ARTIFICIAL INTELLIGENCE APPLICATIONS AND INNOVATIONS

IFIP – The International Federation for Information Processing

IFIP was founded in 1960 under the auspices of UNESCO, following the First World Computer Congress held in Paris the previous year. An umbrella organization for societies working in information processing, IFIP's aim is two-fold: to support information processing within its member countries and to encourage technology transfer to developing nations. As its mission statement clearly states,

> *IFIP's mission is to be the leading, truly international, apolitical organization which encourages and assists in the development, exploitation and application of information technology for the benefit of all people.*

IFIP is a non-profitmaking organization, run almost solely by 2500 volunteers. It operates through a number of technical committees, which organize events and publications. IFIP's events range from an international congress to local seminars, but the most important are:

- The IFIP World Computer Congress, held every second year;
- Open conferences;
- Working conferences.

The flagship event is the IFIP World Computer Congress, at which both invited and contributed papers are presented. Contributed papers are rigorously refereed and the rejection rate is high.

As with the Congress, participation in the open conferences is open to all and papers may be invited or submitted. Again, submitted papers are stringently refereed.

The working conferences are structured differently. They are usually run by a working group and attendance is small and by invitation only. Their purpose is to create an atmosphere conducive to innovation and development. Refereeing is less rigorous and papers are subjected to extensive group discussion.

Publications arising from IFIP events vary. The papers presented at the IFIP World Computer Congress and at open conferences are published as conference proceedings, while the results of the working conferences are often published as collections of selected and edited papers.

Any national society whose primary activity is in information may apply to become a full member of IFIP, although full membership is restricted to one society per country. Full members are entitled to vote at the annual General Assembly, National societies preferring a less committed involvement may apply for associate or corresponding membership. Associate members enjoy the same benefits as full members, but without voting rights. Corresponding members are not represented in IFIP bodies. Affiliated membership is open to non-national societies, and individual and honorary membership schemes are also offered.

ARTIFICIAL INTELLIGENCE APPLICATIONS AND INNOVATIONS

3rd IFIP Conference on Artificial Intelligence Applications and Innovations (AIAI) 2006, June 7-9, 2006, Athens, Greece

Edited by

Ilias Maglogiannis
University of the Aegean, Greece

Kostas Karpouzis
ICCS/NTUA, Greece

Max Bramer
University of Plymouth, United Kingdom

Springer

Library of Congress Control Number: 2006925099

Artificial Intelligence Applications and Innovations
Edited by I. Maglogiannis, K. Karpouzis, and M. Bramer

p. cm. (IFIP International Federation for Information Processing, a Springer Series in Computer Science)

ISSN: 1571-5736 / 1861-2288 (Internet)
ISBN: 10: 0-387-34223-0
ISBN: 13: 9780-387-34223-0
eISBN: 10: 0-387-34224-9
Printed on acid-free paper

Printed in the United States of America.

9 8 7 6 5 4 3 2 1
springer.com

Contents

Foreword

It is our pleasure to welcome you to the Proceedings of AIAI 2006, the 3rd IFIP Conference on Artificial Intelligence Applications & Innovations being held from 7^{th} till 9^{th} of June, in Athens, Greece. Artificial Intelligence applications build on a rich and proven theoretical background to provide solutions to a wide range of real life problems. The ever expanding abundance of information and computing power enables researchers and users to tackle highly interesting issues for the first time, such as applications providing personalized access and interactivity to multimodal information based on user preferences and semantic concepts or human-machine interface systems utilizing information on the affective state of the user. The purpose of the 3rd IFIP Conference on Artificial Intelligence Applications and Innovations (AIAI) is to bring together researchers, engineers and practitioners interested in the technical advances and business and industrial applications of intelligent systems. AIAI 2006 is focused on providing insights on how AI can be implemented in real world applications.

The response to the 'Call for Papers' was overwhelming, attracting submissions from 23 countries. The task of the Technical Program Committee was very challenging putting together a Program containing 87 high quality contributions. The collection of papers included in the proceedings offer stimulating insights into emerging applications of AI and describe advanced prototypes, systems, tools and techniques. AIAI Proceedings will interest not only academics and researchers, but IT professionals and consultants by examining technologies and applications of demonstrable value.

Eight (8) Special Sessions dedicated to specific AI applications are affiliated within the AIAI 2006 conference:

- Adaptive Learning Systems Engineering (organized by Symeon Retalis, Andreas Papasalouros and Kostas Siassiakos)
- Advances in Artificial Intelligence for Integrated Surveillance and Monitoring Systems (organized by Dimitris Vergados and Christos Anagnostopoulos)

- Computational Intelligence in Software Engineering (organized by Andreou Andreas and Efstratios Georgopoulos)
- Computational Intelligence in Medical Imaging (organized by Efthyvoulos Kyriacou and Ilias Maglogiannis)
- Digital Rights Management Techniques and Interoperability of Protection Tools (organized by Sofia Tsekeridou)
- Emerging Multimodal Interfaces (organized by John Soldatos, Dimitris Tzovaras and Kostas Karpouzis)
- Intelligent Analysis of Medical and Biological Data (organized by Vasileios Megalooikonomou and Despina Kontos)
- Semantics in Multimedia Analysis and Natural Language Processing (organized by Anastasios Delopoulos, Vangelis Karkaletsis, George Paliouras and Manolis Wallace)

The wide range of topics and high level of contributions will surely guarantee a very successful conference. We express our special thanks to all who have contributed to the organization and scientific contents of this conference, first to the authors of the papers, then to the special session organizers and finally to the reviewers and members of the Program and Organization Committees.

