A Case Study for Learning from Imbalanced Data Sets

Aijun An, Nick Cercone, and Xiangji Huang

Department of Computer Science, University of Waterloo
Waterloo, Ontario N2L 3G1 Canada
{aan, ncercone, jhuang}@uwaterloo.ca

Abstract. We present our experience in applying a rule induction technique to an extremely imbalanced pharmaceutical data set. We focus on using a variety of performance measures to evaluate a number of rule quality measures. We also investigate whether simply changing the distribution skew in the training data can improve predictive performance. Finally, we propose a method for adjusting the learning algorithm for learning in an extremely imbalanced environment. Our experimental results show that this adjustment improves predictive performance for rule quality formulas in which rule coverage makes positive contributions to the rule quality value.

Keywords: Machine learning, Imbalanced data sets, Rule quality.

1 Introduction

Many real-world data sets exhibit skewed class distributions in which almost all cases are allotted to one or more larger classes and far fewer cases allotted for a smaller, usually more interesting class. For example, a medical diagnosis data set used in [1] contains cases that correspond to diagnoses for a rare disease. In that data set, only 5% of the cases correspond to "positive" diagnoses; the remaining majority of the cases belong to the "no disease" category. Learning with this kind of imbalanced data set presents problems to machine learning systems, problems which are not revealed when the systems work on relatively balanced data sets. One problem occurs since most inductive learning algorithms assume that maximizing accuracy on a full range of cases is the goal [12] and, therefore, these systems exhibit accurate prediction for the majority class cases, but very poor performance for cases associated with the low frequency class. Some solutions to this problem have been suggested. For example, Cardie and Howe [5] proposed a method that uses case-specific feature weights in a case-based learning framework to improve minority class prediction. Some studies focus on reducing the imbalance in the data set by using different sampling techniques, such as data reduction techniques that remove only majority class examples [9] and "up-sampling" techniques that duplicate the training examples of the minority class or create new examples by corrupting existing ones with artificial noise

E. Stroulia and S. Matwin (Eds.): AI 2001, LNAI 2056, pp. 1–15, 2001.

[6]. An alternative to balancing the classes is to develop a learning algorithm that is intrinsically insensitive to class distribution in the training set [11]. An example of this kind of algorithm is the SHRINK algorithm [10] that finds only rules that best summarizes the positive examples (of the small class), but makes use of the information from the negative examples. Another approach to learning from imbalanced data sets, proposed by Provost and Fawcett [13], is to build a hybrid classifier that uses ROC analysis for comparison of classifier performance that is robust to imprecise class distributions and misclassification costs. Provost and Fawcett argued that optimal performance for continuous-output classifiers in terms of expected cost can be obtained by adjusting the output threshold according to the class distributions and misclassification costs. Although many methods for coping with imbalanced data sets have been proposed, there remain open questions. According to [12], one open question is whether simply changing the distribution skew can improve predictive performance systematically. Another question is whether we can tailor the learning algorithm to this special learning environment so that the accuracy for the extreme class values can be improved.

Another important issue in learning from imbalanced data sets is how to evaluate the learning result. Clearly, the standard performance measure used in machine learning - predictive accuracy over the entire region of the test cases is not appropriate for applications where classes are unequally distributed. Several measures have been proposed. Kubat *et al* [11] proposed to use the geometric mean of the accuracy on the positive examples and the accuracy on the negative examples as one of their performance measures. Provost and Fawcette [13] made use of ROC curves that visualize the trade-off between the false positive rate and the true positive rate to compare classifiers. In information retrieval, where relevant and irrelevant documents are extremely imbalanced, recall and precision are used as standard performance measures.

