# Concept Learning and the Problem of Small Disjuncts*

March 31, 1995

**Robert C. Holte**
Computer Science Dept.
University of Ottawa
Ottawa, Canada K1N 6N5

**Liane E. Acker**
IBM-Austin

**Bruce W. Porter**
Department of Computer Sciences
University of Texas at Austin
Austin, Texas 78712

**Abstract**

Ideally, definitions induced from examples should consist of all, and only, disjuncts that are meaningful (*e.g.,* as measured by a statistical significance test) and have a low error rate. Existing inductive systems create definitions that are ideal with regard to large disjuncts, but far from ideal with regard to small disjuncts, where a small (large) disjunct is one that correctly classifies few (many) training examples. The problem with small disjuncts is that many of them have high rates of misclassification, and it is difficult to eliminate the error-prone small disjuncts from a definition without adversely affecting other disjuncts in the definition. Various approaches to this problem are evaluated, including the novel approach of using a bias different than the "maximum generality" bias. This approach, and some others, prove partly successful, but the problem of small disjuncts remains open.

*Support for this research was provided by the Army Research Office under grant ARO-DAAG29-84-K-0060 and the National Science Foundation under grant IRI-8620052.



## The Problem of Small Disjuncts

Systems that learn from examples do not usually succeed in creating a purely conjunctive definition for each concept. Instead, they create a definition that consists of several disjuncts, where each disjunct is a conjunctive definition of a subconcept of the original concept. Table 1 (column 2) shows the number of disjuncts in definitions induced in several different domains by different systems.

The "coverage" of a disjunct is defined as the number of training examples it correctly classifies. A disjunct is called "small" if its coverage is low. Table 1 (column 3) shows the coverage of disjuncts in induced definitions.

There are several reasons for paying special attention to the methods by which small disjuncts are created. First, many concepts include rare or exceptional cases and it is desirable for induced definitions to cover these cases, even if they can only be covered by augmenting the definitions with small disjuncts. Secondly, small disjuncts constitute a significant portion of an induced definition, in the sense that often they collectively match more than 20% of the examples that satisfy a definition.

The problem with small disjuncts, and the main reason for reviewing the methods by which they are created, is that they are much more error prone than large disjuncts. Table 2 illustrates this phenomenon with the definitions created by CN2 [CN89] in a chess endgame domain [Sha87]. The error rate of small disjuncts is high, whereas the error rate of large disjuncts is almost zero. Disjuncts of coverage 10 or less commit 95% of the errors (column 6) even though they match only 41% of the examples (column 5). This pattern of errors is not unique to CN2, or to this domain. A similar pattern occurs in the definitions created by ID3 [Qui86] in this domain, and in the definitions created by CN2 in the lymphography domain [CN87] (Table 3).

Ideally, induced definitions should consist of all, and only, disjuncts that are meaningful (*e.g.,* as measured by a statistical significance test) and have a low error rate. Definitions created by existing methods are ideal with regard to large disjuncts, but far from ideal with regard to small disjuncts. The remainder of this paper evaluates three approaches to eliminating error-prone small disjuncts from a definition without adversely affecting other disjuncts in the definition.

| *System: Domain* | *# Disjuncts* | *Coverage of Disjuncts* | *Concept* |
|---|---|---|---|
| CN2: (99% significance) | | | |
| chess endgame | 8 | 39,18,10,8,7,7,6,5 | win for pawn's side |
| | 8 | 37,10, 8,7,6,5,5,4 | no win for pawn's side |
| lymphography | 4 | 16,10,7,4 | metastases |
| | 4 | 12,7,5,4 | malignant lymphoma |
| AQ15: (truncating all but the largest disjunct) | | | |
| lymphography | 1 | not available | metastases |
| | 1 | " | malignant lymphoma |
| AQ11: | | | |
| soybean | 2 | 34,24 | phytophthora root rot |
| | 2 | 13, 7 | brown stem rot |
| | 3 | 30,19,12 | brown spot |
| | 3 | 10, 7, 5 | anthracnose |
| | 7 | 20,13, 9,8,8,5,4 | frog eye leaf spot |
| | 8 | 13,11,10,8,7,6,3,1 | alternaria leaf spot |
| PROTOS: | | | |
| audiology | 12 | 12, 7,5,4,4,3,2,2,2,2,2,2 | cochlear - unknown |
| | 8 | 20,10,5,4,2,2,2,2 | cochlear - age |
| | 6 | 6, 3,3,3,2,2 | cochlear - possible noise |
| | 4 | 6, 5,3,3 | normal ear |
| | 4 | 13, 3,3,2 | cochlear - age and noise |
| METADENDRAL: | | | |
| aliphatic amines | 5 | not available | bond will break in m.s. |
| estrogenic steroids | 8 | " | " |
| monoketoandrostanes | 8 | " | " |
| diketoandrostanes | 8 | " | " |
| triketoandrostanes | 10 | " | " |

