A Quantitative Study of Small Disjuncts

Gary M. Weiss* **and Haym Hirsh**

Department of Computer Science
Rutgers University
New Brunswick, New Jersey 08903
{gweiss, hirsh}@cs.rutgers.edu

Abstract

Systems that learn from examples often express the learned concept in the form of a disjunctive description. Disjuncts that correctly classify few training examples are known as small disjuncts and are interesting to machine learning researchers because they have a much higher error rate than large disjuncts. Previous research has investigated this phenomenon by performing *ad hoc* analyses of a small number of datasets. In this paper we present a quantitative measure for evaluating the effect of small disjuncts on learning and use it to analyze 30 benchmark datasets. We investigate the relationship between small disjuncts and pruning, training set size and noise, and come up with several interesting results.

Introduction

Systems that learn from examples often express the learned concept as a disjunction. The size of a disjunct is defined as the number of training examples that it correctly classifies (Holte, Acker, and Porter 1989). A number of empirical studies have demonstrated that learned concepts include disjuncts that span a large range of disjunct sizes and that the small disjuncts—those disjuncts that correctly classify only a few training examples—collectively cover a significant percentage of the test examples (Holte, Acker, and Porter 1989; Ali and Pazzani 1992; Danyluk and Provost 1993; Ting 1994; Van den Bosch et al. 1997; Weiss and Hirsh 1998). It has also been shown that small disjuncts often correspond to rare cases within the domain under study (Weiss 1995) and cannot be totally eliminated if high predictive accuracy is to be achieved (Holte et al. 1989). Previous studies have shown that small disjuncts have much higher error rates than large disjuncts and contribute a disproportionate number of the total errors. This phenomenon is known as "the problem with small disjuncts".

There are two reasons for studying the problem with small disjuncts. The first is that small disjuncts can help us answer important machine learning questions, such as: how does the amount of available training data affect learning, how does pruning work and when is it most effective, and how does noise affect the ability to learn a concept? Thus, we use small disjuncts as a lens through which to examine important issues in machine learning. The second reason for studying small disjuncts is to learn to build machine learning programs that "address" the problem with small disjuncts. These learners will improve the accuracy of the small disjuncts without significantly decreasing the accuracy of the large disjuncts, so that the overall accuracy of the learned concept is improved. Several researchers have attempted to build such learners. One approach involves employing a maximum specificity bias for learning small disjuncts, while continuing to use the more common maximum generality bias for the large disjuncts (Holte et al. 1989; Ting 1994). Unfortunately, these efforts have produced, at best, only marginal improvements. A better understanding of small disjuncts and their role in learning may be required before further advances are possible.

In this paper we use small disjuncts to gain a better understanding of machine learning. In the process of doing this, we address a major limitation with previous research—that very few datasets were analyzed: Holte et al. (1989) analyzed two datasets, Ali and Pazzani (1992) one dataset, Danyluk and Provost (1993) one dataset, and Weiss and Hirsh (1998) two datasets. Because so few datasets were analyzed, only relatively weak qualitative conclusions were possible. By analyzing thirty datasets, we are able to draw some quantitative conclusions, as well as form more definitive qualitative conclusions than previously possible.[†]

Description of Experiments

The results presented in this paper are based on 30 datasets, of which 19 were collected from the UCI repository (Blake and Merz 1998) and 11 from researchers at AT&T (Cohen 1995; Cohen and Singer 1999). Numerous experiments were run on these datasets to assess the impact of small disjuncts on learning, especially as factors such as training set size, pruning strategy, and noise level are varied. The majority of experiments use C4.5, a program for inducing decision trees (Quinlan 1993). C4.5 was modified by the authors to collect information related to disjunct size. During the training phase the modified software assigns each disjunct/leaf a value based on the number of training examples it correctly classifies. The number of correctly and incorrectly classified examples associated with each disjunct is then tracked during the testing phase, so that at

* Also AT&T, 20 Knightsbridge Road, Piscataway, New Jersey

† See http://www.cs.rutgers.edu/~gweiss/small_disjuncts.html for a survey of work on small disjuncts.

the end the distribution of correctly/incorrectly classified test examples by disjunct size is known. For example, the software might record the fact that disjuncts of size 3 collectively classify 5 test examples correctly and 3 incorrectly. Some experiments were repeated with RIPPER, a program for inducing rule sets (Cohen 1995), in order to assess the generality of our results.

