ECML/PKDD-2002 Workshop

IDDM-2002

2nd International Workshop on Integration and Collaboration Aspects of Data Mining, Decision Support and Meta-Learning

Helsinki, August 2002

Edited by

Marko Bohanec
Branko Kavšek
Nada Lavrač
Dunja Mladenić

ECML/PKDD-2002 Workshop

IDDM-2002

2nd International Workshop on Integration and Collaboration Aspects of Data Mining, Decision Support and Meta-Learning

Marko Bohanec, Branko Kavšek, Dunja Mladenić, Nada Lavrač
Jožef Stefan Institute, Ljubljana, Slovenia

Overall Aim

This workshop is a follow up to the successful IDDM-2001 workshop, which was held in Freiburg in September 2001 (http://www.cs.bris.ac.uk/~cgc/ECML-PKDD01/cfp.html). It is aimed at researchers and practitioners in Data Mining, Decision Support, and Meta-Learning, especially those interested in the main European research Consortia whose work focuses on the above topics, e.g., METAL, Sol-Eu-Net, and KDNet. Participants will gain a better appreciation of the issues facing the application and deployment of Data Mining solutions in the real world, as well as the approaches to the integration and collaboration of the mentioned disciplines. New ways of working together and combining results will be discussed, fostering further collaboration between participants' organisations. It is hoped that, as a result of this workshop, more people will work together more often, more effectively and in more sensible ways.

Workshop Topics and Goals

IDDM-2002 addresses the integration and collaboration aspects of Data Mining (DM), Decision Support (DS) and Meta-Learning (ML). In particular, this workshop is aimed at trying to upgrade the corresponding approaches and methodologies, such as CRISP-DM, through contributions, addressing the following issues.

Combining Data Mining with Decision Support

DM has a potential for solving DS problems, for example when previous decisions have been recorded as data to be used for analysis with DM tools. On the other hand, DS methodology usually results in a decision model, reflecting expert knowledge of decision makers. How can such expert knowledge be incorporated into problem

solutions by DM? Can it be used as background knowledge in relational data mining? Can such expert knowledge be induced automatically? Are there any systematic methodological means of combining the two approaches to problem solving? How can DM benefit from DS models, especially in cases where the data available for DM is incomplete or of insufficient quality?

Collaborative Data Mining

Usually, DM tasks are solved by a single individual or group of individuals working together. However, with the Internet and advances of group support methodologies and tools, DM tasks could be solved through a collaboration of different groups of researchers at different sites. Novel ideas, reviews of existing approaches, or different modes of collaboration should be explored (e.g., competitive vs. collaborative), and issues addressed such as infrastructure and methods for supporting distant collaborative work (e.g., how to integrate new individuals/groups following the start/stop-any-time principle).

Combining Results of Classifiers and Meta-Learning

Here, the emphasis is on novel ideas and/or reviews of existing approaches to model selection, model combination, model representation and all issues relevant to learning to learn (e.g., landmarking, performance prediction, knowledge transfer, data characterisation, meta-data collection and exploitation, standardised experimental setups/methods, etc).

Relational Data Mining

Most data in standard DM has the form of a single relational table. What if data is stored in multiple relational tables? Thus, how to combine the results of mining separate relational tables? A standard approach in ILP is to consider one table as the master data table, and all others as tables providing background knowledge. What if this is not natural? Would the mining of individual tables and combining results be a better solution? Are there other approaches to this problem?

DM, DS, and ML Integration: Methodology, Tools, and Standardization

This theme includes, but is not limited to, the following topics:

- ML tools for classifier and model selection
- ROC methodology for DM, DS and ML
- Data pre-processing tools and methods for DM and DS
- Representation languages for DM and DS models
- Standards supporting the exchange of DM and DS models for different applications and visualization tools, such as PMML (Predictive Model Markup Language)
- DS shells that seamlessly integrate models developed by DM
- Shared ontology and methodology for solving DM and DS problems

Organisation

Workshop Chairs
Marko Bohanec, Jozef Stefan Institute, Ljubljana, Slovenia
Dunja Mladenic, Jozef Stefan Institute, Ljubljana, Slovenia
Nada Lavrac, Jozef Stefan Institute, Ljubljana, Slovenia