Table 1: This table indicates the number of disjuncts and the coverage of disjuncts in definitions induced by different systems in different domains. This information is gathered from several sources (see Appendix).



| CN2 (99%), Chess endgame domain | | | | | |
|---|---|---|---|---|---|
| *Coverage* | *# Matched* | *# Errors* | *Error Rate* | *% Matched (cumulative)* | *% Errors (cumulative)* |
| 4 | 734 | 121 | 16 | 3 | 6 |
| 5 | 2394 | 373 | 16 | 11 | 25 |
| 6 | 1986 | 404 | 20 | 19 | 45 |
| 7 | 3076 | 423 | 14 | 30 | 67 |
| 8 | 1665 | 295 | 18 | 36 | 82 |
| 9-10 | 1375 | 256 | 19 | 41 | 95 |
| 11-13 | 1111 | 45 | 4 | 45 | 97 |
| 14-30 | 3099 | 60 | 2 | 56 | 100 |
| 31-40 | 5544 | 0 | 0 | 76 | 100 |
| >40 | 6467 | 0 | 0 | 100 | 100 |

Table 2: Data about disjuncts of different coverages in the definitions produced by CN2 (with a significance threshold of 99%) in the KPa7KR chess endgame domain. These numbers do not include the disjuncts corresponding to CN2's default rule, which all are small and have error rates around 50%. 9 training sets of 200 examples each were independently drawn from the dataset of 3196 examples. The definitions produced were evaluated on the entire dataset. Column 2 gives the number of test examples matched, column 3 the number of misclassifications, by disjuncts with the coverage. These numbers are the totals over all the definitions. Column 4 gives the ratio of misclassifications to matches. Column 5 gives the percentage of test examples matched by disjuncts whose coverage is equal to or less than the value in column 1. This value, for row X, is calculated by summing the entries in column 2 in rows X and above, and dividing by the sum of all entries in column 2. Column 6 gives the percentage of misclassifications made by disjuncts whose coverage is equal to or less than the value in column 1.

| *Coverage* | ID3, Chess endgame domain *Error Rate (%)* | *% Matched (cumulative)* | *% Errors (cumulative)* | CN2, Lymphography domain *Error Rate (%)* | *%Matched (cumulative)* | *% Errors (cumulative)* |
|---|---|---|---|---|---|---|
| 1-3 | 22 | 5 | 25 | 39 | 5 | 9 |
| 4 | 25 | 8 | 45 | 3 | 11 | 10 |
| 5 | 12 | 12 | 57 | 46 | 18 | 25 |
| 7 | 9 | 14 | 61 | 27 | 23 | 31 |
| 8 | 11 | 17 | 68 | 28 | 30 | 39 |
| 9-10 | 8 | 19 | 72 | 32 | 38 | 50 |
| 11-13 | 1 | 21 | 73 | 28 | 47 | 61 |
| 14-30 | 4 | 38 | 90 | 19 | 69 | 79 |
| 31-40 | 0 | 72 | 93 | 15 | 100 | 100 |
| >40 | 1 | 100 | 100 | | | |

Table 3: Data corresponding to columns 4, 5, and 6 in Table 2 for the definitions produced by ID3 in the KPa7KR chess endgame domain (left side), and the definitions produced by CN2 (with a significance threshold of 99%) in the lymphography domain (right side). This version of ID3 did no pruning. In the KPa7KR domain, 5 training sets of 200 examples each were independently drawn from the same dataset used in the experiment in Table 2. The 5 definitions produced were evaluated on the entire dataset. In the lymphography domain, 10 runs were made. In each run, the dataset of 142 examples was divided into two equal parts, one for training, the other for testing.