Since pruning eliminates many small disjuncts, consistent with what has been done previously, pruning is disabled for C4.5 and RIPPER for most experiments (as is seen later, however, the same trends are seen even when pruning is not disabled). C4.5 is also run with the –m1 option, to ensure that nodes continue to be split until they only contain examples of a single class, and RIPPER is configured to produce unordered rules so that it does not produce a single default rule to cover the majority class. All experiments employ 10-fold cross validation and the results are therefore based on averages of the test set calculated over 10 runs. Unless specified otherwise, all results are based on C4.5 without pruning.

An Example: The Vote Dataset

In order to illustrate the problem with small disjuncts and introduce a way of measuring this problem, we examine the concept learned by C4.5 from the Vote dataset. Figure 1 shows how the correctly and incorrectly classified test examples are distributed across the disjuncts in this concept. Each bin in the figure spans 10 sizes of disjuncts. The leftmost bin shows that those disjuncts that classify 0-9 training examples correctly cover 9.5 test examples, of which 7.1 are classified correctly and 2.4 classified incorrectly. The fractional values occur because the results are averaged over 10 cross-validated runs. Disjuncts of size 0 occur because when C4.5 splits a node using a feature *f*, the split uses all possible feature values, whether or not the value occurs within the training examples at that node.

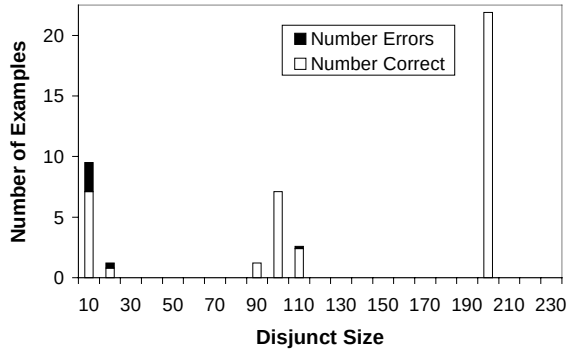

Figure 1: Distribution of Errors

Figure 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. Space limitations prevent us from showing the distribution of disjuncts, but of the 50 disjuncts in the learned concept, 45 of them are associated with the leftmost bin.

The data may also be described using a new measure, *mean disjunct size*. This measure is computed over a set of examples as follows: each example is assigned a value equal to the size of the disjunct that classifies it, and then the mean of these values is calculated. For the concept shown in Figure 1, the mean disjunct size over all test examples is 124—one can also view this as the center of mass of the bins in the figure. The mean disjunct size for the incorrectly (correctly) classified test examples is 10 (133). Since 10 << 133, the errors are heavily concentrated toward the smaller disjuncts.

In order to better show the extent to which errors are concentrated toward the small disjuncts, we plot, for each disjunct size *n*, the percentage of test errors versus percentage of correctly classified test examples covered by disjuncts of size *n* or less. Figure 2 shows this plot for the concept induced from the Vote dataset. It shows, for example, that disjuncts with size 0-4 contribute 5.1% of the correctly classified test examples but 73% of the total test errors. Since the curve in Figure 2 is above the line Y=X, the errors are concentrated toward the smaller disjuncts.

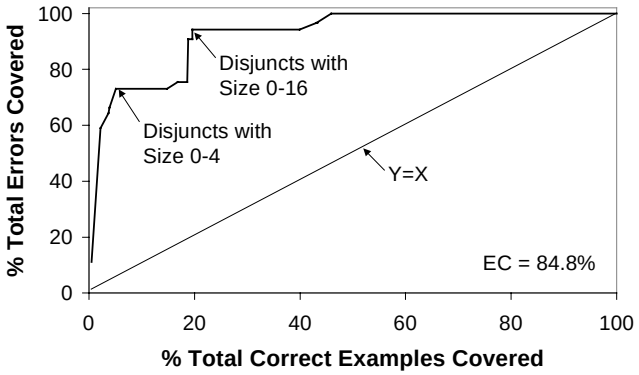

Figure 2: An Error Concentration Curve

To make it easy to compare the degree to which errors are concentrated in the small disjuncts for different concepts, we introduce a measurement called error concentration. *Error Concentration* (EC) is defined as the percentage of the total area above the line Y=X in Figure 2 that falls under the EC curve. EC may take on values between 100 and –100, but is expected to be positive—a negative value indicates that the errors are concentrated more toward the larger disjuncts. The EC value for the concept in Figure 2 is 84.8%, indicating that the errors are highly concentrated toward the small disjuncts.