Program Committee
Hendrik Blockeel, Katholieke Universiteit Leuven, Belgium
Patrick Brezillon, University Paris VI, France
Ivan Bruha, McMaster University, Canada
Peter Flach, University of Bristol, United Kingdom
Dragan Gamberger, Rudjer Boskovic Institute, Croatia
Christophe Giraud-Carrier, ELCA Informatique SA, Switzerland
Salvatore Greco, University of Catania, Italy
Marko Grobelnik, Jozef Stefan Institute, Slovenia
Alipio Jorge, University of Porto, Portugal
Krzysztof Krawiec, Poznan University of Technology, Poland
Steve Moyle, Oxford University, United Kingdom
Vladislav Rajkovič, University of Maribor, Slovenia
Roman Slowinski, Poznan University of Technology, Poland
Jerzy Stefanowski, Poznan University of Technology, Poland
Maarten van Someren, University of Amsterdam, The Netherlands
Olga Stepankova, Czech Technical University, The Czech Republic
Ljupco Todorovski, Jozef Stefan Institute, Slovenia
Tanja Urbancic, Jozef Stefan Institute, Slovenia
Ricardo Vilalta, IBM T.J. Watson Research Center, USA
Takahira Yamaguchi, Shizuoka University, Japan
Blaz Zupan, University of Ljubljana, Slovenia

Paper Acceptance

There were 25 papers submitted to this workshop. Each paper was reviewed by at least two reviewers. On this basis, 19 papers were selected for presentation at the workshop. Among these, 11 were accepted as *full* papers (text limited to 12 pages), and 8 as *short* papers (6 pages and shorter presentation).

Acknowledgements

We gratefully acknowledge support from the European Comission's project Sol-Eu-Net. We also wish to thank the members of the Program Committee for their timely and thorough review of the papers, as well as for their constructive suggestions to authors. We hope you find the workshop's material and presentations stimulating.

Marko Bohanec, Branko Kavšek, Nada Lavrač, and Dunja Mladenić
Helsinki, August 2002

Table of Contents

Prognosis-based Decision Support in Medicine A Divide and Conquer Approach

Ameen Abu-Hanna

Department of Medical Informatics, AMC-University of Amsterdam
Meibergdreef 15, 1105 AZ Amsterdam, The Netherlands

Abstract. Managers of clinical departments are constantly seeking ways to assess and improve their quality of care. In doing so they feel the need for decision support methods and tools. "Prognosis-based decision support" is an elemental way to achieve support where a model is learned for the prediction of a patient outcome indicative of care quality. The idea is that quality of care can be assessed by comparing the model's predictions for a patient population to the actual outcomes of these patients, and hence facilitate making decisions about future improvement of care delivery. This paper first describes the current main stream prognosis-based decision support process which is centered around a global logistic model. It then suggests an improved, more transparent method based on learning prognostic models that identify patient sub-populations (using a classification tree) for which specialized prognostic models (based on local regression) are built. The method is illustrated in the field of intensive care.

1 Introduction

Health Care is expensive and evidence for its effectiveness and efficiency is continuously sought. Many national and international quality programs have been set up to assess health care quality. For example our department is responsible for national registries such as those for intensive care and cardiology and international ones such as that for the European Renal Association. These registries include information about patients and outcomes of care such as mortality and co-morbidity. When the outcome is indicative of the quality of care, registries form valuable vehicles for analytic instruments aimed at the evaluation of the quality of care and supporting decisions for its constant improvement.

Prognostic models for outcome prediction form indispensable ingredients among these evaluative tools [1, 15] enabling what can be termed as "prognosis-based decision support". Intensive care (IC), which will be used throughout the paper for illustrative purposes, is a good example of this. Here various prognostic models, such as the SAPS-II [13], version II of the Simplified Acute Physiology Score, have been developed to estimate the probability of in-hospital mortality. In-hospital mortality concerns deaths in the hospital during or after stay in an Intensive Care Unit (ICU). Like many other prognostic models in medicine these are statistical models that can be characterized by their use of a small

set of covariates, at the heart of which is a *score* variable reflecting the patient's overall severity of illness, and by their reliance on the probabilistic logistic model. The common practice in decision support is to arrange predicted probabilities of death in groups, e.g. quantiles, and compare each group's mean with the fraction of observed deaths corresponding to that group. Discrepancies are then assessed by experts to find out whether actions to improve care delivery should be taken.

This paper aims at enhancing the decision support process by introducing the notion of patient sub-population in the process: Instead of using one global logistic regression model, a classification tree is induced to devise patient groups for which local models, both parametric and nonparametric, can be fit. The idea is twofold: first, clinicians are used to think in terms of sub-groups and hence stratification on these groups brings transparency allowing for focusing on them and, second, the method allows for an ensemble of models that together can fit the data better than one global parametric model.