## Approach 1: Eliminate All Small Disjuncts

The most direct means of eliminating error-prone small disjuncts is to eliminate all small disjuncts by explicitly refusing to create disjuncts whose coverage is below a certain threshold.[1] An immediate objection to this policy is that it has the undesirable effect of creating definitions that do not include the unusual cases of a concept (represented by small but significant disjuncts).

A second objection to eliminating all small disjuncts from a definition is that doing so may significantly increase the definition's error rate. The net effect of eliminating all small disjuncts is difficult to predict, because it depends on the fate of the "emancipated" examples – the examples that were classified by the disjuncts that have been deleted. Table 2 (column 5) and Table 3 (columns 3 and 6), indicate the percentage of examples emancipated by eliminating all disjuncts up to a certain coverage.

Some emancipated examples will match disjuncts that have not been deleted. These may be classified correctly or incorrectly, and a disjunct's error rate on emancipated examples may be much higher than its original error rate. Emancipated examples that fail to match any disjunct may be assigned a default classification, or allowed to pass as errors of omission. Most existing systems have rules, called default rules, for assigning a default classification. These rules often have very high error rates. Consequently, in these systems, there is a considerable chance that emancipated examples will be misclassified. The only examples that ought to be emancipated are those that match disjuncts with high error rates, say, 25% or more. Small disjuncts, although much more error-prone than large disjuncts, do not consistently have error rates high enough to justify a policy of eliminating all small disjuncts.

An error of omission occurs when a test example is not assigned a classification. It is a indication that several classifications of the example are equally strongly supported by the training set. In many circumstances, errors of omission are more desirable than extremely error-prone default classifications. For this reason, the discussion of "approach 3" gives equal consideration to definitions with default rules and those without.

[1] The Nmin parameter in CART [BFOS84] is this type of threshold. ASSISTANT [CKB87] specifies this threshold as a percentage of the original training set. In both CART and ASSISTANT this cutoff is used in building a decision tree that is subsequently pruned.

## Approach 2: Eliminate Undesirable Disjuncts

Techniques that directly measure, or estimate, the significance and error rate of disjuncts are used in several systems (*e.g.,* CN2, CART [BFOS84], ASSISTANT [CKB87], and recent versions of ID3). These techniques reliably eliminate undesirable large disjuncts (*i.e.,* ones that are not meaningful, or have a high error rate), but, as currently used, do not reliably eliminate undesirable small disjuncts. This section considers the prospects of strengthening these techniques so that they reliably eliminate undesirable disjuncts, small and large alike.

### Significance Testing

Tests of statistical significance are used in some systems to determine whether or not to include a disjunct in a definition. Definitions produced using these tests tend to have fewer disjuncts, larger disjuncts, and slightly lower error rates than definitions produced without using them.

Disjuncts whose coverage is too low do not pass significance tests. The coverage at which disjuncts become "insignificantly small" is determined by the significance threshold chosen for the test (typically 90-99%), the number of concepts, and the distribution of training examples among concepts. For example, if a training set has an equal number of examples of two concepts, a disjunct is 99% significant if and only if its coverage is 7 or more. Because significance tests eliminate all small disjuncts, they are subject to the objections raised in the preceding section.

A further problem arises with systems that do not use true significance tests. Most systems use tests that accurately approximate significance tests only for large disjuncts. Some of these systems, such as CN2, apply the approximate tests to small disjuncts despite their inaccuracy. Others, such as ID3, refrain from testing the significance of small disjuncts[2]. In any case, the significance of small disjuncts is not reliably estimated, with the undesirable result that significant small disjuncts may be eliminated and insignificant ones retained. This problem is not insuperable: [Nib87] gives an exact test for significance.

### Error-Rate Estimation

Error rate cannot be tested exactly: it can only be estimated. Like approximate tests of significance, techniques for estimating error rate are not entirely reliable for small disjuncts.

[2] [Qui86, page 154]. The action taken in lieu of a significance test is not described.



For example, [CKB87] reports that the technique of Niblett and Bratko[3] "seems to make rather pessimistic estimation about the information contained in learning data (it overestimates the error rate of subtrees) ... post-pruning is too drastic when the number of learning examples per class per attribute is low."