We present our experience in applying rule induction techniques to an extremely imbalanced data set. The task of this application is to identify promising compounds from a large chemical inventory for drug discovery. The data set contains nearly $30,000$ cases, only 2% of which are labeled as potent molecules. To learn decision rules from this data set, we applied the ELEM2 rule induction system [2]. The learning strategies used in ELEM2 include sequential covering and post-pruning. A number of rule quality formulas are incorporated in ELEM2 for use in the post-pruning and classification processes. Different rule quality formulas may lead to generation of different sets of rules, which in turn results in different predictions for the new cases. We have previously evaluated the rule quality formulas on a number of benchmark datasets [3], but none of them is extremely imbalanced. Our objective in this paper is to provide answers to the following questions. First, we would like to determine how each of these rule quality formulas reacts to the extremely imbalanced class distribution and which of the rule quality formulas is most appropriate in this kind of environment. Second, we would like to know whether reducing the imbalance in the

data set can improve predictive performance. Third, we would like to compare different measures of performance to discover whether there is correlation between them. Finally, we would like to know whether a special adjustment of the learning algorithm can improve predictive performance in an extremely imbalanced environment. The paper is organized as follows. In Section 2, we describe our data set and the application tasks related to the data set. We then briefly describe the learning and classification algorithms used in our experiment. In Section 6 we present our experiments and experimental results. We conclude the paper with a summary of our findings from the experiments.

2 Domain of the Case Study

The data set we used was obtained from the National Cancer Institute through our colleagues in the Statistics Department at the University of Waterloo. It concerns the prediction of biological potency of chemical compounds for possible use in the pharmaceutical industry. Highly potent compounds have great potential to be used in new medical drugs. In the pharmaceutical industry, screening every available compound against every biological target through biological tests is impossible due to the expense and work involved. Therefore, it is highly desirable to develop methods that, on the basis of relatively few tested compounds, can identify promising compounds from a relatively large chemical inventory.

2.1 The Data Set

Our data set contains $29,812$ tested compounds. Each compound is described by a set of descriptors that characterize the chemical structure of the molecule and a binary response variable that indicates whether the compound is active or not. 2.04% of these compounds are labeled as active and the remaining ones as inactive. The data set has been randomly split into two equal-sized subsets, each of which contains the same number of active compounds so that the class distribution in either of the subsets remain the same as in the original data set. We use one subset as the training set and the other as the testing test in our experiments.

2.2 Tasks and Performance Measures

One obvious task is to learn classification rules from the training data set and use these rules to classify the compounds in the test set. Since it is the active compounds that are of interest, appropriate measures of classification performance are not the accuracy on the entire test set, but the precision and recall on the active compounds. *Precision* is the proportion of true active compounds among the compounds predicted as active. *Recall* is proportion of the predicted active compounds among the active compounds in the test set.

However, simply classifying compounds is not sufficient. The domain experts would like identified compounds to be presented to them in decreasing order of a prediction score with the highest prediction indicating the most probably active compound so that identified compounds can be tested in biological systems one by one starting with the compound with the highest prediction. Therefore, in addition to classification, the other task is to rank the compounds in the test set according to a prediction score. To be cost effective, it is preferred that a high proportion of the proposed lead compounds actually exhibit biological activity.

3 The Learning Algorithm

ELEM2 [2] is used to learn rules from the above bio-chemistry data set. Given a set of training data, ELEM2 learns a set of rules for each of the classes in the data set. For a class C, ELEM2 generates a disjunctive set of conjunctive rules by the *sequential covering* learning strategy, which sequentially learns a single conjunctive rule, removes the examples covered by the rule, then iterates the process until all examples of class C is covered or until no rule can be generated. The learning of a single conjunctive rule begins by considering the most general rule precondition, then greedily searching for an attribute-value pair that is most relevant to class C according to the following attribute-value pair evaluation function: $SIG_C(av) = P(av)(P(C|av) - P(C))$, where av is an attribute-value pair and P denotes probability. The selected attribute-value pair is then added to the rule precondition as a conjunct. The process is repeated by greedily adding a second attribute-value pair, and so on, until the hypothesis reaches an acceptable level of performance. In ELEM2, the acceptable level is based on the consistency of the rule: it forms a rule that is as consistent with the training data as possible. Since this "consistent" rule may overfit the data, ELEM2 then "post-prunes" the rule after the initial search for this rule is complete.