A balance must be sought between the ability to divide patients into possibly many sub-groups to enhance focus on their physiological homogeneity on the one hand, and on the other hand, lumping up patient descriptions based on the more abstract notion of overall severity of illness that avoids fragmentation and allows for parametric models that assume monotonicity (higher score implies higher probability of death). Our approach to this problem is to devise patient sub-populations, using a classification tree, based on the physiological variables that *underlie* the severity of illness score while using the overall lumped score itself in the local prognostic models. The hypothesis is hence that the severity of illness score holds additive value that can be tapped and made explicit by the tree. The number of sub-groups can be controlled by pruning the tree at places where the local models fail to enhance prognostic performance. We show in this paper that in measuring prognostic performance for decision support in a quality of care program, the popular error rate and even area under the Receiver Operator Characteristic (ROC) curve–which are measures of *accuracy*– play a relatively minor role in the evaluation of models and we explain why measures of *precision* are usually more appropriate.

The paper is organized as follows. Prognosis-based decision support is introduced in Sect. 2 and illustrated in intensive care. Then in Sect. 3 aspects of model evaluation are provided where it is explained why measures of precision are important in decision support of quality of care programs. Sect. 4 presents our method for improved decision support based on a hybrid model and is illustrated in two variants. Sect. 5 concludes this paper.

2 Prognosis-based DS for Quality of Care

The need for assessing and assuring the quality of care in medicine arises from various reasons. In the intensive care the technologies to treat and monitor organ failure of critically ill patients are not only costly as consequence of the price tags of the advanced equipment, they also require a large number of skilled personnel to maintain and use these technical facilities 24 hours a day, 7 days a week.

In addition, differences in the view on the best delivery of care, professional ambitions, budgetary constraints and insurance regulations now have prompted physicians and managers to assess the quality of IC treatment.

One way to achieve objective appraisal of the quality of the care process, especially if experimentation is impossible or unethical, is prognosis-based: outcome data such as mortality, morbidity and length of stay are compared to *predicted* outcome values. This is practiced in various programs such as those described in [18, 10]. The predictions should take into account the characteristics of the patient population admitted to an ICU. This is called case-mix adjustment and is usually quantified by a score of the severity of illness of each patient at the time of admission. In the main, existing IC case mix adjustment models are concerned with predicting mortality. Prognostic models that make these case-mix adjusted predictions lie at the heart of quality assessment efforts.

2.1 Current Practice using Logistic Regression

Two major prognostic systems in IC are the Acute Physiology and Chronic Health System, APACHE [11] and the Simplified Acute Physiology Score, SAPS [13]. Both come in different versions (e.g. I and II). These systems, like many others, are based on logistic regression.

A logistic regression model is a parametric model that specifies the probability of a dichotomous variable Y having the value 1 –indicating the occurrence of an event such as death– given the values of the covariates of the model. It has the following form:

$$p(Y = 1|\boldsymbol{x}) = \frac{e^{g(\boldsymbol{x})}}{1 + e^{g(\boldsymbol{x})}} \tag{1}$$

where $\boldsymbol{x} = (x_1, ..., x_m)$ is the covariate vector. For m variables (also called predictors) the *logit function* $g(\boldsymbol{x})$ has the following form:

$$g(\boldsymbol{x}) = \beta_0 + \sum_{i=1}^{m} \beta_i x_i \tag{2}$$

where β_i, $i = 1, ..., m$, denote the coefficients of the m predictors. Fitting a logistic regression model implies finding estimates of the $\beta_0, ..., \beta_m$ that maximize the likelihood of the model (that is, the probability of the data given the model). Note that the conditional probability in Eq. (1) is in effect the conditional expectation $E(Y = 1|\boldsymbol{x})$ because Y is binary.

The monotonicity of logistic regression means that it is not a good idea to represent a raw physiological feature such as temperature of the patient directly as a covariate because an increase e.g. from 35 to 37 Celsius degrees should have an opposite effect on the probability of the adverse event than an increase from, say, 37 to 39 Celsius degrees. Because of this and due to other reasons,

the APACHE and SAPS use one or more aggregate *scores* for the quantification of the severity of illness as covariates instead of the individual features. A higher score corresponds to greater deviation from the healthy status and hence a higher probability of death. Different elements contribute to this total additive score such as physiological variables e.g. heart rate, and white blood cell count; demographic variables e.g. age; and covariates concerning earlier chronic history. For example the total SAPS-II score ranges from 0 to 163 points and is related to a probability of hospital mortality based on the following logit model:

$$g(score) = -7.7631 + 0.0737\, score + 0.9971\, ln(score + 1)$$

Such models are used by quality of care programs such as the National Intensive Care Evaluation (NICE) program. NICE is a foundation established in 1996 by an initiative of a professional group of intensivists to gain insight and to improve the effectiveness and efficiency of Dutch intensive care units (ICUs). Many different ICUs in the Netherlands are currently participating in NICE. The following is a sketch of the procedure followed by NICE in supporting decisions about quality of care.