### The Need for Both Significance and Error-Rate Testing

No existing system tests both significance and error rate. "Pre-pruning" systems use significance testing; "post-pruning" systems use error-estimation [Nib87]. Indeed, post-pruning systems have a rather strong disregard for the significance of disjuncts, their sole objective being to eliminate from a definition as many disjuncts as possible without suffering too great an increase in error rate. A one-test approach is sufficient to eliminate undesirable large disjuncts, because for large disjuncts, significance and error rate are highly correlated.

However, in order to eliminate all undesirable small disjuncts, it is necessary to test both significance and error rate. This is because, for small disjuncts, error rate is not related to significance in any simple way. Neither is it related to "entropy", a measure that is often used in conjunction with significance tests. The lack of a simple relation between error rate and entropy is evident in the definitions produced by CN2[4]. CN2 creates disjuncts one at a time, evaluating, at each step, the entropy of all possible new disjuncts on the portion of the training set not covered by existing disjuncts. The new disjunct with the lowest entropy is included in the definition only if it passes a significance test. Thus, the disjunct selected at a given step had lower entropy, at that step, than the disjuncts selected at later steps, and it was statistically significant. If error rate were related to entropy, disjuncts in neighbouring steps would have similar error rates. That this does not occur is evident in the following data, which describes a typical definition.

| *Step #* | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | default |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *Error Rate* | 18 | 0 | 0 | 3 | 2 | 42 | 8 | 0 | 2 | 0 | 5 | 9 | 0 | 0 | 12 | 48 |
| *Coverage* | 8 | 36 | 58 | 5 | 19 | 5 | 19 | 7 | 5 | 8 | 8 | 5 | 5 | 6 | 5 | |

[3] This and other error-estimation techniques are described in [Nib87, page 43].

[4] Clark and Niblett have observed empirically that rules of low entropy tend to have high significance ([CN87, page 18] mistakenly reports this as a relation between rules of high entropy and high significance). Thus, if error rate is not related to entropy, then neither is it related to significance.

## Approach 3: Make Small Disjuncts Highly Specific

The techniques considered in the previous sections have all been based on properties (coverage, significance, entropy, and error rate) that are defined in terms of the set of training examples that match a disjunct. There will usually be many different disjuncts that match the same set of training examples, and these will all be indistinguishable by the previous techniques. That is, they will all have identical estimates of error rate, significance, entropy, and so on.

To select among disjuncts that are indistinguishable on the basis of the training set, inductive systems employ an extra-evidential preference criterion, or "bias" [Mit80]. Definitions produced using different biases, will usually have different error rates and different distributions of errors across disjuncts. It is possible that the problem of error-prone small disjuncts is caused by the use of the "maximum generality" bias (defined below). This bias is used by many inductive systems, including ID3 and CN2. The use of a different bias might result in definitions in which all disjuncts, large and small alike, have low error rates. This approach has been explored experimentally, by comparing the definitions produced by CN2 when it is biased in different ways.

Three biases are compared in this section. All are defined in terms of a disjunct's "specificity", which is defined as the number of conditions in a disjunct (recall that a disjunct is the conjunction of one or more conditions). "Generality" is the opposite of specificity. To compare the definitions produced by the different biases, a training set of about 200 examples was drawn at random from the 3196 examples in the KPa7KR (chess endgame) dataset. CN2, using each bias, was run on the training set. The definitions produced were evaluated using the entire dataset. This procedure was repeated for 9 independently drawn training sets. The cumulative results of these 9 runs are given in Table 4 (see [Ack88] for more details).

CN2's original bias is the maximum generality bias. An inductive system using this bias, having decided to create a disjunct that matches a particular set of training examples, selects a maximally general disjunct that matches those examples and no others. The definitions produced by CN2 using this bias are described in Table 2 and the top row of Table 4. The problem of error-prone small disjuncts is evident in this data. Small disjuncts (coverage 5 or less) have an error rate of 16%, whereas large disjuncts have an error rate of 6.1%. Ignoring examples classified by the default rule, small disjuncts commit about 25% of the errors even though they match only about 10% of the examples.