June, 2006

AIAI 2006 Conference Chairs:

Ilias Maglogiannis, University of Aegean, Greece,

Kostas Karpouzis, ICCS/NTUA, Greece,

Max Bramer, University of Portsmouth, UK

Acknowledgments

AIAI 2006 conference is co-organized by the Department of Information and Communications Systems Engineering of the University of Aegean and Athens Information Technology (AIT). AIAI 2006 is the official conference of WG12.5 "Artificial Intelligence Applications" working group of IFIP TC12 the International Federation for Information Processing Technical Committee on Artificial Intelligence (AI).

Conference General Chairs

Ilias Maglogiannis, University of Aegean, Greece
Kostas Karpouzis, NTUA, Greece
Max Bramer, University of Portsmouth, UK

Organizing Committee Chairs

Manolis Wallace, University of Indianapolis, Athens Campus, Greece
Lazaros Polymenakos, Athens Information Technology, Greece

Publicity Chair

Constantine D. Spyropoulos, NCSR Demokritos, Greece

Technical Program Committee

Program Committee Chairs

Ilias Maglogiannis, University of Aegean, Greece
John Soldatos, AIT, Greece
John Debenham, University of Technology, Australia

Program Committee Members

Olusola Abidogun, University of the Western Cape, South Africa
Christos Anagnostopoulos, University of Aegean, Greece
Ioannis Anagnostopoulos, University of Aegean, Greece
Andreou Andreas, University of Cyprus, Cyprus
Grigorios Beligiannis, University of Patras, Greece
Zdzislaw Bubnicki, Wroclaw University of Technology, Poland
Luigia Carlucci Aiello, University di Roma La Sapienza, Italy
Weiqin Chen, University of Bergen, Norway
Anastasios Delopoulos, Aristotle University of Thessaloniki, Greece
Yves Demazeau, CNRS/IMAG Institute, France
Christos Douligeris, University of Piraeus, Greece
Efstratios Georgopoulos, Technological Educational Institute of Kalamata, Greece
Starhes Hadjieftymiades, University of Athens, Greece
Timo Honkela, Helsinki University of Technology, Finland
Achilles Kameas, Hellenic Open University, Greece
Vangelis Karkaletsis, NCSR Demokritos, Greece
Stefanos Kollias, National Technical University of Athens, Greece
Despina Kontos, Temple University, USA
George Kormentzas, University of Aegean, Greece
Dimitris Kosmopoulos, NCSR Demokritos, Greece
Efthyvoulos Kyriacou, University of Cyprus, Cyprus
Daoliang Li, China Agricultural University, China
Filia Makedon, Department of Computer Science, Dartmouth College, USA
Yannis Manolopoulos, Aristotle University Thessaloniki, Greece
Kinshuk, Massey University, New Zealand
Vasileios Megalooikonomou, Temple University, USA
Eunika Mercier-Laurent, KIM, France
Alex Nanopoulos, Aristotle University Thessaloniki, Greece
Nikos Nikolaidis, Aristotle University of Thessaloniki, Greece

Daniel O'Leary, University of Southern California, USA
Geroge Papakonstantinou, National Technical University of Athens, Greece
Andreas Papasalouros, National Technical University of Athens, Greece
Constantinos Pattichis, University of Cyprus, Cyprus
Aristodemos Pnevmatikakis, Athens Information Technology, Greece
Lazaros Polymenakos, Athens Information Technology, Greece
Symeon Retalis, University of Piraeus, Greece
Angelos Rouskas, University of Aegean, Greece
Pierre-Yves Schobbens, University of Namur, Belgium
Kostas Siassiakos, University of Piraeus, Greece
Harry Skiannis, NCSR Demokritos, Greece
Derek Sleeman, University of Aberdeen, UK
Costas Spyropoulos, NCSR Demokritos, Greece
Andreas Stafylopatis, NTUA, Greece
Dimitris Stamoulis, University of Athens, Greece
Olga Stepankova, Czech Technical University in Prague, Czech Republic
Kostas Stergiou, University of Aegean, Greece
Vagan Terziyan, MIT Department, University of Jyvaskyla, Finland
Panayiotis Tsanakas, National Technical University of Athens, Greece
Sophia Tsekeridou, University of Thrace, Greece
Dimitris Vergados, University of Aegean, Greece
George Vouros, University of Aegean, Greece
Manolis Wallace, University of Indianapolis, Athens Campus, Greece
Michalis Xenos, Hellenic Open University, Greece
Ioannis Zaharakis, Computer Technology Institute (CTI), Greece

Local Ordinal Classification

Sotiris B. Kotsiantis
Educational Software Development Laboratory
Department of Mathematics
University of Patras, Greece
sotos@math.upatras.gr

Abstract. Given ordered classes, one is not only concerned to maximize the classification accuracy, but also to minimize the distances between the actual and the predicted classes. This paper offers an organized study on the various methodologies that have tried to handle this problem and presents an experimental study of these methodologies with the proposed local ordinal technique, which locally converts the original ordinal class problem into a set of binary class problems that encode the ordering of the original classes. The paper concludes that the proposed technique can be a more robust solution to the problem because it minimizes the distances between the actual and the predicted classes as well as improves the classification accuracy.