Results

In this section we present the EC values for 30 datasets and demonstrate that, although they exhibit the problem with small disjuncts to varying degrees, there is some structure to this problem. We then present results that demonstrate how small disjuncts are affected by pruning, training set size, and noise. Due to space limitations, only a few key results are presented in this section. More detailed results are presented elsewhere (Weiss and Hirsh 2000).

Error Concentration for 30 Datasets

C4.5 was applied to 30 datasets and the results, ordered by EC, are summarized in Table 1. We also list the percentage of test errors contributed by the smallest disjuncts that cover 10% of the correctly classified test examples. Note the wide range of EC values and the number of concepts with high EC values.

EC Rank	Dataset	Dataset Size	Error Rate (%)	Largest Disjunct	Number Leaves	% Errors at 10% Correct	Error Conc.
1	kr-vs-kp	3196	0.3	669	47	75.0	87.4
2	hypothyroid	3771	0.5	2697	38	85.2	85.2
3	vote	435	6.9	197	48	73.0	84.8
4	splice-junction	3175	5.8	287	265	76.5	81.8
5	ticket2	556	5.8	319	28	76.1	75.8
6	ticket1	556	2.2	366	18	54.8	75.2
7	ticket3	556	3.6	339	25	60.5	74.4
8	soybean-large	682	9.1	56	175	53.8	74.2
9	breast-wisc	699	5.0	332	31	47.3	66.2
10	ocr	2688	2.2	1186	71	52.1	55.8
11	hepatitis	155	22.1	49	23	30.1	50.8
12	horse-colic	300	16.3	75	40	31.5	50.4
13	crx	690	19.0	58	227	32.4	50.2
14	bridges	101	15.8	33	32	15.0	45.2
15	heart-hungarian	293	24.5	69	38	31.7	45.0
16	market1	3180	23.6	181	718	29.7	44.0
17	adult	21280	16.3	1441	8434	28.7	42.4
18	weather	5597	33.2	151	816	25.6	41.6
19	network2	3826	23.9	618	382	31.2	38.4
20	promoters	106	24.3	20	31	32.8	37.6
21	network1	3577	24.1	528	362	26.1	35.8
22	german	1000	31.7	56	475	17.8	35.6
23	coding	20000	25.5	195	8385	22.5	29.4
24	move	3028	23.5	35	2687	17.0	28.4
25	sonar	208	28.4	50	18	15.9	22.6
26	bands	538	29.0	50	586	65.2	17.8
27	liver	345	34.5	44	35	13.7	12.0
28	blackjack	15000	27.8	1989	45	18.6	10.8
29	labor	57	20.7	19	16	33.7	10.2
30	market2	11000	46.3	264	3335	10.3	4.0

Table 1: Error Concentration for 30 Datasets

While dataset size is not correlated with error concentration, error rate clearly is—concepts with low error rates (<10%) tend to have high EC values. Based on the error rate (ER) and EC values, the entries in Table 1 seem to fit naturally into the following three categories.

1. Low-ER/High-EC: includes datasets 1-10
2. High-ER/Medium-EC: includes datasets 11-22
3. High-ER/Low-EC: includes datasets 23-30

Note that there are no learned concepts with very high EC and high ER, or with low EC and low ER. Of particular interest is that fact that for those datasets in the Low-ER/High-EC group, the largest disjunct in the concept classifies a significant portion of the total training examples, whereas this is not true for the datasets in the High-ER/Low-EC group. Due to space considerations, the results for C4.5 with pruning are not included in Table 1, but the average EC value over the 30 datasets with pruning is 33.5. While this is less than the average without pruning (47.1), it still is well above 0, indicating that even after pruning a substantial proportion of the errors are still concentrated in the smaller disjuncts.

Comparison with Results from RIPPER

Some learning methods, such as neural networks, do not have a notion of a disjunct, while others, such as nearest neighbor methods, do not form disjunctive concepts, but generate something very close, since clusters of examples can be viewed as disjuncts (Van den Bosch et al. 1997). C4.5 is used for most experiments in this paper because it is well known and forms disjunctive concepts. In order to support the generality of any conclusions we draw from the results using C4.5, we compare the EC values for C4.5 with those of RIPPER, a rule learner that also generates disjunctive concepts. The comparison is presented in Figure 3, where each point represents the EC values for a single dataset. Since the results are clustered around the line Y=X, both learners tend to produce concepts with similar EC values, 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 EC values are large—the same 10 datasets generate the largest 10 EC values for both learners.