To post-prune a rule, ELEM2 computes a rule quality value according to one of the 11 statistical or empirical formulas. The formulas include *a weighted sum of rule consistency and coverage (WS), a product of rule consistency and coverage (Prod), the χ^2 statistic (Chi), the G2 likelihood ratio statistic (G2), a measure of rule logical sufficiency (LS), a measure of discrimination between positive and negative examples (MD), information score (IS), Cohen's formula (Cohen), Coleman's formula (Coleman), the C1 and C2 formulas*. These formulas are described in [3,4]. In post-pruning, ELEM2 checks each attribute-value pair in the rule in the reverse order in which they were selected to determine if removal of the attribute-value pair will decrease the rule quality value. If not, the attribute-value pair is removed and the procedure checks all the other pairs in the same order again using the new rule quality value resulting from the removal of that attribute-value pair to discover whether another attribute-value pair can be removed. This procedure continues until no pair can be removed.

4 The Classification Method

The classification procedure in ELEM2 considers three possible cases when a new example matches a set of rules. (1)*Single match.* The new example satisfies one or more rules of the same class. In this case, the example is classified to the class indicated by the rule(s). (2)*Multiple match.* The new example satisfies more than one rule that indicates different classes. In this case, ELEM2 activates a conflict resolution scheme for the best decision. The conflict resolution scheme computes a decision score for each of the matched classes as follows: $DS(C) = \sum_{i=1}^{k} Q(r_i)$, where r_i is a matched rule that indicates C, k is the number of this kind of rules, and $Q(r_i)$ is the rule quality of r_i. The new example is then classified into the class with the highest decision score. (3)*No match.* The new example e is not covered by any rule. Partial matching is considered where some attribute-value pairs of a rule match the values of corresponding attributes in e. If the partially-matched rules do not agree on the classes, a partial matching score between e and a partially-matched rule r_i with n attribute-value pairs, m of which match the corresponding attributes of e, is computed as $PMS(r_i) = \frac{m}{n} \times Q(r_i)$. A decision score for a class C is computed as $DS(C) = \sum_{i=0}^{k} PMS(r_i)$, where k is the number of partially-matched rules indicating class C. In decision making, e is classified into the class with the highest decision score.

5 Ranking the Test Examples

The classification procedure of ELEM2 produces a class label for each test example. To meet the requirement of our particular application, we design another prediction procedure which outputs a numerical score for each test example. The score is used to compare examples as to whether an example more likely belongs to a class than another example. Intuitively, we could use the decision score computed in the classification procedure to rank the examples. However, that decision score was designed to distinguish between classes for a given example. It consists of either *full*-matching scores (when the example fully matches a rule) or *partial*-matching scores (when no rule is fully matched with the example, but partial matching exists). It is possible that an example that only partially matches some rules of class C obtains a higher decision score than an example that fully matches one rule of C, even though the fully matched example is more likely to belong to C than the partially matched example.

In order to rank examples according to their likelihood of belonging to a class we need to design a criterion that can distinguish between examples given the class. To do so, we simply adjust the calculation of the decision score in the classification procedure to consider both kinds of matches (full and partial matches) in calculating a score for an example. The score is called the *ranking score* of an example with respect to a class. For class C and example e, we first compute a *matching score* between e and a rule r of C using $MS(e, r) = \frac{m}{n} \times Q(r)$, where n is the number of attribute-value pairs that r contains and m is the

number of attribute-value pairs in r that are matched with e. Note that this calculation covers a full match when $m = n$, a partial match when $< m < n$, and no match when $m = 0$. The ranking score of e with respect to C is defined as $RS(e, C) = \sum_{i=0}^{k} MS(e, r_i)$, where r_i is a rule of C and k is the number of rules of C.

The ranking algorithm of ELEM2 ranks the test examples according to both the predicted class label (produced by ELEM2's classification program) for the example and the ranking score of that example with respect to a specified class C, e.g., the minority class for an imbalanced data set. It places test examples that are classified into the specified class C in front of other test examples and ranks the examples in each group in decreasing order of the ranking score with respect to C.