1 Introduction

Ordinal classification can be viewed as a bridging problem between the two standard machine-learning tasks of classification and regression. In ordinal classification, the target values are in a finite set (like in classification) but there is an ordering among the elements (like in regression, but unlike classification).

Although Machine Learning (ML) algorithms for ordinal classification are rare, there are a number of statistical approaches to this problem. However, they all rely on specific distributional assumptions for modeling the class variable and also assume a stochastic ordering of the input space [9]. The ML community has mainly addressed the issue of ordinal classification in two ways. One is to apply classification algorithms by discarding the ordering information in the class attribute [2]. The other is to apply regression algorithms by transforming class values to real numbers [9]. This paper proposes a local ordinal technique that locally converts the original ordinal problem into a set of binary problems encoding the ordering of the original classes. Experimental results show that this technique minimizes the distances between the actual and the predicted class, as well as improves the prediction accuracy.

Please use the following format when citing this chapter:
Kotsiantis, Sotiris, 2006, in IFIP International Federation for Information Processing, Volume 204, Artificial Intelligence Applications and Innovations, eds. Maglogiannis, I., Karpouzis, K., Bramer, M., (Boston: Springer), pp. 1–8

This paper is organized as follows: the next section discusses the different techniques that have been presented for handling ordinal classification problems. In section 3, we describe the proposed technique. In Section 4, we present the experimental results of our methodology using different distribution algorithms and compare these results with those of other approaches. In the final section of the paper we discuss further work and some conclusions.

2 Techniques for Dealing with Ordinal Problems

Classification algorithms can be applied to ordinal prediction problems by discarding the ordering information in the class attribute. However, some information that could improve the performance of a classifier is lost when this is done.

The use of regression algorithms to solve ordinal problems has been examined in [9]. In this case each class needs to be mapped to a numeric value. However, if the class attribute represents a truly ordinal quantity, which, by definition, cannot be represented as a number in a meaningful way, there is no upright way of devising an appropriate mapping and this procedure is ad hoc.

Another approach is to reduce the multi-class ordinal problem to a set of binary problems using the one-against-all approach [2]. In the one-against-all approach, a classifier is trained for each of the classes using as positive examples the training examples that belong to that class, and as negatives all the other training examples. The estimates given by each binary classifier are then coupled in order to obtain class probability membership estimates for the multi-class problem [2].

A more sophisticated approach that enables classification algorithms to make use of ordering information in ordinal class attributes is presented in [7]. Similarly with previous method, this method converts the original ordinal class problem into a set of binary class problems that encode the ordering of the original classes. However, to predict the class value of an unseen instance this algorithm needs to estimate the probabilities of the m original ordinal classes using $m - 1$ models. For example, for a three class ordinal problem, estimation of the probability for the first ordinal class value depends on a single classifier: Pr(*Target* < *first value*) as well as for the last ordinal class: Pr(*Target* > *second value*). Whereas, for class value in the middle of the range, the probability depends on a pair of classifiers and is given by

Pr(*Target* > *first value*) * (1 − Pr(*Target* > *second value*)).

3 Proposed Technique

The proposed technique is based on the previous referred sophisticated technique [7]; however, we do not apply this technique globally but locally. If all training instances are taken into account when classifying a new test case, the classifier works as a global method, while when the nearest training instances are taken into account, the classifier works as a local method, since only data local to the area around the testing instance contribute to the classification.

Generally, local methods have significant advantages when the probability measure defined on the space of symbolic features for each class is very complex, but can still be described by a collection of less complex local approximations [1]. The proposed algorithm builds the required number of classifiers for each point to be estimated, taking into account only a subset of the training points. This subset is chosen on the basis of the preferable distance metric between the testing point and the training point in the input space.

In other words, the proposed technique consists of the four steps in Fig. 1.

1. Determine a suitable distance metric.
2. Find the k nearest neighbors using the selected distance metric.
3. Estimate the probabilities of the m original ordinal classes with m − 1 models using as training instances these k instances
4. The estimates given by each binary classifier are then coupled in order to obtain class probability membership estimates

Fig. 1. Local Ordinal Technique

The proposed ensemble has some free parameters such as the distance metric. In our experiments, we used the most well known -Euclidean similarity function- as distance metric. We also used k=50 since about this size of instances is appropriate for a simple algorithm to built a precise model [6].

A key feature of our method is that it does not require any modification of the underlying learning algorithm; it is applicable as long as the classifier produces class probability estimates. In the following section, we empirically evaluate the performance of our approach with the other well known techniques.

4 Experiments

To test the hypothesis that the above method improves the generalization performance on ordinal prediction problems, we performed experiments on real-world ordinal datasets donated by Dr. Arie Ben David (http://www.cs.waikato.ac.nz/ml/weka/). We also used well-known datasets from many domains from the UCI repository [3]. However, the used UCI datasets represented numeric prediction problems and for this reason we converted the numeric target values into ordinal quantities using equal-size binning. This unsupervised discretization method divides the range of observed values into three equal size intervals. The resulting class values are ordered, representing variable-size intervals of the original numeric quantity. This method was chosen because of the lack of numerous benchmark datasets involving ordinal class values.

All accuracy estimates were obtained by averaging the results from 10 separate runs of stratified 10-fold cross-validation. It must be mentioned that we used the free available source code for most algorithms by the book [11]. In the following we present the empirical results obtained using Decision Stump (DS) [8], RepTree [11] and Naïve Bayes (NB) [5] algorithms as base learners. All of them produce class probability estimates.