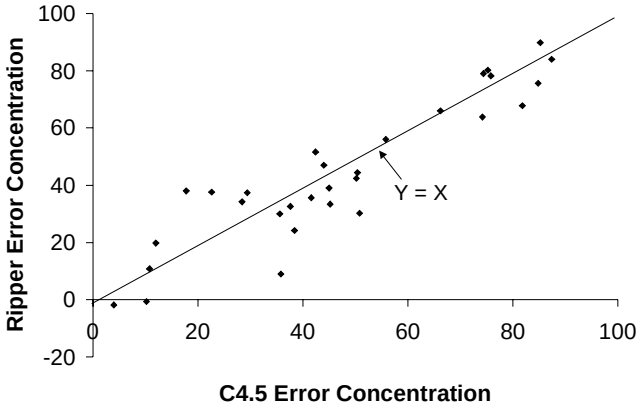

Figure 3: Comparison of C4.5 and RIPPER EC Values

The agreement shown in Figure 3 supports our belief that there is a fundamental property of the underlying datasets that is responsible for the EC values. We believe this property is the relative frequency of rare and general cases in the "true", but unknown, concept to be learned. We recognize, however, that a concept that has many rare cases when expressed as a disjunctive concept may not have them when expressed in a different form. We believe this does not significantly decrease the generality of our results given the number of learners that form disjunction-like concepts.

The Effect of Pruning

Pruning is not used for most of our experiments because it partially obscures the effects of small disjuncts. Nonetheless, small disjuncts provide an opportunity for better understanding how pruning works. Figure 4 displays the same information as Figure 1, except that the results were generated using C4.5 with pruning. Pruning causes the overall error rate to decrease to 5.3% from 6.9%.

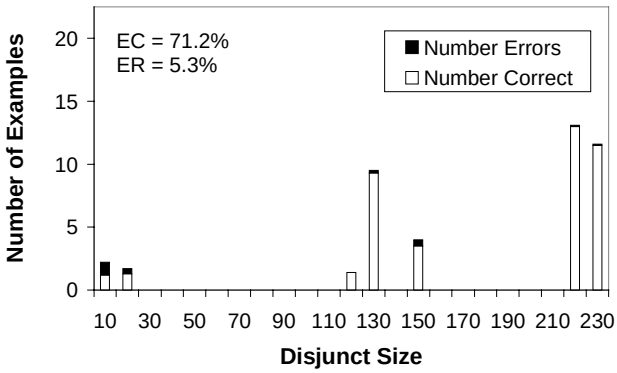

Figure 4: Distribution of Errors with Pruning

Comparing Figure 4 with Figure 1 shows that with pruning the errors are less concentrated toward the small disjuncts (the decrease in EC from 84.8 to 71.2 confirms this). It is also apparent that with pruning far fewer examples are classified by disjuncts with size less than 30. This is because the distribution of disjuncts has changed—whereas before there were 45 disjuncts of size less than 10, after pruning there are only 7. So, pruning eliminates most of the small disjuncts and many of the "emancipated" examples are then classified by the larger disjuncts. Overall, pruning causes the EC to decrease for 23 of the 30 datasets—and the decrease is often large. Looking at this another way, pruning causes the mean disjunct size associated with both the correct and incorrectly classified examples to increase, but the latter increases more than the former. Even after pruning the problem with small disjuncts is still quite evident—after pruning the average EC for the first 10 datasets is 50.6.

Figure 5 plots the absolute improvement in error rate due to pruning against EC rank. The first 10 datasets, which are in the low-ER/high-EC group, show a moderate improvement in error rate. The datasets in the high-ER/medium-EC group, which starts with the Hepatitis dataset, show more improvement, but have more room for improvement due to their higher error rate. The datasets in the high-ER/low-EC group, which start with the Coding dataset, show a net *increase* in error rate. These results suggest that pruning helps when the problem with small disjuncts is quite severe, but may actually increase the error rate in other cases.