1. Each participating ICU provides its care-related data following a standard format respecting agreements which are semi-formalized in a data dictionary. For each patient more than 200 items are collected including patient description and various outcomes including length of stay and mortality.
2. The information from all participants is accumulated and stored in the NICE registry after data quality validation procedures have taken place.
3. Based on data from all ICUs various prognostic models are constructed including SAPS and APACHE.
4. Each prognostic model is validated on a large test set that includes data from all participating ICUs. The Hosmer-Lemeshow tests for goodness of fit are usually used in this step which imply the comparison between predicted probabilities and observed proportions of mortality within various probability groups.
5. Once a prognostic model is found satisfactory it may be considered as predicting the "average outcome of a national" ICU. It is then used to predict probability outcomes of new data from each ICU. For *each* ICU the model's predictions are once again lumped into probability groups and compared with the actual proportion of mortality in that ICU.
6. Representatives of the participating units meet periodically and discuss discrepancies between their "performance" and the "national average" predictions for their patients. The reasons for the discrepancy are sought: is there a discrepancy in definitions of some data items? is the model to be trusted for a specific risk group? is there something that can be learned from those ICUs that are faring better than average?
7. Based on the outcomes of the last step, decisions are made for future improvement of care.

3 Model Evaluation Aspects

Before we present our method for constructing prognostic models for improved decision support a word is due about model evaluation aspects in quality of care programs. Consider two different models M1 and M2 that when given a specific score, $score_0$, as the covariate value they provide two different mortality probability estimates, say .55 and .85. If a prognosis is sought for *individual* patients with $score_0$, both models will predict non-survival for each of these patients (because the probability assigned for the event –non-survival– is greater than the probability assigned for survival). Assuming there are more non-survivors than survivors among the patients with $score_0$, these two models are equally effective in discriminating between the two survival states and hence will inflict the same total error rate (the Bayes error rate) in this group.

There is however a substantial difference between the probability estimates of the two models which cannot be ignored if one is interested in the *probability* of an event within a *group* of patients. To judge the performance of an ICU, one compares the *estimated probability* of non-survival with the *observed mortality rate* of a specific group. If 70% of patients with $score_0$ did not survive then according to M1 the ICU is performing very poorly but according to M2 it is performing much better than expected for that group of patients. We are hence interested in a *precise* model that provides honest estimates of the *true probability* of an event rather than merely a discriminating model with the ability to assign the highest probability to the actual event or class. However, error rate still forms one of the most used measures for evaluating classifiers (in machine learning and statistics) [7]. Along with other measures of accuracy, such as ROC (Receiver Operator Characteristic) and the aggregated area under its curve (AUC), these measures are useful but alone do not tell the whole story. See [7] for a comprehensive framework of evaluation methods and measures.

Current main stream evaluation of logistic regression models such as the APACHE-II and SAPS-II models usually rely on the Hosmer-Lemeshow statistics [8] where it is referred to as calibration (see [17] for a more general discussion). These are essentially precision measures that group predicted probabilities in *non-overlapping* regions. A major disadvantage of the Hosmer-Lemeshow statistics is that they have been shown to be quite sensitive to the cut-off points used to form the expected probability regions [9].

In this work we used various measures of accuracy and precision in order to inspect the performance of prognostic models. These include the Brier score, also known as the mean quadratic error as a measure of accuracy. For precision we obtain information about the "true" probability from the *test* set in two ways: A direct one where local regression is used to smooth mortality data where subsequently the quadratic error of the predicted ones is used as the (im)precision measure. In a second indirect way we compare a transformation *summary* obtained from the predicted probabilities on the test set to a similar summary obtained from the true observed outcomes from that test set. The difference between the summaries leads to a statistic that is significantly different

from 0 when precision is bad. We use the logarithmic transformation [7] to obtain each summary.