6 Experiments with the Pharmaceutical Data Set

6.1 Comparison on Rule Quality Formulas

Our first objective is to determine how each of the rule quality formulas incorporated in ELEM2 reacts to the imbalance in our data set. To achieve this goal, we run ELEM2 with different rule quality formulas on our training data set. For each formula, a set of rules is generated. We then test these rules by running the classification program of ELEM2 to classify the examples in the test set. This program generates a discrete output for each test example, which is the predicted class label for that example. The performance of this classifier is measured by *precision* and *recall* (defined in Section 2.2) on the smaller class that corresponds to the active compounds. We also combine precision and recall by way of a geometric mean (g-mean) defined as $\sqrt{precision * recall}$. Figure 1 shows the precision, recall and g-mean of ELEM2's classification program using different rule quality formulas. Generally, formulas that produce higher recalls give lower precisions and formulas that give lower recalls produce higher precisions. In terms of g-mean, the G2 (*the G2 likelihood ratio statistic*) formula produces the best result, while the WS (a weighted sum of rule consistency and rule coverage) and Prod (a product of rule consistency and rule coverage) formulas have the worst performance.

We then run the ranking program of ELEM2 to rank the test examples according to the ranking score defined in Section 5. The performance of this program is measured by recall-level precisions, case-level precisions and an average precision.[1] *Recall-level precisions* are the precisions at a list of recall cutoff

[1] These measures are used in the TREC competitions of the information retrieval community [8]. We adopt these measures for use in our application because the requirement for our application (presenting predicted active compounds in an order in which the most probably active compounds are ranked first) is similar to the requirement in information retrieval, which ranks the retrieved documents according to the degree of relevance.

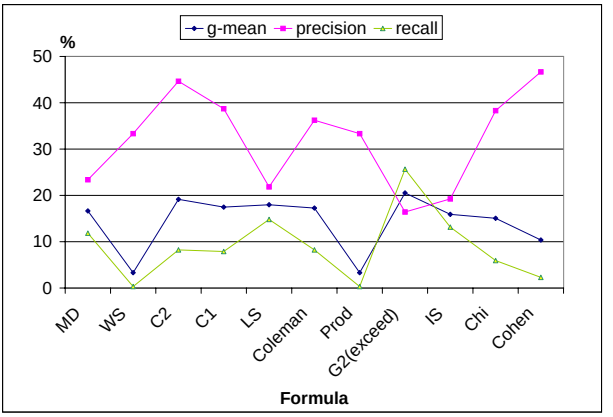


Fig. 1. Classification Performance of the Formulas

values. The recall cutoff values used are 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9 and 1. A graph on recall-level precisions depicts tradeoffs between precision and recall. *Case-level precisions* are the precisions at a list of *case* cutoff values. A case cutoff value is a number of cases being "retrieved". A precision at a case cutoff value n is the precision of the first n examples in the ranked list of test examples. The case cutoff values we used are 5, 10, 20, 30, 50, 100, 200, 400, 800, and 1000. Compared to recall-level precisions, case-level precisions give a better picture on the precisions at the top ranked cases. *Average precision* is the average of the precision values at the points where active compounds were correctly recognized in the run.

Figure 2 illustrates recall-level precisions and case-level precisions of the results generated by the ranking program using different formulas. In the figure, we only show the results for 7 formulas; the curves for our remaining 4 formulas (whose performance ranked medium) were deleted for graph clarity. The average precisions from each of the 11 formulas are shown in Figure 3. From recall-precision curves, we observe that formula G2 takes the lead generally, especially in the small to middle recall cutoff region. However, at the recall cutoff value of 0.1, formula LS (*measure of logical sufficiency*) takes the lead, followed by formula MD (*measure of discrimination*). The right graph of Figure 2 presents a clearer picture on the top ranked cases, which shows that LS is the "winner" for the top 50 cases and the χ^2 statistic (Chi) also performs well within this top region. In terms of average precision, Figure 3 shows that G2 takes the lead, followed by LS and then MD.

We also evaluate the result of each run using the ROC convex hull method proposed by Provost and Fawcett [13]. A ROC curve shows how the percentage of