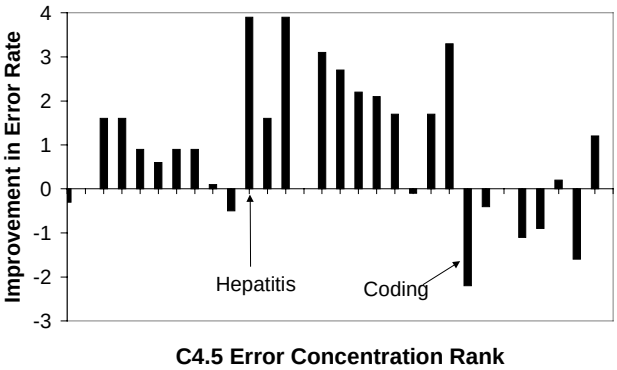

Figure 5: Absolute Improvement in Error Rate vs. EC Rank

The Effect of Training Set Size

Small disjuncts provide an opportunity to better understand how training set size affects learning. We again apply C4.5 to the Vote dataset, except that this time a different 10% (not 90%) of the dataset is used for training for each of the 10 cross-validation runs. Thus, the training set size is 1/9 the size it was previously. As before, each run employs a different 10% of the data for testing. The resulting distribution of examples is shown in Figure 6.

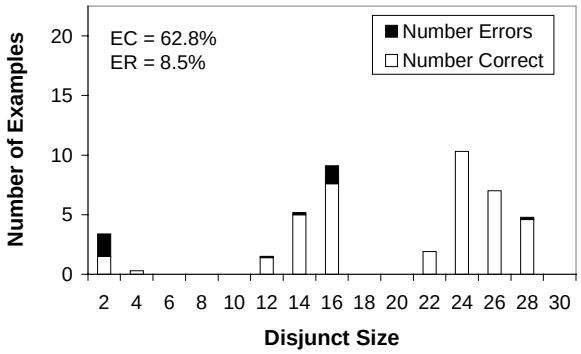

Figure 6: Distribution of Errors (10% Training Data)

Comparing the distribution of errors between Figures 1 and 6 shows that errors are less concentrated toward the smaller disjuncts in Figure 6. This is consistent with the fact that the EC decreases from 84.8 to 62.8 and the mean disjunct size over all examples decreases from 124 to 19, while the mean disjunct size of the errors decreases only slightly from 10.0 to 8.9. Analysis of the results from the 30 datasets shows a similar phenomenon—for 27 of the 30 datasets the EC decreases as the training set size decreases.

These results suggest that the definition of small disjuncts should factor in training set size. To investigate this further, the error rates of disjuncts with specific sizes (0, 1, 2, etc.) were compared as the training set size was varied. Because disjuncts of a specific size for most concepts cover very few examples, statistically valid comparison were possible for only 4 of the 30 datasets. The results for the Coding dataset are shown in Figure 7. Results for the other datasets are available in Weiss and Hirsh (2000).

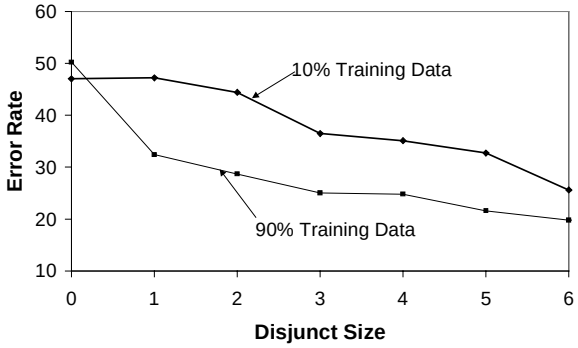

Figure 7: Effect of Training Size on Disjunct Error Rate

Figure 7 shows that the error rates for the smallest disjuncts decrease significantly when the training set size is increased. These results further suggest that the definition of small disjuncts should take training set size into account.

The Effect of Noise

Rare cases cause small disjuncts to be formed in learned concepts. The inability to distinguish between these rare cases (i.e., true exceptions) and noise may be largely responsible for the difficulty in learning in the presence of noise. This conjecture was investigated using synthetic datasets (Weiss 1995) and two real-world datasets (Weiss and Hirsh 1998). We extend this previous work by analyzing 27 datasets (technical difficulties prevented us from handling 3 of the datasets). We restrict the discussion to (random) class noise, since the differing number of attributes in each dataset makes it difficult to fairly compare the impact of attribute noise across datasets.

