on providing a better understanding of small disjuncts. Danyluk and Provost [8] observed that in the domain they were studying, when they trained using noisy data, classifier accuracy suffered severely. They speculated that this occurred because (1) it is difficult to distinguish between noise and true exceptions and (2) in their domain, errors in measurement and classification often occur systematically rather than randomly. Thus, they speculated that it was difficult to distinguish between erroneous consistencies and correct ones. This speculation formed the basis for the work by Weiss [17] and Weiss and Hirsh [19]. Weiss [17] investigates the interaction between noise, rare cases, and small disjuncts using synthetic data sets, for which the true "concept" is known and can be manipulated. Some synthetic data sets were constructed from concepts that included many rare, or exceptional cases, while others were constructed from concepts that mainly included general cases. The research showed that the rare cases tended to form small disjuncts in the induced classifier. It further showed that systematic attribute noise, class noise, and missing attributes can each cause the small disjuncts to have higher error rates than the large disjuncts, and also cause those test examples that correspond to rare cases to be misclassified more often than those test examples corresponding to common cases. That paper also provided an explanation for this behavior: it is asserted that attribute noise in the training data can cause the common cases to look like the rare cases, thus "overwhelming" the rare cases and causing the wrong subconcept to be learned.

The majority of research on small disjuncts focuses on ways to address the problem with small disjuncts. Holte et al. [9] evaluate several strategies for improving learning in the presence of small disjuncts. They show that the strategy of eliminating all small disjuncts is ineffective, because the emancipated examples are then even more likely to be misclassified. The authors focus on a strategy of making small disjuncts highly specific and argue that while a maximum generality bias, which is used by systems such as ID3, is appropriate for large disjuncts, it is not appropriate for small disjuncts. To test this claim, they ran experiments where a maximum generality bias is used for the large disjuncts and a maximum specificity bias is used for the small disjuncts (for a maximum specificity bias *all* conditions satisfied by the training examples covered by a disjunct are added to the disjunct). The experimental results show that with the maximum specificity bias, the resulting disjuncts cover fewer cases but have much lower error rates. Unfortunately, the emancipated examples increase the error rate of the large disjuncts to the extent that the overall error rates remain roughly the same. Although the authors also experiment with a more

selective bias that produces interesting results, it does not demonstrably improve learning.

Ting [15] evaluates a method for improving the performance of small disjuncts that also uses a maximum specificity bias. However, unlike the method employed by Holte et al. [9], this method does not affect (and therefore cannot degrade) the performance of the large disjuncts. The basic approach is to use C4.5 to determine if an example is covered by a small or large disjunct. If it is covered by a large disjunct, then C4.5 is used to classify the example. However, if the example is covered by a small disjunct, then IB1, an instance-based learner, is used to classify the example. Instance-based learning is used in this case because it can be considered an extreme example of the maximum specificity bias. In order to use this hybrid learning method, there must be a specific criterion for determining what is a small disjunct. The paper empirically evaluates alternative criteria, based on a threshold value and (1) the absolute size of the disjunct, (2) the relative size of the disjunct, or (3) the error rate of the disjunct. For each criterion, only the best result, produced using the best threshold, is displayed. The results are therefore overly optimistic because the criteria/threshold values are selected using the test data rather than an independent holdout set. Thus, although the observed results are encouraging, it cannot be claimed that the composite learner is very successful in addressing the problem with small disjuncts.

Carvalho and Freitas [3] employ a hybrid method similar to that used by Ting [15]. They also use C4.5 to build a decision tree and then, for each training example, use the size of the leaf covering that example to determine if the example is covered by a small or large disjunct. The training examples that fall into each small disjunct are then fed together into a genetic algorithm-based learner that forms rules to specifically cover the examples that fall into that individual disjunct. Test examples that fall into leaves corresponding to large disjuncts are then assigned a class label based on the decision tree; test examples that fall into a small disjunct are classified by the rules learned by the genetic algorithm for that particular disjunct. Their results are also encouraging, but, because they are based on only a few data sets, and because, as with the results by Ting [15], the improvements in error rate are only seen for certain specific definitions of "small disjunct," it cannot be concluded that this research substantially addresses the problem with small disjuncts.

Several other approaches are advocated for addressing the problem with small disjuncts. Quinlan [13] tries to minimize the problem by improving the probability estimates used to assign a class label to a disjunct. A naive estimate of the error rate of a disjunct is the proportion of the training examples that it misclassifies. However, this estimate performs quite poorly for small disjuncts, due to the small number of examples used to form the estimate. Quinlan describes a method for improving the accuracy estimates of the small disjuncts by taking the class distribution into account. The motivation for this work is that for unbalanced class distributions one would expect the disjuncts that predict the majority class to have a lower error rate than those predicting the minority class (this is the test-distribution effect described in Section 9.8). Quinlan incorporates these *prior probabilities* into the error rate estimates. However, instead of using the overall class distribution as the

prior probability, Quinlan generates a more representative measure by calculating the class distribution only on those training examples that are ”close” to the small disjunct – that is, fail to satisfy at most one condition in the disjunct. The experimental results demonstrate that Quinlan's error rate estimation model outperforms the naive method, most significantly for skewed distributions.