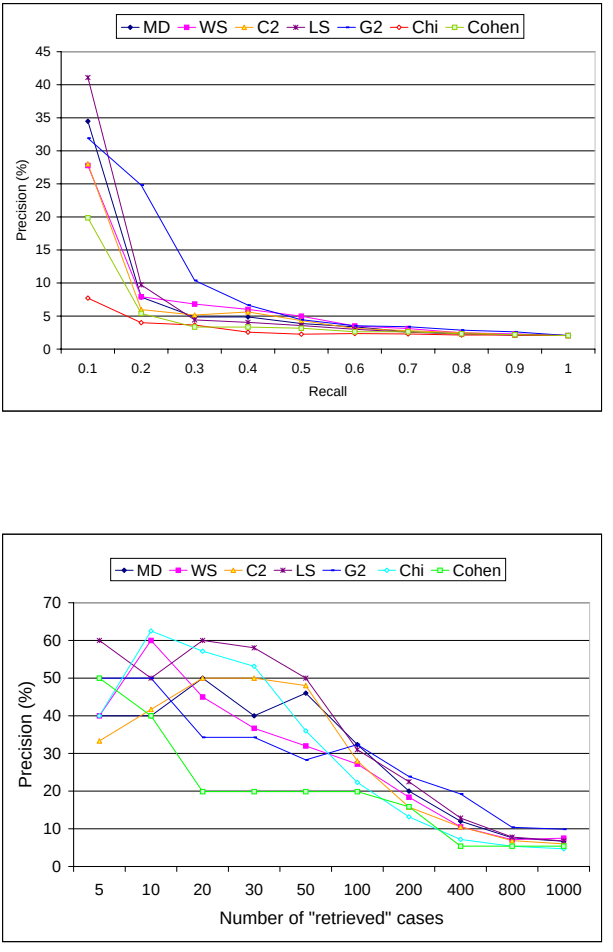


Fig. 2. Recall-level Precisions (top) and Case-level Precisions (bot)

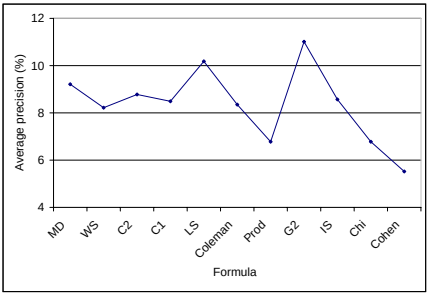


Fig. 3. Average Precisions

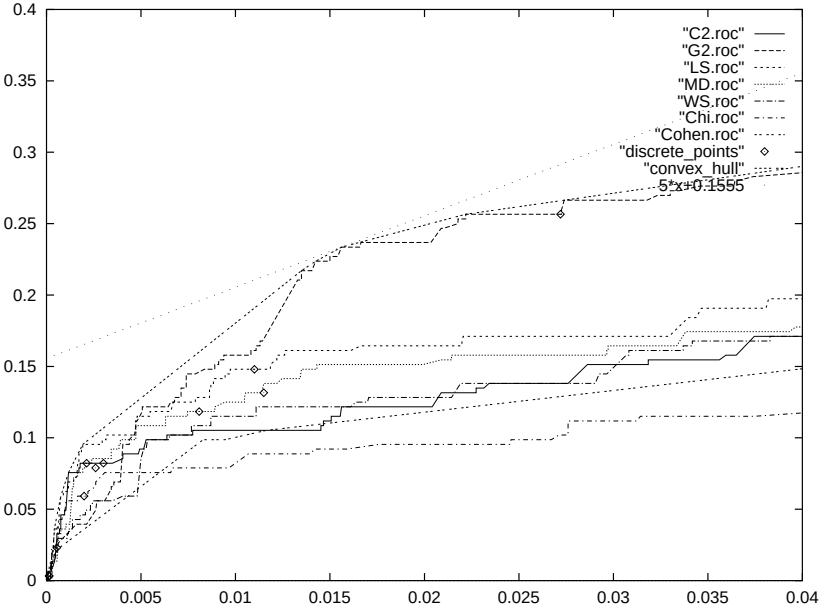


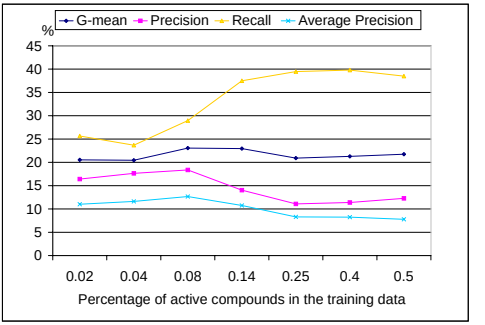
Fig. 4. ROC curves and the ROC convex hull from 7 Formulas