| | *Small Disjuncts (coverage $\leq 5$)* | | | *Large Disjuncts* | | | *Default Rule* | | | *Overall* |
|---|---|---|---|---|---|---|---|---|---|---|
| *Bias* | *E* | *M* | *(P)* | *E* | *M* | *(P)* | *E* | *M* | *(P)* | *P* |
| maximum generality | 494 | 3128 | (16%) | 1483 | 24323 | (6.1%) | 613 | 1313 | (47%) | 9.0% |
| maximum specificity | 174 | 1707 | (10%) | 1998 | 25386 | (7.9%) | 794 | 1671 | (48%) | 10.3% |
| selective specificity | 279 | 2453 | (11%) | 1682 | 24810 | (6.8%) | 730 | 1501 | (49%) | 9.4% |

Table 4: Data indicating the effect of using different biases on the definitions produced by CN2 in the KPa7KR chess endgame domain. Small disjuncts are those of coverage 5 or less. Column 1 indicates the bias used for small disjuncts; the maximum generality bias is always used for large disjuncts. See the text for a description of these biases, and the experimental setup. Columns labelled E give the number of misclassifications, columns labelled M the total number of matched test examples. These numbers are totals over all the runs using the bias specified in column 1. Columns labelled P give the ratio of misclassifications to matches (E/M).

The maximum generality bias works well for large disjuncts, but not for small disjuncts. This suggests restricting its use to large disjuncts, and using a different bias for small disjuncts. The maximum specificity bias seems, on the face of it, to be appropriate for small disjuncts. An inductive system using this bias, having decided to create a disjunct that matches a particular (small) set of training examples, selects the disjunct consisting of all the conditions that are satisfied by those examples.

The middle row in Table 4 describes the definitions produced by CN2 using the maximum generality bias for large disjuncts and the maximum specificity bias for small disjuncts. In these definitions, the large disjuncts are identical to those in the original definitions, but the small disjuncts are maximally specific instead of being maximally general. The small disjuncts created using this bias match many fewer examples than are matched by the small disjuncts created using the original bias (1707 compared to 3128). The error rate of small disjuncts has decreased considerably, indicating that the 1421 examples emancipated by using maximally specific disjuncts were a major source of error.

Unfortunately, use of the maximum specificity bias for small disjuncts has adverse affects on other parts of the definition. The emancipated examples, 75% of which are classified by large disjuncts, are misclassified at a rate of almost 50%, which is double the rate at which they were misclassified by the small disjuncts. Consequently, there is a net increase in the error rate of the definitions that is unacceptably large.

The maximum specificity bias moves in the right direction, but it goes too far. Using a "selective specificity" bias, an inductive system, having decided to create a disjunct that matches a particular (small) set of training examples, would select the disjunct consisting of the conditions that are satisfied by those examples and that meet certain other requirements. These other requirements are what make the specificity selective. A disjunct produced using this type of bias may be maximally specific, maximally general, or neither, depending on whether all, none, or some of the conditions meet the requirements.

The particular selective specificity bias used in this experiment required the conditions in the disjunct for subset $S$ of training set $T$ to match no more than 25% of the examples in $T-S$ whose class differs from that of the majority of $S$. For example, suppose there are two classes, $C_1$ and $C_2$, that the majority of examples in $S$ are in $C_1$, and that $G$ is a maximally general disjunct for $S$. Then a condition matching all the examples in $S$ is added to $G$, according to this selective specificity bias, if and only if it matches fewer than 25% of the $C_2$ examples in $T-S$.

The bottom row in Table 4 describes the definitions produced by CN2 using the maximum generality bias for large disjuncts and the selective specificity bias for small disjuncts. The small disjuncts produced using the selective specificity bias are superior to those produced using the other biases. They have a reasonably low error rate, which they did not have when the maximum generality bias was used, and they are doing a significant amount of the classification, which they did not do when the maximum specificity bias was used. When the selective specificity bias is used for small disjuncts, large disjuncts have a slightly higher error rate than when the maximal generality bias is used. Likewise, the error rate of definitions is slightly higher using the selective specificity bias than it is using the maximum generality bias. However, this difference is due entirely to the error rates of the default rules. Ignoring the default rules, the definitions produced by both biases match almost the same number of examples (95% of the test set) and have almost identical error rates (7.2%). Thus, the problem of error-prone small disjuncts is solved, to a significant degree, by using the maximum generality bias for large disjuncts and the selective specificity bias for small disjuncts.