Our precision measures do not require division in non-overlapping probability groups in the expected probability of mortality, and hence do not share the drawbacks of the Hosmer-Lemeshow statistics mentioned in [9]. Our precision measures rely on some smoothing aspect of the neighborhood of a point of interest. This neighborhood is dynamic and overlaps with other neighborhoods. Moreover our measures will be obtained on patient groups sharing various inherent characteristics (e.g. physiological variables) as will be demonstrated below.

4 Suggested Improved Method: divide and conquer

In this section we motivate the use of a method aimed at a better understanding of the IC data and the models fitted to it in order to enhance decision making. In this method we try to exploit information that is implicit in the severity of illness score covariate(s). For illustration purposes we will concentrate on a model that uses only one score as in the SAPS model. As an example of the information that may be masked by a score consider that a patient with unscheduled surgery (8 points) and normal heart rate (0 points) will score as a patient with a medical admission (6 points) with a heart rate between 40 and 69 (2 points), assuming for simplicity that they score the same for the other attributes. Our approach is to create sub-groups that share some of their underlying attribute values.

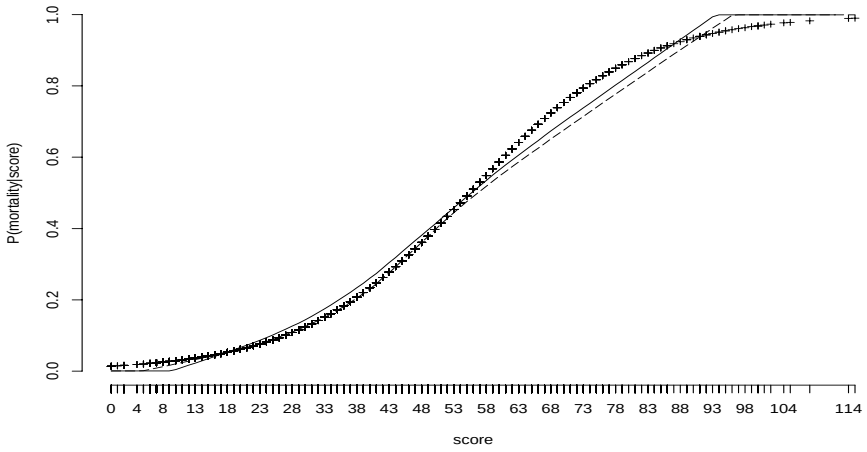


Fig. 1. Predictions of mortality on a test set by logistic regression (points marked with +) and Lowess (solid line). The dashed line represents Lowess smoothing on the test set itself.

A classification tree is used[1] to create these groups while using mortality as the classification attribute. The idea is that creating groups (the divide step) with different distribution of the mortality attribute and fitting local models (the conquer step) on them will prove advantageous if compared to the original global model. The total score however will be the sole covariate in the local models fitted once a group is identified. For the local models we have experimented with both (parametric) logistic regression models and nonparametric local regression models based on Cleveland's Lowess procedure [6]. As an example consider the predictions of two models (trained previously on a separate training set of some similar patient group) in Fig. 1: the predictions are those of Logistic regression (with its characteristic S-shape), and the non-parametric local regression (the solid line) according to Lowess. The dashed line represents Lowess smoothing on the test set itself in order to provide an idea of the "true" mortality function. Note how the two models over or underestimate this "true" probability in this test set.

Table 1. Characteristics of important attributes in the dataset.

Variable name	*Description*		*Mean±s.d.*	*Normal range*
syst.min	minimal systolic blood pressure		92.0±32.4	100-199
urine.24	urine production in first 24 hrs		2.6±2.3	>1
heartr.min	minimum heart rate		71.2±23.0	70-119
bicarb.min	minimum bicarbonate		22.4±5.2	≥20
bicarb.max	maximum bicarbonate		25±4.5	≥20
gcs.low	lowest Glasgow Coma Scale		13.8±3.2	15
wbc.max	maximum white blood cell count		12.1±11.4	1-19.9
Variable name	*Description*	*value*	*Freq.*	
adm.type	admission type			
	medical admission	1	45.3%	
	unscheduled surgical	2	17.7%	
	scheduled surgical	3	37.0%	

As an illustration of the method, consider the simple global logistic model, termed, N-SAPS (the N of NICE), that was developed on a training set of 5218 IC admissions of NICE, using only the SAPS-II score as covariate (without the logarithm of the score, in order to concentrate only on the score). This model is later evaluated on a separate test set of 2585 admissions. The N-SAPS model has the following form:

$$g(score) = -4.2548 + 0.0737\, score$$

[1] We use the *rpart* function available in the Splus statistical package which is an implementation of CART [4].