Although detailed results for class noise are presented elsewhere (Weiss and Hirsh 2000) the results indicate that there is a subtle trend for datasets with higher EC values to experience a greater increase in error rate from class noise. What is much more apparent, however, is that many concepts with low EC values are *extremely* tolerant of noise, whereas none of the concepts with high EC's are. For example, two of the low-EC datasets, blackjack and labor, are so tolerant of noise that when 50% random class noise is added to the training set (i.e., the class value is replaced with a randomly selected valid value 50% of the time), the error rate on the test set increases by less than 1%. The other effect is that as the amount of class noise is increased, the EC tends to decrease. Thus, as noise is added, across almost all of the concepts a greater percentage of the errors come from the larger disjuncts. This helps explain why we find a low-ER/high-EC group of concepts and a high-ER/medium-EC group of concepts: adding noise to concepts in the former increases their error rate and decreases their error concentration, making them look more like concepts in the latter group.

Noise also sometimes causes a radical change in the size of the disjuncts. For low-ER/high-EC group, 10% class noise causes the mean disjunct size of these concepts to shrink, on average, to one-ninth the original size. For the datasets in the high-ER/low-EC group, the same level of noise causes almost no change in the mean disjunct size—the average drops by less than 1%.

Discussion

Many of the results in this paper can be explained by understanding the role of small disjuncts in learning. Learning algorithms tend to form large disjuncts to cover general cases and small disjuncts to cover rare cases. Concepts with many rare cases are harder to learn than those with few, since general cases can be accurately sampled with less training data. This supports the results in Table 1. Concepts in the low-ER/high-EC group contain very general cases—the largest disjunct in these concepts cover, on average, 43% of the correctly classified training examples. The general cases are learned very accurately (the largest disjunct learned in all 10 runs of the Vote dataset never covers *any* test errors). The datasets that are easy to learn and have low error rates have high EC values because so few errors occur in the large disjuncts.

Pruning is the most widespread strategy for addressing the problem with small disjuncts. As was shown earlier, pruning eliminates many small disjuncts. The emancipated examples are then classified using other disjuncts. While this tends to cause the error rate of these other disjuncts to increase, the overall error rate of the concept tends to decrease. Pruning reduces C4.5's average error rate on the 30 datasets from 18.4% to 17.5%, while reducing the EC from 84.8 to 71.2. While this average 0.9% error rate reduction is significant, it is useful to compare this reduction to an "idealized" strategy, where the error rate for the small disjuncts is equal to the error rate of the other (i.e., medium and large) disjuncts. While such a strategy is not achievable, it provides a way of gauging the effectiveness of pruning at addressing the problem of small disjuncts.

Table 2 compares the error rates (averaged over the 30 datasets) resulting from various strategies. The idealized strategy is applied using two scenarios, where the smallest disjuncts covering a total of 10% (20%) of the training examples were assigned an error rate based on the remaining 90% (80%) of the examples.

No Pruning	Default Pruning	Idealized (10%)	Idealized (20%)
18.4%	17.5%	15.2%	13.5%

Table 2: Comparison of Pruning to Idealized Strategy

Table 2 shows that the idealized strategy, even when applied to only 10% of the examples, significantly outperforms C4.5's pruning strategy. These results provide a motivation for finding strategies that better address the problem with small disjuncts.

For many real-world problems, such as identifying those customers likely to buy a product, one is more interested in finding individual classification rules that have high precision than in finding the concept with the best *overall* accuracy. In these situations, pruning seems a questionable strategy, since it tends to decrease the precision of the larger (presumably more precise) disjuncts. To investigate this further, precision/recall curves were generated with and without pruning for each dataset, by starting with the largest disjunct and progressively adding smaller disjuncts. The curves for all 30 datasets were averaged, and the results are shown in Figure 8.

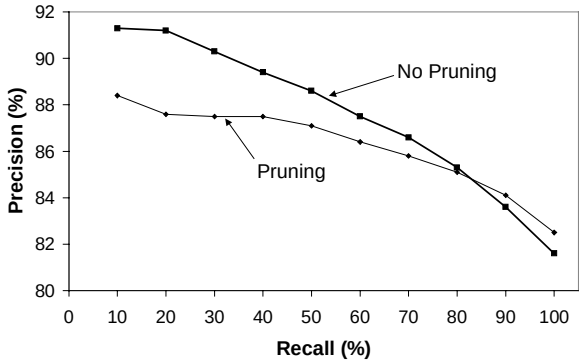

Figure 8: Effect of Pruning on Precision/Recall Curve

The figure shows that while pruning improves predictive accuracy (precision at 100% recall), it reduces the precision of a solution for most values of recall. The break-even point occurs at only 80% recall. This suggests that the use of pruning should be tied to the performance metric most appropriate for a given learning task.