correctly recognized active compounds (recall or "true positive rate") depends on the "false positive rate", i.e., the percentage of the incorrectly classified inactive compounds. ROC curves illustrate tradeoffs between recall and false alarm rate for continuous output classifiers. The performance of a discrete classifier (which outputs only class labels) can be depicted as a point in the ROC space. A classifier is optimal for some conditions if and only if it lies on the northwest boundary (i.e., above the line y=x) of the convex hull of the set of points and curves in the ROC space. A nice feature of the ROC convex hull method is that the optimal classifier in terms of expected cost can be determined using *iso-performance lines* [13] in the ROC space according to the class distribution and the misclassification costs. Figure 4 depicts the ROC curves generated from the results of 7 formulas. Again the curves for the 4 other formulas were deleted for clarity. Figure 4 also shows the points corresponding to the performance of ELEM2's "discrete" classifier. Each point in the graph corresponds to a rule quality formula that was used to generate the classifier. The convex hull of these 7 curves and 11 points is shown in the picture. We notice that none of the discrete classifiers is optimal because their corresponding points are not on the convex hull curve. An optimal performance in terms of misclassification costs and class distribution can be obtained by setting a threshold for the continuous output value for the continuous "classifier" whose curve intersects with the convex hull. In our application, the cost of missing an active compound (cost of a false negative error) is potentially much higher than the cost of screening an inactive compound in the lab (cost of a false positive error). Suppose the false negative cost is 10 times higher than the false positive cost and the true distribution of the data is the same as the distribution in the training data. We can draw an *iso-performance line* (the straight line of $5x + 0.1555$) in the ROC space in Figure 4 based on the formula provided in [13], which intersects the convex hull. The intersection of this line and the convex hull is the point that determines the threshold value for the continuous-output classifier in order to obtain the optimal performance. These ROC curves also clearly show that G2 is the leading formula, followed by LS and then MD, which correlates with the conclusion obtained from average precisions in Figure 3.

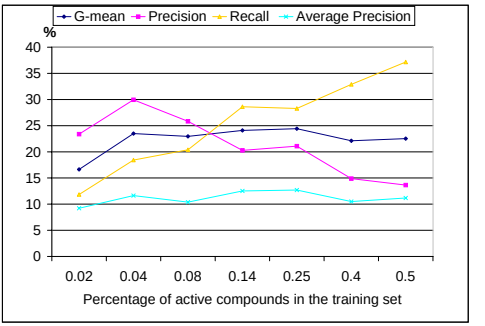
6.2 Balancing the Data

We would like to discover whether decreasing the imbalance in the training data set would improve the predictive performance. For this purpose, we created 6 additional training sets by duplicating the examples of active compounds to increase the prevalence of active compounds in the training data. Distributions of active compounds in these 6 training sets are 4%, 8%, 14%, 25%, 40% and 50%, respectively.

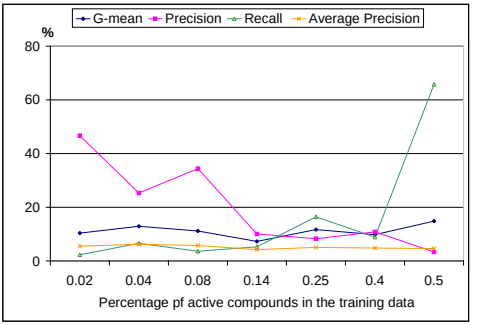
We picked three formulas (G2, MD, Cohen) ranging from good to poor based on the above results for use in this experiment. Figure 5, illustrates the results of increasing the minority class prevalence in terms of g-mean, precision, recall and average precision for the three formulas, respectively. All the three graphs



a



b



c

Fig. 5. Results of Increasing Prevalence for the G2 (**a**), MD (**b**) and Cohen's (**c**) Formulas

indicate that, generally, as the percentage of the active compounds in the training set increases, *recall* increases, but *precision* decreases. As a result, *g-mean* does not have significant changes. Also, *average precision* (for continuous output classifiers) does not change significantly either.