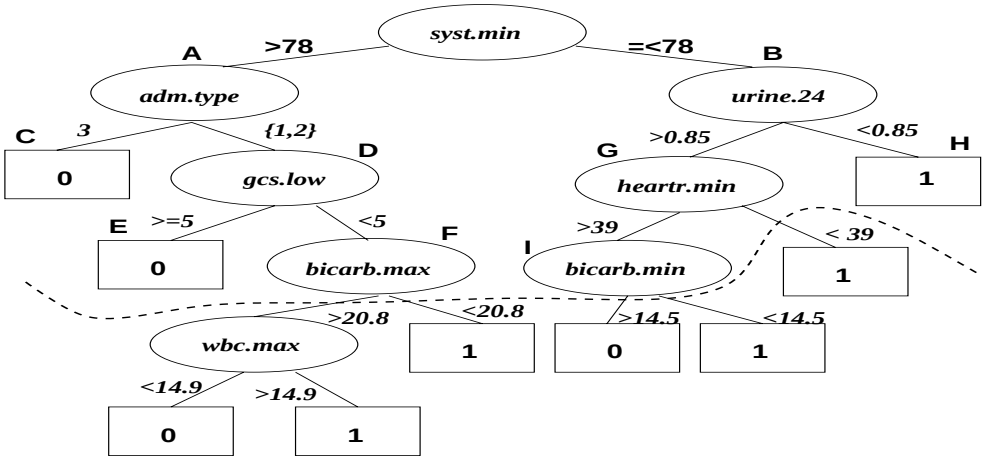


Fig. 2. The IC classification tree based on physiological attributes and admission type, see Table 1 for attribute descriptions. The leaves denote the majority class. The nodes above the dashed line constitute the identified groups with sufficient size.

N-SAPS was then compared with a model constructed according to our method whose classification tree appears in Fig. 2. Attribute names are explained in Table 1. The restriction to a binary tree is aimed at combating fragmentation and because all attributes are either continuous or ordinal (admission type can be clinically viewed this way in the sense that higher values of admission type indicate more serious conditions). Information gain based on entropy was used as the criterion for the selection of attributes in the tree. Note that patients with the same score could end up in different nodes in this tree as most scores can be obtained by different combinations of e.g. physiological variables.

When viewing the probability functions in the different nodes on the training set by smoothing the raw mortality data (using Lowess) there were obvious differences between them. The functional form at each node turned out to be *qualitatively consistent* in random samples of the data set, as long as there were sufficient instances in each node. This suggests that the tree is exploiting information which has not been explicitly used in the score and hence is masked from N-SAPS. One way to use this insight in decision support is simply to induce the tree partition in data from a different ICU, or data from the same ICU but taken at a different time and inspect the conformance to this qualitative behavior to detect differences.

When formally inspecting the model performance by obtaining our performance measures for the prognostic models on the tree nodes with at least 100 cases (the nodes appearing above the dashed line in Fig. 2) the following results were obtained (see Tables 2 and 3): The N-SAPS model has been outperformed in accuracy and precision in most of the cases meaning that the attributes underlying the score variable do have an added value. The local Lowess models did not outperform the local parametric logistic regression model indicating perhaps

Table 2. Accuracy measures for the three models (values have been multiplied by $*10^3$). Bold values mean accuracy is significantly better than in the N-SAPS model.

TLR and TLW stand for the local logistic and Lowess models.

		Node and ♯instances in test set								
		A	B	C	D	E	F	G	H	I
Model	*Msr*	2072	513	900	1172	1063	109	402	111	372
N-SAPS	*Brier*	104.67	173.96	50.106	146.56	137.01	239.74	183.57	139.17	176.04
TLR	*Brier*	104.62	174.14	**49.73**	146.12	136.82	238.91	182.36	**126.17**	174.83
TLW	*Brier*	105.01	175.15	**49.861**	146.53	137.40	**231.74**	183.17	128.08	175.44

Table 3. Direct and indirect precision measures (values have been multiplied by $*10^3$). Values in bold mean they are significantly better than the N-SAPS model values. Boxed values mean statistically significant bad precision.

