Eleni Stroulia
Stan Matwin (Eds.)

LNAI 2056

Advances in Artificial Intelligence

14th Biennial Conference of the Canadian Society for Computational Studies of Intelligence, AI 2001
Ottawa, Canada, June 2001, Proceedings

Springer

Eleni Stroulia Stan Matwin (Eds.)

Advances in Artificial Intelligence

14th Biennial Conference of the Canadian Society
for Computational Studies of Intelligence, AI 2001
Ottawa, Canada, June 7-9, 2001
Proceedings

Springer

Series Editors

Jaime G. Carbonell, Carnegie Mellon University, Pittsburgh, PA, USA
Jörg Siekmann, University of Saarland, Saabrücken, Germany

Volume Editors

Eleni Stroulia
University of Alberta, Department of Computer Science
Edmonton, AB, Canada T6G 2E8
E-mail: stroulia@cs.ualberta.ca

Stan Matwin
University of Ottawa, School of Information Technology and Engineering
Ottawa, ON, Canada K1N 6N5
E-mail: stan@site.uottawa.ca

Cataloging-in-Publication Data applied for

Die Deutsche Bibliothek - CIP-Einheitsaufnahme

Advances in artificial intelligence : proceedings / AI 2001, Ottawa,
Canada, June 7 - 9, 2001. Eleni Stroulia ; Stan Matwin (ed.). – Berlin ;
Heidelberg ; New York ; Barcelona ; Hong Kong ; London ; Milan ;
Paris ; Singapore ; Tokyo : Springer, 2001
(... biennial conference of the Canadian Society for Computational
Studies of Intelligence ; 14)
(Lecture notes in computer science ; Vol. 2056 : Lecture notes in
artificial intelligence)
ISBN 3-540-42144-0

CR Subject Classification (1998): I.2

ISBN 3-540-42144-0 Springer-Verlag Berlin Heidelberg New York

Springer-Verlag Berlin Heidelberg New York
a member of BertelsmannSpringer Science+Business Media GmbH

http://www.springer.de

Typesetting: Camera-ready by author, data conversion by PTP-Berlin, Stefan Sossna
Printed on acid-free paper SPIN: 10781551 06/3142 5 4 3 2 1 0

Lecture Notes in Artificial Intelligence 2056

Subseries of Lecture Notes in Computer Science
Edited by J. G. Carbonell and J. Siekmann

Lecture Notes in Computer Science

Edited by G. Goos, J. Hartmanis and J. van Leeuwen

Springer
Berlin
Heidelberg
New York
Barcelona
Hong Kong
London
Milan
Paris
Singapore
Tokyo

Preface

AI 2001 is the 14th in the series of Artificial Intelligence conferences sponsored by the Canadian Society for Computational Studies of Intelligence/Société canadienne pour l'étude de l'intelligence par ordinateur. As was the case last year too, the conference is being held in conjunction with the annual conferences of two other Canadian societies, Graphics Interface (GI 2001) and Vision Interface (VI 2001). We believe that the overall experience will be enriched by this conjunction of conferences.

This year is the "silver anniversary" of the conference: the first Canadian AI conference was held in 1976 at UBC. During its lifetime, it has attracted Canadian and international papers of high quality from a variety of AI research areas. All papers submitted to the conference received at least three independent reviews. Approximately one third were accepted for plenary presentation at the conference. The best paper of the conference will be invited to appear in Computational Intelligence.

This year, we have some innovations in the format of the conference. In addition to the plenary presentations of the 24 accepted papers, organized in topical sessions, we have a session devoted to short presentations of the accepted posters, and a graduate symposium session. With this format, we hope to increase the level of interaction and to make the experience even more interesting and enjoyable to all the participants. The graduate symposium is sponsored by AAAI, who provided funds to partially cover the expenses of the participating students.

Many people contributed to the success of this conference. The members of the program committee coordinated the refereeing of all submitted papers. They also made several recommendations that contributed to other aspects of the program. The referees provided reviews of the submitted technical papers; their efforts were irreplaceable in ensuring the quality of the accepted papers. Our thanks also go to Howard Hamilton and Bob Mercer for their invaluable help in organizing the conference. We also acknowledge the help we received from Alfred Hofmann and others at Springer-Verlag.

Lastly, we are pleased to thank all participants. You are the ones who make all this effort worthwhile!

June 2001 Eleni Stroulia, Stan Matwin

Organization

AI 2001 is organized by the Canadian Society for Computational Studies of Intelligence (Société canadienne pour l'étude de l'intelligence par ordinateur).

Program Committee

Program Co-chairs:	Stan Matwin (University of Ottawa) Eleni Stroulia (University of Alberta)
Committee Members:	Irene Abi-Zeid (Defence Research Establishment Valcartier) Fahiem Bacchus (University of Toronto) Ken Barker (University of Texas at Austin) Sabine Bergler (Concordia University) Nick Cercone (University of Waterloo) Michael Cox (Wright State University) Chrysanne DiMarco (University of Waterloo) Toby Donaldson (TechBC) Renee Elio (University of Alberta) Ali Ghorbani (University of New Brunswick) Jim Greer (University of Saskatchewan) Howard Hamilton (University of Regina) Graeme Hirst (University of Toronto) Robert Holte (Univesity of Ottawa) Nathalie Japkowicz (Univesity of Ottawa) Guy LaPalme (Université de Montréal) Dekang Lin (University of Alberta) André Trudel (Acadia University) Joel Martin (National Research Council) Gord McCalla (University of Saskatchewan) Robert Mercer (University of Western Ontario) John Mylopoulos (University of Toronto) Witold Pedrycz (University of Alberta) Fred Popowich (Simon Fraser University) Yang Qiang (Simon Fraser University) Bruce Spencer (University of New Brunswick) Ahmed Tawfik (University of Windsor) Afzal Upal (Daltech / Dalhousie University) Peter van Beek (University of Waterloo) Kay Wiese (TechBC)

Referees

Irene Abi-Zeid
Fahiem Bacchus
Ken Barker
Sabine Bergler
Nick Cercone
Michael Cox
Toby Donaldson
Renee Elio
Dan Fass
Ali Ghorbani
Paolo Giorgini
Jim Greer
Howard Hamilton
Graeme Hirst
Robert Holte
Nathalie Japkowicz
Guy LaPalme
Dekang Lin
André Trudel
Joel Martin
Gord McCalla
Tim Menzies
Robert Mercer
John Mylopoulos
Witold Pedrycz
Fred Popowich
Bruce Spencer
Ahmed Tawfik
Afzal Upal
Peter van Beek
Kay Wiese

Sponsoring Institutions

AAAI, American Association for Artificial Intelligence

Table of Contents

Posters

Graduate Symposium Contributions

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