7 Adjusting the Learning Algorithm

Finally, we would like to determine whether adjusting the learning algorithm for an imbalanced data set would improve predictive performance. By analyzing the rules generated from each rule quality formula, we found that some formulas lead to generation of very few rules for the majority class. This is due to the fact that, when post-pruning a rule, removing an attribute-value pair from a rule for the majority class can greatly increase the coverage of the rule. In this case, for some rule quality measures in which the rule coverage makes a positive contribution, the value of rule quality is mostly likely to increase when removing an attribute-value pair, which results in general rules that cover a large number of cases of both the majority class and the minority class. This kind of rule does not describe well the instances in the majority class and has limited power in discriminating between the two classes. Therefore, we adjust the learning algorithm to only post-prune the rules generated for the minority class when the data set is extremely imbalanced. This adjustment is based on the assumption that we have enough training cases for the majority class and there is no noise in the training set for this class. We still post-prune the rules for the minority class because the training examples for the minority class is relatively rare and we do not want the rules to overfit the minority class examples.

We use five rule quality formulas that led to generation of a relatively small number of rules for the majority class, based on the above experiments, to test our strategy for adjusting the learning algorithm. The left graph of Figure 6 compares, in terms of g-mean and average precision, the results for pruning only minority class rules to the results for pruning rules for both classes. The results show that this adjustment greatly improves the predictive performance of these formulas. The right graph of Figure 6 shows the improvement on the recall-level precisions for the χ^2 statistic formula.

8 Conclusions

We have compared a number of rule quality formulas on an extremely imbalanced data set for identifying active chemical compounds. The rule quality formulas are used in ELEM2's rule induction and classification processes. Among the 11 tested statistical and empirical formulas, the G2 likelihood ratio statistic outperforms others in terms of g-mean, average precision and recall-level precisions. The ROC analysis also shows that G2 gives the best results. Other formulas that perform relatively well on this data set include the measure of logical sufficiency (LS) and the measure of discrimination (MD). In evaluating these formulas, we

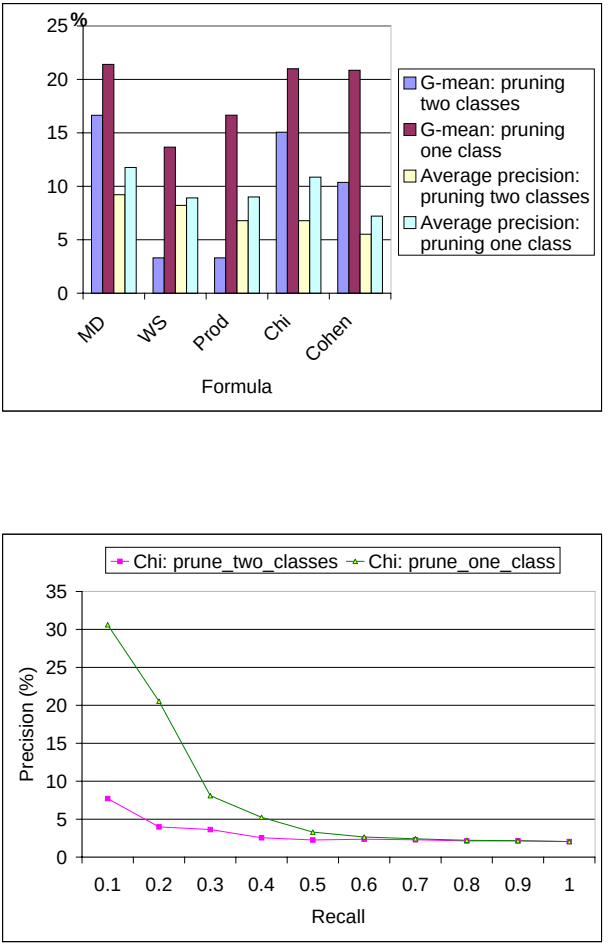


Fig. 6. Differences between pruning rules for 2 classes and pruning rules for the minority class

observed that ROC curves give a clearer picture than recall-precision curves on the overall performance of continuous output classifiers. Case-level precision curves produce a better picture on precisions at the top ranked cases. Another good measure of performance is average precision, which is good at ranking the evaluated continuous output classifiers. In our evaluation of rule quality formulas, the conclusion drawn from average precisions correlates well with the observation on ROC curves.