		Node and ♯instances in test set								
		A	B	C	D	E	F	G	H	I
Model	*Msr*	2072	513	900	1172	1063	109	402	111	372
N-SAPS	*direct*	0.59	0.52	1	1.1	0.52	16.42	2.1	10.1	1.75
	indirect	-987.57	613.81	-3107.4	1725.1	1175.5	2528.5	1127.6	-785.28	264.57
TLR	*direct*	**0.57**	**0.34**	**0.85**	**0.35**	0.47	**15.95**	**0.4**	**0.42**	**0.35**
	indirect	-694.4	-42.701	-537.47	-265.71	-635.82	2051	-243.46	-76.171	-784.05
TLW	*direct*	**0.42**	**0.2**	**0.88**	**0.4**	**0.3**	**11**	**0.56**	**0.91**	**0.73**
	indirect	-1397.7	478.68	-1147.4	-407.25	-574.09	1633.4	398.05	-442.94	-290.18

that the nodes do not contain sufficient instances to allow for a complete non-parametric model. The non-parametric model is however quite useful in order to inspect the qualitative shape of the mortality function.

When inspecting the results were N-SAPS could be improved, the following sub-populations are notable:

Node **C** Patients with a normal to high systolic blood pressure who are admitted after scheduled surgery; these patients are relatively healthy.

Node **F** Patients with a normal to high systolic blood pressure who are admitted after unscheduled surgery or for a non-surgical (medical) reason and who have a very disturbed neurological status reflecting a high severity of illness.

Node **H** Patients with low blood pressure and low urine production reflecting possible heart and renal failures. These patients are seriously ill.

This result is in line with other studies in which the precision of current (global) logistic regression models is often found lacking at the extremes: patients which are relatively healthy and patients which are seriously ill. The difference is however that in our analysis one can characterize these groups in terms of their attribute values instead of establishing that the model does not calibrate for very low and very high scores where the different patient populations cannot be further discerned.

4.1 Deviation-based Trees

From an epidemiological point of view it is interesting for decision makers to communicate about patients using the notion of risk factors. We can employ our method to identify important risk factors that complement the severity of illness score that can be viewed as an overall risk factor. Instead of building trees based on the underlying physiological and other raw attributes one can use attributes that pertain to *deviation* from the values in the normal ranges. For example instead of using the ranges of heart-rate we use the "penalty" scores for heart-rate as a co-variate in the induction of the classification tree.

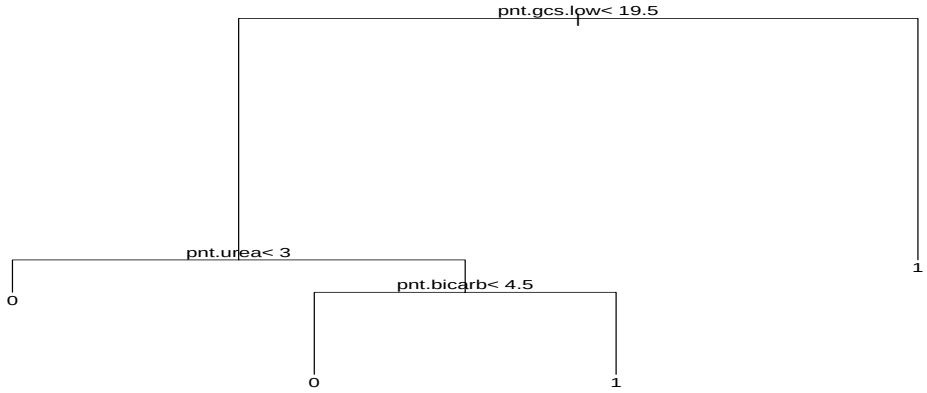


Fig. 3. Classification tree based on deviation, the prefix "pnt" indicates (deviation) points of the corresponding attribute. Each condition corresponds to the *left* branch. The 0 and 1 represent survival and death respectively.

To illustrate, consider Fig. 3 that shows a deviation-based tree induced from a similar intensive care training set (under slightly different circumstances). As pointed out in [16] one should be careful in interpreting risk factors. In fact due to the meaning of the new attributes as adversary conditions, any path that mixes a greater-than (">") condition (meaning worse condition) together with a less-than ("<") condition should be interpreted with care. For example, in the high risk group "$pnt.tgc.low < 19.5$ & $pnt.urea \geq 3$ & $pnt.bicarb \geq 4.5$" one should not conclude that the low value of *pnt.gcs.low* itself is associated with high risk. The fact that low *pnt.gcs.low* appears in the path of a high risk group is not by definition an indication of an interaction with the other factors in the path but could be simply indicative of being used as a "rest" category that the tree has created when using high *pnt.gcs.low* for constructing the first group.

This is a phenomenon introduced by the way trees are induced of which decision makers must be aware.