Table 1 shows the results for the DS algorithm applied (a) without any modification of DS, (b) in conjunction with the ordinal classification method presented in Section 2 (Ordinal DS), (c) in conjunction with the multiclass classification method presented in Section 2 (Multiclass DS) and (d) using the proposed technique (Local Ordinal DS).

In Table 1, for each data set the algorithms are compared according to classification accuracy (the rate of correct predictions) and to mean absolute error:

$$\frac{|p_1 - a_1| + |p_2 - a_2| + \ldots + |p_n - a_n|}{n}$$

where p: predicted values and a: actual values. Moreover, in Table 1, we represent as "v" that the specific algorithm performed statistically better than the proposed method according to t-test with $p<0.05$. Throughout, we speak of two results for a dataset as being "significant different" if the difference is statistical significant at the 5% level according to the corrected resampled t-test [10], with each pair of data points consisting of the estimates obtained in one of the 100 folds for the two learning methods being compared. On the other hand, "*" indicates that proposed method performed statistically better than the specific algorithm according to t-test with $p<0.05$.

As one can observe from the aggregated results in Table 1, the proposed technique is more accurate than the remaining approaches from 2% to 5%. Moreover, it manages to minimize the distances between the actual and the predicted classes. The reduction of the mean absolute error is about 27% compared to the Ordinal DS and 30% compared to the simple DS, while it exceeds the 138% compared to the Multiclass DS. It must be also mentioned that the proposed method is statistically more accurate and has statistically less mean absolute error than the remaining methods in numerous datasets.

Similarly, Table 2 shows the results for the NB algorithm applied (a) without any modification of NB, (b) in conjunction with the ordinal classification method presented in Section 2 (Ordinal NB), (c) in conjunction with the multiclass classification method presented in Section 2 (Multiclass NB) and (d) using the proposed technique (Local Ordinal NB).

As one can see from the aggregated results in Table 2, the proposed technique is more accurate in classification accuracy than the remaining techniques from 2% to 5%. Furthermore, it minimizes the distances between the actual and the predicted classes. In detail, the reduction of the mean absolute error is about 25% compared to the Ordinal NB and 17% compared to simple NB, while it overcomes the 158% compared to Multiclass NB. It must be also stated that the proposed method is statistically more accurate and has statistically less mean absolute error than the remaining methods in a lot of datasets.

Similarly, Table 3 shows the results for the RepTree algorithm applied (a) without any modification of RepTree, (b) in conjunction with the ordinal classification method presented in Section 2 (Ordinal RepTree), (c) in conjunction with the multiclass classification method presented in Section 2 (Multiclass RepTree) and (d) using the proposed technique (Local Ordinal RepTree).

As one can notice from the aggregated results in Table 3, the proposed technique is more accurate in classification accuracy than the remaining techniques from 1% to 2%. What is more, it minimizes the distances between the actual and the predicted classes since the reduction of the mean absolute error is about 15% compared to the Ordinal RepTree and simple RepTree, while it overcomes the 138% compared to Multiclass RepTree. The proposed method is also statistically more accurate and has statistically less mean absolute error than the remaining methods in many datasets.

Table 1. Results for DS algorithm

Dataset		Local Ordinal DS	Multiclass DS	Ordinal DS	DS
auto93	accuracy	80.90	80.57	79.59	81.32
	MeanError	0.14	0.34*	0.18	0.18
autoHorse	accuracy	95.24	91.17	89.63*	91.17
	MeanError	0.04	0.30*	0.09*	0.09*
autoMpg	accuracy	79.67	79.76	78.01	79.61
	MeanError	0.14	0.35*	0.20*	0.21*
autoPrice	accuracy	88.11	89.80	89.80	86.05
	MeanError	0.09	0.31*	0.10	0.13*
bodyfat	accuracy	97.57	99.12	99.12	91.98*
	MeanError	0.02	0.29*	0.01	0.10*
cleveland	accuracy	70.32	71.63	71.14	71.93
	MeanError	0.21	0.37*	0.26*	0.26*
Cloud	accuracy	84.69	87.72	83.43	84.51
	MeanError	0.11	0.32*	0.13	0.14*
Cpu	accuracy	98.09	97.76	97.76	98.24
	MeanError	0.01	0.28*	0.02	0.02
Era	accuracy	25.69	22.08*	24.13	21.81*
	MeanError	0.18	0.20*	0.18*	0.19*
Esl	accuracy	65.53	44.48*	53.72*	43.03*
	MeanError	0.09	0.20*	0.13*	0.16*
fishcatch	accuracy	97.35	92.37*	92.37*	90.56*
	MeanError	0.03	0.30*	0.07*	0.10*
housing	accuracy	79.58	74.81	75.77	70.39*
	MeanError	0.15	0.36*	0.23*	0.28*
hungarian	accuracy	79.06	81.78	81.78	81.78
	MeanError	0.15	0.34*	0.20*	0.20*
Lev	accuracy	61.79	43.86*	49.03*	42.40*
	MeanError	0.20	0.31*	0.25*	0.26*
lowbwt	accuracy	57.25	61.80	61.90	61.90
	MeanError	0.30	0.39*	0.31	0.31
pharynx	accuracy	68.98	73.85	73.85	73.85
	MeanError	0.25	0.37*	0.25	0.25
servo	accuracy	89.72	83.36*	83.24*	83.36*
	MeanError	0.09	0.31*	0.13*	0.12*
Strike	accuracy	98.85	99.06	99.06	99.06
	MeanError	0.01	0.27*	0.01	0.01
swd	accuracy	56.11	51.38*	54.56	51.80*
	MeanError	0.26	0.36*	0.29*	0.30*