The success of this approach depends, to some extent, on having defined "small" as $coverage \leq 5$. Table 5 gives the results of repeating the preceding experiments with "small" defined as $coverage \leq 9$. These results are similar to the previous ones in three important



| | Small Disjuncts ($coverage \leq 9$) | | | Large Disjuncts | | | Default Rule | | | Overall |
|---|---|---|---|---|---|---|---|---|---|---|
| *Bias* | *E* | *M* | *(P)* | *E* | *M* | *(P)* | *E* | *M* | *(P)* | *P* |
| maximum generality | 1789 | 10535 | (17%) | 188 | 16916 | (1.1%) | 613 | 1313 | (47%) | 9.0% |
| selective specificity | 1007 | 8142 | (12%) | 335 | 17598 | (1.9%) | 1583 | 3024 | (52%) | 10.2% |

Table 5: Same as Table 4, except here "small" means coverage 9 or less.

ways. First, the error rate of small disjuncts is reasonably low when the selective specifity bias is used for small disjuncts but not when the maximal generality bias is used. Secondly, both biases produce large disjuncts with low error rates. Thirdly, the error rate of definitions is higher when the selective specifity bias than when the maximal generality bias is used, and this difference is entirely due to the increased use and error rate of the default rule.

There are also significant differences between the definitions produced using the different definitions of "small". The error rate of definitions is considerably lower using $coverage \leq 5$. However, if default rules are ignored, the opposite is true. Using $coverage \leq 9$, the collective error rate of large and small disjuncts is only 5.2%, compared to 7.2% using $coverage \leq 5$. On the other hand, using $coverage \leq 9$, the large and small disjuncts collectively match only 90% of the test examples, compared to 95% using $coverage \leq 5$.

## Conclusions

This paper has demonstrated that existing concept learning systems do well at creating large disjuncts, but poorly at creating small ones. Some of the causes of this poor behaviour have been identified. Improvements that are suggested by this analysis are (1) use exact significance tests; (2) test both significance and error-rate; and (3) use errors of omission instead of default classifications whenever possible. A fourth suggestion, that different biases ought to be used for large and small disjuncts, was investigated experimentally. The use of the maximum generality bias for large disjuncts and a selective specificity bias for small disjuncts partly solved the problem of small disjuncts. This result is relatively insensitive to the exact definition of "small".

## Acknowledgements

We are especially grateful to Peter Clark, of the Turing Institute (Glasgow), who provided the source for CN2, the definitions it produced on the lymphography dataset, and the KPa7KR dataset generated by Alen Shapiro[Sha87]. Peter also answered many questions, and gave helpful criticism of early drafts of this paper. We also wish to to thank Ray Mooney, for his reconstruction of ID3, Ray Bareiss, for providing the definitions produced by Protos, and Larry Rendell, for his comments on an early draft of this paper.

### Appendix: The Sources of Data Used in Table 1

The definitions induced by METADENDRAL are from [BSW+76]. The definitions induced by CN2 in the chess endgame domain are original [Ack88]. The definitions induced by CN2 in the lymphography domain were provided by Peter Clark, of the Turing Institute (Glasgow). Certain properties of these definitions are reported in [CN87]. In both these domains, there existed definitions based on several different training sets. These definitions varied, of course, in the number of disjuncts and the coverage of the disjuncts. Table 1 describes typical definitions.

The definitions induced by AQ15 are from [Mic87], and those by AQ11 from [MC80]. AQ11 is the only system surveyed that produces definitions whose disjuncts overlap. Coverage, in this case, may be defined in several ways. Table 1 reports the total number of examples matched by a disjunct. This gives larger values for coverage than alternative definitions, such as the number of examples matched by the disjunct and no other disjunct.

The definitions induced by Protos were provided by Ray Bareiss, of Vanderbilt University. Certain properties of these definitions are reported in [BPW87]. Protos is an incremental, exemplar-based learning system. Unlike all the other systems surveyed, which are nonincremental and rule-based, Protos does not attempt to minimize the number of disjuncts in the definitions it produces. Consequently, Protos's definitions often have many disjuncts with a coverage of 1 (there were an average of 10 such disjuncts in each definition described in Table 1). To prevent the Protos definitions from skewing the data, these disjuncts have been ignored.