Van den Bosch et al. [16] advocate the use of instance-based learning for domains with many small disjuncts. They are mainly interested in language learning tasks, which they claim result in many small disjuncts, or “pockets of exceptions.” In particular, they focus on the problem of learning word pronunciations. Because instance-based learning does not form disjunctive concepts, rather than determining disjunct sizes, they instead compute cluster sizes, which they view as analogous to disjunct size. They determine cluster sizes by repeatedly selecting examples from the data, forming a ranked list of the 100 nearest neighbors, and then they determine the rank of the nearest neighbor with a different class value – this value minus 1 is considered to be the cluster size. This method, as well as the more conventional method of measuring disjunct size via a decision tree, shows that the word pronunciation domain has many small disjuncts. The authors also try an information-theoretic weighted similarity matching function, which effectively rescales the feature space so that ”more important” features have greater weight. When this is done, the size of the average cluster is increased from 15 to 25. Unfortunately, error rates were not specified for the various clusters and hence one cannot measure how effective this strategy is for addressing the problem with small disjuncts.

The problem of learning from imbalanced data where the classes are represented in unequal proportions is a common problem that has received a great deal of attention [4, 5, 10, 21]. Our results in Section 9.8 provide a link between the problem of learning from imbalanced data and the small disjuncts problem. A similar link was provided by Jo and Japkowicz [11], who also showed that a method that deals with the problem of small disjuncts, cluster-based oversampling, can also improve the performance of classifiers that learn from imbalanced data. This supports the notion that a better understanding of small disjuncts can lead the design of better classification methods.

9.10 Conclusion

This chapter makes several contributions to the study of small disjuncts and, more generally, classifier learning. First, the degree to which small disjuncts affect learning is quantified using a new measure, error concentration. Because error concentration is measured for a large collection of data sets, for the first time it is possible to draw general conclusions about the impact that small disjuncts have on learning. The experimental results show that, as expected, for many classifiers errors are highly concentrated toward the smaller disjuncts – however the results also show that for a substantial number of classifiers this simply is not true. Our research also indicates

that the error concentration for the classifiers induced using C4.5 and Ripper is highly correlated, indicating that error concentration measures some "real" aspect of the concept being learned and is not totally an artifact of the learner. Finally, our results indicate that classifiers with relatively low error rates almost always have high error concentrations while this is not true of classifiers with high error rates. Analysis indicates that this is due to the fact that classifiers with low error rates generally contain some very accurate large disjuncts. We conclude from this that concepts that can be learned well tend to contain very general cases and that C4.5 and Ripper generate classifiers with similar error concentrations because they are both able to form accurate large disjuncts to cover these general cases.

Another contribution of this chapter is that it takes an in-depth look at pruning. This is particularly important because previous research into small disjuncts largely ignores pruning. Our results indicate that pruning eliminates many of the small disjuncts in the induced classifier and that this leads to a reduction in error concentration. These results also show that pruning is more effective at reducing the error rate of a classifier when the unpruned classifier has a high error concentration. Pruning is evaluated as a method for addressing the problem with small disjuncts and is shown to be of limited effectiveness. Our analysis also shows that because pruning distributes the errors that were concentrated in small disjuncts to the more accurate, larger disjuncts, pruning can actually degrade classifier performance when one may be selective in applying the induced classification rules.

In this chapter we also show how factors such as training set size, noise, and class imbalance affect small disjuncts and error concentration. This provides a better understanding not only of small disjuncts, but also of how these important, real-world factors affect inductive learning. As an example, the results in Section 9.6 permit us to explain how increasing the amount of training data leads to an improvement in classifier accuracy. These results, which show that increasing the amount of training data leads to an increase in error concentration, suggest that the additional training data allows the general cases within the concept to be learned better than before, but that it also introduces many new small disjuncts. These small disjuncts, which correspond to rare cases in the concept, are formed because there is now sufficient training data to ensure that they are sampled. These small disjuncts are error prone, however, due to the small number of training examples used to determine the classification. The small disjuncts in the induced classifier may also be error prone because, as the results in Section 9.7 and previous research [17, 19] indicate, noisy data causes erroneous small disjuncts to be formed. Our results indicate that pruning is somewhat effective at combating the effect of noise on classifier accuracy because of its ability to handle small disjuncts. Finally, the results in this chapter also indicate that class imbalance can worsen the problem with noise and small disjuncts. This may help explain why a balanced class distribution often leads to classifiers that are more robust than those induced from the naturally occurring class distribution.