Veteran	accuracy	90.45	91.26	90.80	91.26
	MeanError	0.10	0.31*	0.11	0.11
AVERAGE	accuracy	78.25	75.88	76.43	74.80
	MeanError	0.13	0.31	0.16	0.17

Table 2. Results for NB algorithm

Dataset		Local Ordinal NB	Multiclass NB	Ordinal NB	NB
auto93	accuracy	84.36	76.28	74.01	76.18
	MeanError	0.10	0.33*	0.17*	0.16
autoHorse	accuracy	95.14	91.06	90.87	90.67*
	MeanError	0.03	0.29*	0.06*	0.06*
autoMpg	accuracy	82.56	80.65	70.11*	78.89
	MeanError	0.12	0.32*	0.20*	0.15*
autoPrice	accuracy	90.31	91.51	91.45	90.25
	MeanError	0.07	0.30*	0.06	0.07
bodyfat	accuracy	88.96	79.64*	77.22*	81.34*
	MeanError	0.08	0.32*	0.16*	0.13*
cleveland	accuracy	72.45	74.82	75.51	73.31
	MeanError	0.19	0.34*	0.18	0.19
Cloud	accuracy	90.30	91.70	92.04	89.95
	MeanError	0.07	0.30*	0.07	0.08
Cpu	accuracy	97.81	97.56	94.87	97.56
	MeanError	0.01	0.28*	0.04*	0.02
Era	accuracy	23.25	24.73	25.07	24.88
	MeanError	0.18	0.20*	0.18	0.18
Esl	accuracy	67.37	66.84	54.65*	67.52
	MeanError	0.09	0.19*	0.12*	0.10*
fishcatch	accuracy	97.42	89.92*	88.13*	90.10*
	MeanError	0.02	0.30*	0.08*	0.07*
housing	accuracy	81.44	74.76*	56.15*	73.14*
	MeanError	0.13	0.34*	0.29*	0.19*
hungarian	accuracy	81.17	83.95	83.95	83.95
	MeanError	0.13	0.31*	0.12v	0.12v
Lev	accuracy	59.95	56.24*	57.95	56.12*
	MeanError	0.20	0.31*	0.23*	0.23*
lowbwt	accuracy	60.10	58.79	58.52	59.53
	MeanError	0.29	0.39*	0.30	0.30
pharynx	accuracy	70.17	71.09	71.13	70.52
	MeanError	0.24	0.36*	0.25	0.25*
servo	accuracy	87.59	87.24	86.48	87.12
	MeanError	0.10	0.31*	0.12*	0.12*
Strike	accuracy	99.19	99.06	99.06	99.05
	MeanError	0.01	0.27*	0.02*	0.02*
swd	accuracy	50.17	57.31v	56.01v	56.77v
	MeanError	0.27	0.35*	0.26v	0.26v
Veteran	accuracy	89.31	88.48	88.70	86.88
	MeanError	0.09	0.32*	0.12*	0.13*
AVERAGE	accuracy	78.45	77.08	74.59	76.69
	MeanError	0.12	0.31	0.15	0.14

Table 3. Results for RepTree algorithm

Dataset		Local Ordinal RepTree	Multiclass RepTree	Ordinal RepTree	RepTree
auto93	accuracy	82.41	79.73	80.14	80.06
	MeanError	0.14	0.35*	0.20*	0.19*
autoHorse	accuracy	94.45	92.34	94.01	93.17
	MeanError	0.05	0.29*	0.07	0.07
autoMpg	accuracy	81.68	81.34	80.66	80.41
	MeanError	0.14	0.34*	0.17*	0.17*
autoPrice	accuracy	88.86	87.99	88.35	87.81
	MeanError	0.09	0.31*	0.10	0.11
bodyfat	accuracy	96.78	98.88	98.88	98.80
	MeanError	0.03	0.27*	0.01v	0.01v
cleveland	accuracy	71.08	71.73	68.39	71.36
	MeanError	0.21	0.36*	0.26*	0.24*
Cloud	accuracy	86.32	88.54	87.78	88.70
	MeanError	0.12	0.31*	0.11	0.10
Cpu	accuracy	98.04	97.00	96.95	97.29
	MeanError	0.01	0.28*	0.04*	0.03*
Era	accuracy	25.68	19.24*	26.20	26.60
	MeanError	0.18	0.20*	0.18	0.18
Esl	accuracy	66.08	60.59*	62.65	62.37
	MeanError	0.10	0.19*	0.11	0.11*
fishcatch	accuracy	96.71	94.88	94.05	94.70
	MeanError	0.03	0.28*	0.05	0.04
housing	accuracy	80.43	79.51	79.03	78.65
	MeanError	0.16	0.34*	0.18	0.18
hungarian	accuracy	78.62	78.70	78.46	78.46
	MeanError	0.17	0.34*	0.19	0.19
Lev	accuracy	63.16	60.43*	60.79	59.87*
	MeanError	0.20	0.31*	0.20	0.21*
lowbwt	accuracy	56.87	58.89	58.47	58.63
	MeanError	0.32	0.40*	0.34	0.33
pharynx	accuracy	69.79	65.06*	65.01 *	65.31*
	MeanError	0.28	0.40*	0.34*	0.34*
servo	accuracy	93.31	91.42	92.71	90.72
	MeanError	0.06	0.30*	0.07	0.08*
Strike	accuracy	98.97	99.21	99.21	99.21
	MeanError	0.01	0.27*	0.01	0.01
swd	accuracy	56.99	57.45	57.68	56.46
	MeanError	0.27	0.35*	0.26	0.27
Veteran	accuracy	89.20	91.26	91.19	90.90
	MeanError	0.11	0.31*	0.11	0.12
AVERAGE	accuracy	78.77	77.71	78.03	77.97
	MeanError	0.13	0.31	0.15	0.15