We also observed that increasing prevalence of the minority class in the training data does not improve predictive performance on our test data. This is because our learning algorithm (and many others for that matter) is based on statistical measures and assumes that the classifier will operate on data drawn from the same distribution as the training data. In terms of adjusting learning algorithm for extremely imbalanced data sets, we found that allowing rules for the majority class to "overfit" (without pruning) can improve predictive performance for rule quality formulas in which coverage of a rule makes a positive contribution to the rule quality value. Our future work includes evaluating a variety of statistical and machine learning methods on this imbalanced data set.

Acknowledgment. The authors are members of the Institute for Robotics and Intelligent Systems (IRIS) and wish to acknowledge the support of the Networks of Centres of Excellence of the Government of Canada, the Natural Sciences and Engineering Research Council, and the participation of PRECARN Associates Inc. We would like to thank Ms. Yuanyuan Wang and Professor Will Welch of the Statistics Department at University of Waterloo for passing the data set to us. We would also like to thank Stan Young and Ray Lam of Glaxo Wellcome Inc. for providing us with information on the data set used in the paper.

References

1. Aha, D. and Kibler, D. 1987. "Learning Representative Exemplars of Concepts: An Initial Case Study." *Proceedings of the Fourth International Conference on Machine Learning*, Irvine, CA.
2. An, A. and Cercone, N. 1998. "ELEM2: A Learning System for More Accurate Classifications." *Proceedings of the 12th Biennial Conference of the Canadian Society for Computational Studies of Intelligence, AI'98* (*Lecture Notes in Artificial Intelligence 1418*), Vancouver, Canada.
3. An, A. and Cercone, N. 2000. "Rule Quality Measures Improve the Accuracy of Rule Induction: An Experimental Approach", *Proceedings of the 12th International Symposium on Methodologies for Intelligent Systems*, Charlotte, NC. pp.119-129.
4. Bruha, I. 1996. "Quality of Decision Rules: Definitions and Classification Schemes for Multiple Rules", in Nakhaeizadeh, G. and Taylor, C. C. (eds.): Machine Learning and Statistics, The Interface. Jone Wiley & Sons Inc.
5. Cardie, C and Howe, N. 1997. "Improving Minority Class Prediction Using Case-Specific Feature Weights", *Proceedings of the Fourteenth International Confernece on Machine Learning*, Morgan Kaufmann. pp.57-65.
6. DeRouin, E., Brown, J., Beck, H., Fausett, L. and Schneider, M. 1991. "Neural Network Training on Unequally Represented Classes", In Dagli, C.H., Kumara, S.R.T. and Shin, Y.C. (eds.), *Intelligent Engineering Systems Through Artifical Neural Networks*, ASME Press. pp.135-145.
7. Duda, R., Gaschnig, J. and Hart, P. 1979. "Model Design in the Prospector Consultant System for Mineral Exploration". In D. Michie (ed.), *Expert Systems in the Micro-electronic Age*. Edinburgh University Press, Edinburgh, UK.
8. Harman, D.K. (ed.) 1995. *Overview of the Third Text REtrieval Conference (TREC-3)*, NIST Special Publication. pp. A5-A13.

9. Kubat, M. and Matwin, S. 1997. "Addressing the Curse of Imbalanced Training Sets: One-Sided Sampling". *Proceedings of the Fourteenth International Conference on Machine Learning*, Morgan Kaufmann. pp.179-186.
10. Kubat, M., Holte, R. and Matwin, S. 1997. "Learning when Negative Examples Abound," *Proceedings of ECML-97*, Springer. pp.146-153.
11. Kubat, M., Holte, R. and Matwin, S. 1998. "Machine Learning for the Detection of Oil Spills in Satellite Radar Images", *Machine Learning*, 30, pp.195-215.
12. Provost, F. 2000 "Machine Learning from Imbalanced Data Sets", *Invited paper for the AAAI'2000 Workshop on Imbalanced Data Sets*, *http://www.stern.nyu.edu/˜fprovost/home.html#Publications* .
13. Provost, F. and Fawcett, T. 2000. "Robust Classification for Imprecise Environments", to appear in *Machine Learning*.