Finally, some induced rules reported in the literature have been deliberately excluded from the table. Two of the 4 definitions created by CN2 and AQ15 in the lymphography domain, 9 of the 15 definitions created by AQ11 in the soybean domain, and 21 of the 26 definitions



created by Protos in the audiology domain have been excluded because they are based on too few training examples (10 or less). Most of these definitions had one disjunct. All definitions in the thyroid disease domain [KA87], the protein secondary structure domain [Kin87], and the breast cancer domain [CN87] have been excluded because their classification accuracy is not significantly better than that achieved by assigning every example to the most common class in the domain. All definitions created by CN2 in the primary tumor domain [CN87] have been excluded because their classification accuracy is well below 50%.

## References

[Ack88] Liane E. Acker. Varying the degree of generalization in concept learning: An empirical study. Technical Report AI88-89, Computer Sciences Department, University of Texas at Austin, USA 78712, 1988.

[BFOS84] Leo Breiman, Jerome H. Friedman, Richard A. Olshen, and Charles J. Stone. *Classification and Regression Trees*. Wadsworth International Group, Belmont, California, 1984.

[BPW87] E. Ray Bareiss, Bruce W. Porter, and Craig C. Wier. Protos: An exemplar-based learning apprentice. In Pat Langley, editor, *Proceedings of the Fourth International Workshop on Machine Learning*, pages 12–23. Morgan Kaufmann, Los Altos, California, 1987.

[BSW+76] Bruce G. Buchanan, D.H. Smith, W.C. White, R.J. Gritter, E.A. Feigenbaum, J. Lederberg, and Carl Djerassi. Applications of artificial intelligence for chemical inference. 22. automatic rule formation in mass spectrometry by means of the Meta-DENDRAL program. *Journal of the American Chemical Society*, 98(20):6168–6178, 1976.

[CKB87] B. Cestnik, I. Kononenko, and I. Bratko. ASSISTANT 86: A knowledge-elicitation tool for sophisticated users. In Ivan Bratko and Nada Lavrac, editors, *Progress in Machine Learning*, pages 31–45. Sigma Press, Wilmslow, England, 1987.

[CN87] Peter Clark and Tim Niblett. Induction in noisy domains. In Ivan Bratko and Nada Lavrac, editors, *Progress in Machine Learning*, pages 11–30. Sigma Press, Wilmslow, England, 1987.

[CN89] Peter Clark and Tim Niblett. The CN2 induction algorithm. *Machine Learning*, 1989. (to appear).

[KA87] Dennis Kibler and David W. Aha. Learning representative exemplars of concepts: An initial case study. In Pat Langley, editor, *Proceedings of the Fourth International Workshop on Machine Learning*, pages 24–30. Morgan Kaufmann, Los Altos, California, 1987.

[Kin87] Ross D. King. An inductive learning approach to the problem of predicting a protein's secondary structure from its amino acid sequence. In Ivan Bratko and Nada Lavrac, editors, *Progress in Machine Learning*, pages 230–250. Sigma Press, Wilmslow, England, 1987.

[MC80] R.S. Michalski and R.L. Chilausky. Knowledge acquisition by encoding expert rules versus computer induction from examples: A case study involving soybean pathology. *International Journal of Man-Machine Studies*, 12:63–87, 1980.

[Mic87] R. S. Michalski. How to learn imprecise concepts: A method for employing a two-tiered knowledge representation in learning. In Pat Langley, editor, *Proceedings of the Fourth International Workshop on Machine Learning*, pages 50–58. Morgan Kaufmann, Los Altos, California, 1987.

[Mit80] Tom M. Mitchell. The need for biases in learning generalizations. Technical Report CBM-TR-117, Computer Science Department, Rutgers University, New Jersey, 1980.

[Nib87] Tim Niblett. Constructing decision trees in noisy domains. In Ivan Bratko and Nada Lavrac, editors, *Progress in Machine Learning*, pages 67–78. Sigma Press, Wilmslow, England, 1987.

[Qui86] J. Ross Quinlan. The effect of noise on concept learning. In J.G. Carbonell R.S. Michalski and T.M. Mitchell, editors, *Machine Learning: An Artificial Approach, Volume II*, pages 149–166. Morgan Kaufmann, Los Altos, California, 1986.

[Sha87] Alen D. Shapiro. *Structured Induction in Expert Systems*. Addison-Wesley, 1987.