5 Conclusion

This paper is devoted to the problem of learning to predict ordinal (i.e., ordered discrete) classes. The local ordinal classification method discussed in this paper is applicable in conjunction with any learning algorithm that can output class probability estimates. According to our experiments in synthetic and real ordinal data sets, it manages to minimize the distances between the actual and the predicted classes, without harming but actually improving the classification accuracy in conjunction with DS, RepTree and NB algorithms. Drawing more general conclusions from these experimental data seems unwarranted. Our results so far show that the proposed methodology for predicting ordinal classes can be naturally derived from classification algorithms, but more extensive experiments will be needed to establish the precise capabilities and relative advantages of this methodology.

For large datasets, the benefit of local ordinal models is somewhat offset by the cost of storing and querying the training dataset for each test set instance. For this reason, in a following project we will focus on the problem of reducing the size of the stored set of instances while trying to maintain or even improve generalization performance by avoiding noise and over-fitting. In [4], numerous instance selection methods that can be combined with the proposed technique can be found.

References

1. C. G. Atkeson, A.W. Moore, S. Schaal, Locally weighted learning. Artificial Intelligence Review 11 (1997) 11–73.
2. E. L. Allwein, R. E. Schapire, Y. Singer, Reducing multiclass to binary: A unifying approach for margin classifiers. Journal of Machine Learning Research 1 (2000) 113–141.
3. C.L. Blake, C.J. Merz, UCI Repository of machine learning databases. Irvine, CA: University of California, Department of Information and Computer Science. [http://www.ics.uci.edu/~mlearn/MLRepository.html] (1998).
4. H. Brighton, C. Mellish, Advances in Instance Selection for Instance-Based Learning Algorithms, Data Mining and Knowledge Discovery, 6 (2002) 153–172.
5. P. Domingos, M. Pazzani, On the optimality of the simple Bayesian classifier under zero-one loss. Machine Learning, 29 (1997) 103-130.
6. E. Frank, M. Hall, B. Pfahringer, Locally weighted naive Bayes. Proc. of the 19th Conference on Uncertainty in Artificial Intelligence. Acapulco, Mexico. Morgan Kaufmann (2003).
7. E. Frank, M. Hall, A simple approach to ordinal prediction, L. De Raedt and P. Flach (Eds.): ECML 2001, LNAI 2167, (2001) 145-156, Springer-Verlag Berlin.
8. W. Iba, P. Langley, Induction of one-level decision trees. Proc. of the Ninth International Machine Learning Conference (1992). Aberdeen, Scotland: Morgan Kaufmann.
9. S. Kramer, G. Widmer, B. Pfahringer, M. DeGroeve, Prediction of ordinal classes using regression trees. Fundamenta Informaticae (2001).
10. C. Nadeau, Y. Bengio, Inference for the Generalization Error. Machine Learning 52(3): 239-281 (2003).
11. I. Witten, E. Frank, Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations, Morgan Kaufmann, San Mateo (2000).

Using Genetic Algorithms and Decision Trees for *a posteriori* Analysis and Evaluation of Tutoring Practices based on Student Failure Models

Dimitris Kalles and Christos Pierrakeas
Hellenic Open University, Laboratory of Educational Material and Educational Methodology, Sachtouri 23, 26222, Patras, Greece
{kalles, pierrakeas}@eap.gr

Abstract. Many students who enrol in the undergraduate program on informatics at the Hellenic Open University (HOU) fail the introductory course exams and drop out. We analyze their academic performance, derive short rules that explain success or failure in the exams and use the accuracy of these rules to reflect on specific tutoring practices that could enhance success.

1 Introduction

The Hellenic Open University's (HOU) primary goal is to offer university-level education using distance learning methods and to develop the appropriate material and teaching methods to achieve this goal. The HOU offers both undergraduate and postgraduate studies and its courses were initially designed and first offered in 1998 following the distance learning methodology of the British Open University. The HOU was founded in 1992 and currently (2005) nearly 25,000 students are enrolled.

The undergraduate programme in informatics is heavily populated, with more than 2,000 enrolled students. About half of them currently attend junior courses on mathematics, software engineering, programming, databases, operating systems and data structures. A key observation is that substantial failure rates are consistently reported at the introductory courses.

Such failures skew the academic resources of the HOU system towards filtering the input rather than polishing the output, from a quantitative point of view. Even though this may be perfectly acceptable from an educational, political and administrative point of view, we must analyse and strive to understand the mechanism and the reasons of failure. This could significantly enhance the ability of HOU to fine-tune its tutoring and admission policies without compromising academic rigour.

Please use the following format when citing this chapter:
Kalles, Dimitris, Pierrakeas, Christos, 2006, in IFIP International Federation for Information Processing, Volume 204, Artificial Intelligence Applications and Innovations, eds. Maglogiannis, I., Karpouzis, K., Bramer, M., (Boston: Springer), pp. 9–18

There are two key educational problems that have been identified as being core aspects of these failures. The first is that these courses are heavy on mathematics and adult students have not had many opportunities to sharpen their mathematical skills since high-school graduation (which has typically occurred at about 10 years prior to enrolling at HOU). The second is that the lack of a structured academic experience may have rendered dormant one's general learning skills and attitudes.