We believe that an understanding of small disjuncts is important in order to properly appreciate the difficulties associated with classifier learning, because, as this chapter clearly shows, it is often the small disjuncts that determine the overall performance of a classifier. We therefore hope that the metrics provided in this chapter

can be used to better evaluate the performance of classifiers and will ultimately lead to the design of better classifiers. The research in this chapter also enables us to better understand how various real-world factors, like noise and class imbalance, impact classifier learning. This is especially important as data mining tackles more difficult problems.

References

1. Ali, K.M., Pazzani, M.J.: Reducing the small disjuncts problem by learning probabilistic concept Descriptions. In: Petsche, T. (ed.) Computational Learning Theory and Natural Learning Systems, Volume 3, MIT Press, Cambridge, MA (1992)
2. Asuncion, A., Newman, D.J.: UCI Machine Learning Repository. University of California, Irvine, School of Information and Computer Science. http://www.ics.uci.edu/~mlearn/MLRepository.html. Cited Sept 2008
3. Carvalho D.R., Freitas A.A.: A hybrid decision tree/genetic algorithm for coping with the problem of small disjuncts in data mining. In: Proceedings of the 2000 Genetic and Evolutionary Computation Conference, pp. 1061–1068 (2000)
4. Chawla N.V., Bowyer K.W., Hall L.O., Kegelmeyer W.P.: SMOTE: Synthetic minority oversampling technique. Journal of Artificial Intelligence Research, 16, 321–357 (2002)
5. Chawla N.V., Cieslak D.A., Hall L.O., Joshi A.: Automatically countering imbalance and its empirical relationship to cost. Data Mining and Knowledge Discovery, 17(2), 225–252 (2008)
6. Cohen W.: Fast effective rule induction. In: Proceedings of the Twelfth International Conference on Machine Learning, pp. 115–123 (1995)
7. Cohen W., Singer Y.: A simple, fast, and effective rule learner. In: Proceedings of the Sixteenth National Conference on Artificial Intelligence, pp. 335–342 (1999)
8. Danyluk A.P., Provost F.J.: Small disjuncts in action: learning to diagnose errors in the local loop of the telephone network. In: Proceedings of the Tenth International Conference on Machine Learning, pp. 81–88 (1993)
9. Holte R.C., Acker L.E., Porter B.W.: Concept learning and the problem of small disjuncts. In: Proceedings of the Eleventh International Joint Conference on Artificial Intelligence, pp. 813–818 (1989)
10. Japkowicz N., Stephen S.: The class imbalance problem: a systematic study. Intelligent Data Analysis 6(5), 429–450 (2002)
11. Jo T., Japkowicz, N. Class imbalances versus small disjuncts. SIGKDD Explorations 6(1), 40–49 (2004)
12. Quinlan J.R.: The effect of noise on concept learning. In: Michalski R.S., Carbonell J.G., Mitchell T.M. (eds.), Machine Learning, an Artificial Intelligence Approach, Volume II, Morgan Kaufmann, San Francisco, CA (1986)
13. Quinlan J.R.: Technical note: improved estimates for the accuracy of small disjuncts. Machine Learning, 6(1) (1991)
14. Quinlan J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann, San Mateo, CA (1993)
15. Ting K.M.: The problem of small disjuncts: its remedy in decision trees. In: Proceedings of the Tenth Canadian Conference on Artificial Intelligence, pp. 91–97 (1994)
16. Van den Bosch A., Weijters A., Van den Herik H.J., Daelemans W.: When small disjuncts abound, try lazy learning: A case study. In: Proceedings of the Seventh Belgian-Dutch Conference on Machine Learning, pp. 109–118 (1997)
17. Weiss G.M.: Learning with rare cases and small disjuncts. In: Proceedings of the Twelfth International Conference on Machine Learning, pp. 558–565 (1995)

18. Weiss G.M.: Mining with rarity: A unifying framework, SIGKDD Explorations 6(1), 7–19 (2004)
19. Weiss G.M., Hirsh H.: The problem with noise and small disjuncts. In: Proceedings of the Fifteenth International Conference on Machine Learning, pp. 574–578 (1998)
20. Weiss G.M., Hirsh H.: A quantitative study of small disjuncts. In: Proceedings of the Seventeenth National Conference on Artificial Intelligence, Austin, Texas, pp. 665–670 (2000)
21. Weiss G.M., McCarthy K., Zabar B.: Cost-Sensitive Learning vs. Sampling: Which is best for handling unbalanced classes with unequal error costs? In: Proceedings of the 2007 International Conference on Data Mining, pp. 35–41 (2007)
22. Weiss G.M., Provost F.: Learning when training data are costly: the effect of class distribution on tree induction. Journal of AI Research 19, 315–354 (2003)