Identifying Error Prone Small Disjuncts

Almost all strategies for addressing the problem with small disjuncts treat small and large disjuncts differently. Consequently, if we hope to address this problem, we need a way to effectively distinguish between the two. The definition that a small disjunct is a disjunct that correctly classifies few training examples (Holte, et al. 1989) is not particularly helpful in this context. What is needed is a method for determining a good threshold *t*, such that disjuncts with size less than *t* have a much higher error rate than those with size greater than *t*. Based on our results we suggest that the threshold *t* should be based on the relationship between disjunct size and error rate, since error rate is not related to disjunct size in a simple way, and more specifically, using error concentration. Based on the EC curve in Figure 2, for example, it seems reasonable to conclude that the threshold for the Vote dataset should be 4, 16, or a value in between. For datasets such as Market2 or Labor, where the EC is very low, we may choose not to distinguish small disjuncts from large disjuncts at all.

Conclusion

This paper provides insight into the role of small disjuncts in learning. By measuring error concentration on concepts induced from 30 datasets, we demonstrate that the problem with small disjuncts occurs to varying degrees, but is quite severe for many of these concepts. We show that even after pruning the problem is still evident, and, by using RIPPER, showed that our results are not an artifact of C4.5.

Although the focus of the paper was on measuring and understanding the impact of small disjuncts on learning, we feel our results could lead to improved learning algorithms. First, error concentration can help identify the threshold for categorizing a disjunct as small, and hence can be used to improve the effectiveness of variable bias system in addressing the problem with small disjuncts. The EC value could also be used to control the pruning strategy of a learning algorithm, since low EC values seems to indicate that pruning may actually decrease predictive accuracy. A high EC value is also a clear indication that one is likely to be able to trade-off reduced recall for greatly improved precision.

References

Ali, K. M. and Pazzani, M. J. 1992. Reducing the Small Disjuncts Problem by Learning Probabilistic Concept Descriptions, in T. Petsche editor, *Computational Learning Theory and Natural Learning Systems*, Volume 3.

Blake, C. L. and Merz, C. J. 1998. UCI Repository of ML Databases, http://www.ics.uci.edu/~mlearn/MLRepository.html, Irvine, CA: University of California, Department. of Computer Science.

Cohen, W. 1995. Fast Effective Rule Induction. In *Proceedings of the Twelfth International Conference on Machine Learning*, 115-123.

Cohen, W. and Singer, Y. 1999. A Simple, Fast, and Effective Rule Learner. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence*, 335-342. Menlo Park, Calif.: AAAI Press.

Danyluk, A. P. and Provost, F. J. 1993. 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*, 81-88.

Holte, R., C., Acker, L. E., and Porter, B. W. 1989. Concept Learning and the problem of small disjuncts. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, 813-818. San Mateo, CA: Morgan Kaufmann.

Quinlan, J. R. 1993. *C4.5: Programs for Machine Learning*. San Mateo, CA: Morgan Kaufmann.

Ting, K. M. 1994. The Problem of Small Disjuncts: its Remedy in Decision Trees. In *Proceedings of the Tenth Canadian Conference on Artificial Intelligence*, 91-97.

Van den Bosch, A., Weijters, A., Van den Herik, H.J. and Daelemans, W. 1997. When Small Disjuncts Abound, Try Lazy Learning: A Case Study. In *Proceedings of the Seventh Belgian-Dutch Conference on Machine Learning*, 109-118.

Weiss, G. M. 1995. Learning with Rare Cases and Small Disjuncts. In *Proceedings of the Twelfth International Conference on Machine Learning*, 558-565.

Weiss, G. M. and Hirsh, H. 1998. The Problem with Noise and Small Disjuncts. In *Proceedings of the Fifteenth International Conference on Machine Learning*, 574-578.

Weiss, G. M. and Hirsh, H. 2000. A Quantitative Study of Small Disjuncts: Experiments and Results, Technical Report, ML-TR-42, Computer Science Department., Rutgers University. Also available from http://www.cs.rutgers.edu/~gweiss/papers/index.html.