Our approach to investigating this problem uses increasingly rudimentary technology for data analysis. We use genetic algorithms to derive short decision trees that explain student failure [1, 2].

In this paper we expand that work by investigating differences in the accuracy of the induced models. We focus on short models that are easier to communicate among peers and question whether these differences might be attributed to the versatility of the tutoring practices. The results support our intuition about which practices better smooth out the disadvantages that arise due to some students' special circumstances. These results are now used as supporting data when we attempt to convince fellow tutors of the potential of some specific tutoring practices.

This paper is structured in three subsequent sections. In the next section, we briefly review the problem of predicting student performance at large, and the related techniques we have been using at HOU. We then single out three modules which have clearly different policies in dealing with students who have failed an exam and devise a set of experiments to observe whether these policies can be evaluated by a machine learning model. Finally, we argue about the ability to carry out these experiments at a larger scale and discuss the potential implications of our findings from an educational point of view.

2 Background

The work reported in this paper is part of an effort to analyze data at an institutional level, so we first briefly cover some essential background. We first present the application domain, then we present some key aspects of the technology used and, finally, we summarize the results obtained to date.

2.1 Operational issues

The educational philosophy of Open Universities around the world is to promote "life long education" and to provide adults with "a second educational chance" [3]. The method used is known as "distance learning" education, hence the widely used acronym ODL standing for Open-and-Distance-Learning.

In open and distance learning, dropout rates are definitely higher than those in conventional universities. Relatively recently, the Open Learning journal published a volume on issues on student retention in open and distance learning, where similarities and differences across systems is discussed, highlighting issues of institutions, subjects and geographic areas [4].

The vast majority (up to 98%) of registered students in the "Informatics" program, upon being admitted at HOU, selects the module "Introduction to

Informatics" (INF10). Following that, and according to university recommendations, they will typically select the modules "Fundamental Software Engineering" (INF11) and "Mathematics" (INF12). These modules are the most heavily populated and serve as test-beds for experimentation.

A module is the basic educational unit at HOU. It runs for about ten months and is the equivalent of about 3-4 conventional university semester courses. A student may register with up to three modules per year. For each module, a student is expected to attend five plenary class meetings throughout the academic year (a class contains about thirty students). Each meeting is about four hours long and may be structured along tutor presentations, group-work and review of assigned homework. Furthermore, each student must turn in some written assignments (typically four or six), which contribute towards the final grade, before sitting a written exam.

We have embarked on an effort to analyze the performance of high-risk students [1, 2, 5]. Key demographic characteristics of students (such as age, sex, residence etc), their marks in written assignments and their presence or absence in plenary meetings may constitute the training set for the task of explaining (and predicting) whether a student would eventually pass or fail a specific module. It is important to mention that the great majority of students dropped out after failing to deliver the first one or two written assignments. It is, thus, reasonable to assert that predicting a student's performance can enable a tutor to take early remedial measures by providing more focused coaching, especially in issues such as priority setting and time management.

2.2 Summarizing the technology: decision trees and genetic algorithms

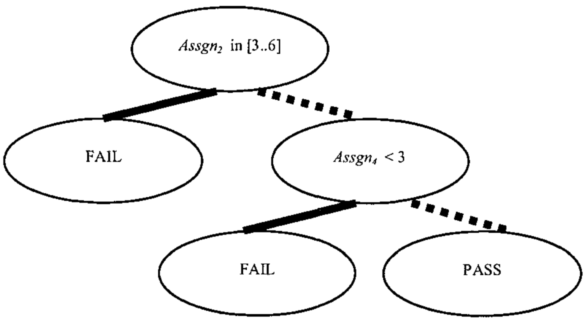

Fig. 1. A sample decision tree

A decision tree [6] for the failure analysis problem could look like the one in Figure 1. In essence, it conveys the information that a mediocre grade at an assignment, turned in at about the middle (in the time-line) of the module (containing 4 assignments altogether), is an indicator of possible failure at the exams, whereas a non-mediocre grade refers the alert to the last assignment. An excerpt of a training set that could have produced the above tree could be the one shown in Table 1.

Table 1. A sample decision tree training set

$Assgn_1$	$Assgn_2$	$Assgn_3$	$Assgn_4$	Exam
...	...	...	...	...
4.6	7.1	3.8	9.1	PASS
9.1	5.1	4.6	3.8	FAIL
7.6	7.1	5.8	6.1	PASS
...	...	...	...	...

Genetic algorithms can directly evolve binary decision trees [7] that explain and/or predict the success/failure patterns of junior undergraduate students. To do so, we evolve populations of trees according to a fitness function that allows for fine-tuning decision tree size vs. accuracy on the training set. At each time-point (in genetic algorithms dialect: *generation*) a certain number of decision trees (*population*) is generated and sorted according to some criterion (*fitness*). Based on that ordering, certain transformations (*genetic operators*) are performed on some members of the population to produce a new population. This is repeated until a predefined number of generations is reached (or no further improvement is detected).

These concepts form the basis of the GATREE system [8], which was built using the GAlib toolkit [9]. A mutation may modify the test attribute at a node or the class label at a leaf. A cross-over may exchange parts between decision trees.

The GATREE fitness function is:

$$fitness(Tree_i) = CorrectClassified_i^2 * \frac{x}{size_i^2 + x}.$$

The first part of the product is the actual number of training instances that a decision tree (a member of a population) classifies correctly. The second part of the product (the size factor) includes a factor x which has to be set to an arbitrary big number. Thus, when the size of the tree is small, the size factor is near one, while it decreases when the tree grows big. This way, the payoff is greater for smaller trees. Of course, this must be exercised with care since we never know whether a target concept can be represented with a decision tree of a specific size.

2.3 Summarizing past findings and setting the context

Initial experimentation [1] consisted of several Machine Learning techniques to predict student performance with reference to the final examination. The WEKA toolkit [10] was used and the key finding, also corroborated by our tutoring experience, is that success in the initial written assignments is a strong indicator of success in the examination. A surprising finding was that demographics were not important.

Follow-up experimentation [2] using the GATREE system [8] initially produced significantly more accurate and shorter decision trees. That stage confirmed the qualitative validity of the original findings (also serving as result replication) and set the context for experimenting with accuracy-size trade offs. That experimentation spanned three academic years, covered the three introductory modules INF10, INF11 and INF2, and validated that genetic induction of decision trees could indeed produce very short and accurate trees that could be used for explaining failures.

We have already documented that drop-out is a significant issue in ODL universities. What is most important, however, is that drop-out usually occurs early in the studies. Failure on a senior year course should simply postpone graduation as the fundamental commitment to studying has been already made. However, failure in a junior course, and for the HOU case, this refers to the INF10, INF11 and INF12 modules, can contribute to a decision to drop out both because the learning investment is not yet large enough to warrant a certain attitude of persistence and because the student may not have had the time to familiarize oneself with the distance learning mode of education (which, given time, allows one to dovetail studying more effectively with other activities).

By regulations, a student who fails a module examination can sit the exam on the following academic year. Such students are only assigned to student groups for examination purposes and the group tutor is responsible for marking their papers only; we thus refer to them as "virtual" students (should they fail their exam for a second year, they must take the module afresh, in which case they are conventionally assigned to a group and cease to be virtual).

Virtual students are not entitled to attending plenary sessions, and to having their assignments graded by the group tutor (as a matter of fact they are not even requested to submit assignments). In practice this regulation may be relaxed by a tutor, who may opt to extend an invitation to attend some plenary sessions to these virtual students usually. Usually, all tutors of a module will either accept or decline to relax the regulation. Of course, there is no focused follow-up of the progress of virtual students, as opposed to the case with typical students.

Any attempt to address these realities involves a political decision that must necessarily take into account the university's administrative regulations.

One step taken by tutors of the INF10 and INF11 modules is to hold a plenary marking session of tutors for each module after an examination, and to discuss variations in individual marking styles based on a predefined assignment of points to exam questions. This is especially important for problems that involve design or prose argumentation. We note that this practice is not widespread within HOU.

A further ad hoc step taken (during the 2003-4 academic year) by the INF11 tutors was to group all virtual students in one group and assign one experienced tutor to that group, as opposed to the usual practice of distributing virtual students across tutors. These students were fully supported by an asynchronous discussion forum and by synchronous virtual classrooms. The tutor did neither hold a physical meeting nor correct any assignments. This was in line with the HOU regulations and, coincidentally, served as a convenient constraint on the "degrees of freedom" of the educational experiment.

We now establish interesting indicators on the effectiveness of these approaches.

3 The experimental environment

We use GATREE for all experiments (even the basic version allows for unlimited experimentation with the x parameter in the fitness function, essentially treating x as an accuracy-vs.-size bias "knob").

For all experiments we used the default settings for the genetic algorithm operations (cross-over probability at 0.99, mutation probability at 0.01, error rate at 0.95 and replacement rate at 0.25). All experiments were carried out using 10-fold cross-validation, on which all averages are based. Because the data sets are reasonably large, ranging from 500 to 1000 student records, and because 10-fold cross-validation is a widely acceptable testing methodology, we opt to not report standard deviations. The experiments were made with a generations/population:150/150 configuration.

All data refer to the 2003-4 academic year. They do not differentiate between typical and virtual students.

Our methodology is the following: we attempt to use the student data sets to develop success/failure models represented as decision trees. We then use the differences between the models derived when we omit some attributes to reflect on the importance of these attributes. The results are then used to comment on alternative educational policies for dealing with virtual students.

We first try to deal with the issue whether we might be able to obtain an overall (typical and virtual students included) model that deals with explaining (and, ultimately, predicting) exam success, across the three modules that have three distinct policies.

The first experimental session attempted to produce short decision trees that could be used to explain the failure model of students in each module. For this, the x knob was set to 1000 (the minimum possible value). For each module, four (4) experimental batches were conducted and the results are shown in Table 3.

Table 2. Results for x=1000, gen/pop:150/150 GATREE decision trees

Data Set	Accuracy (in %)	Size (in nodes)
INF10: Basic	78.20	3
INF10: Basic_T	78.20	3
INF10: Basic_Y	82.58	6
INF10: Basic_TY	82.02	6
INF11: Basic	82.82	5
INF11: Basic_T	82.05	5
INF11: Basic_Y	81.28	6
INF11: Basic_TY	81.54	6
INF12: Basic_T	62.37	6
INF12: Basic_T	63.39	6
INF12: Basic_Y	67.97	6
INF12: Basic_TY	68.81	6