5 Conclusions

From this work, one can postulate that using the hybrid method of a classification tree with local prognostic parametric and non-parametric models provide better insight into the data and hence enhances decision support. One may conclude that the attributes underlying severity of illness scores can indeed contribute to better models. It is interesting to note that in our search for a balance for granularity of covariates we started from the severity of score which turns out to be too lumped, while in other approaches such as [3] one seeks a balance by searching for aggregations of (too) low level features.

The idea of comparing logistic regression with classification trees (e.g. as done in [14]) or the combination of a classification tree with other models are in themselves not new. In [12], for example, Naive Bayes Classifiers are fit on some of the tree nodes to boost classification. Recently a related idea to ours has been proposed in [5] for learning "Treed Models" based on Bayesian methods where also logistic regression models have been proposed. The contribution of our work lies in providing a motivated synthesis of modeling and evaluation concepts tailored to the specific constraints of decision support in quality of care programs with emphasis on precision. This emphasis on precision would, for example, not allow for models such as Naive Bayes Classifiers which are known, in the main, to have good and robust classification error but are often quite imprecise.

Further work includes putting our method to the test in the decision support process to see how it affects the intensivists' decision making. Another topic is to empirically investigate when to stop growing the tree and how to opportunistically combine all (here three) model types for providing the best prognosis. An important sequel to the work presented in this paper is the treatment of other outcome measures than mortality, such as length of stay in the ICU and organ failure scores registered daily for each patient, that although more complex to handle they do provide more sensitive measures of quality of care.

One might conclude that our results provide proof of concept that in this era of large amounts of electronic data, there is room for a variety of modeling concepts for enhancing decision support. We feel that the inspection of interesting patient sub-populations is an important enrichment to the traditional logistic regression models.

Acknowledgment Thanks are due to the board of the National Intensive Care (NICE) foundation for its support and feedback. The board consists of: G.J. Scheffer, R. Bosman, E. de Jonge, J.C.A. Joore, N.F. de Keizer, H.H.M. Korsten, J.G. van der Hoeven, P.H.J. van der Voort. Furthermore, we are grateful to all NICE participants for collecting the data. Special thanks to Nicolette de Keizer for feedback on this work.

References

1. Abu-Hanna A and Lucas PJF. Prognostic Models in Medicine AI and Statistical Approaches, (Abu-Hanna A. and Lucas PJF, eds.). Special issue of Methods of Information in Medicine 2001, 40:1–5.
2. Abu-Hanna A and de Keizer N. A Classification-Tree Hybrid Method for Studying Prognostic Models in Intensive Care, AIME 2001, 99–108.
3. Bohanec M, Zupan B, Rajkovic V. Applications of Qualitative Multi-attribute Decision Models in Health Care. International journal of Medical Informatics. 2000, vol. 58-59, 191–205.
4. Breiman L, Friedman JH, Olshen RA, Stone CJ. *Classification and Regression Trees.* Belmont: Wadsworth, 1984.
5. Chipman H, George E, McCulloch R. Bayesian Treed Models. Machine Learning, 48, 299-320, 2002.
6. Cleveland WS. Robust Locally Weighted Regression and Smoothing Scatterplots. J. Amer. Statist. Assoc. 74, 829–836, 1979.
7. Hand DJ. *Construction and Assessment of Classification Rules.* Chichester: John Wiley and Sons, 1997.
8. Hosmer DW and Lemeshow S. *Applied Logistic Regression*, Wiley, New-York, 1989.
9. Hosmer DW, Hosmer T, Le Cessie S, Lemeshow S. A Comparison of Goodness-of-fit Tests for the Logistic Regression Model. Statistics in Medicine 1997; 16:965–980.
10. de Keizer N. An Infrastructure for Quality Assessment in Intensive Care; Prognostic Models and Terminological Systems. PhD Thesis, 2000, University of Amsterdam.
11. Knaus W, Draper E, Wagner D, Zimmerman J. APACHE II: a Severity of Disease Classification System. Crit Care Med 1985; 13:818–829.
12. Kohavi R. Scaling Up the Accuracy of Naive-Bayes Classifiers: a Decision-Tree Hybrid. Proc. of the Second Int. Conference on Knowledge Discovery and Data Mining. 1996; 202–207.
13. Le Gall J, Lemeshow S, Saulnier F. A New Simplified Acute Physiology Score (SAPS-II) Based on a European/North American Multicenter Study. JAMA 1993; 270:2957–2963.
14. Long WJ. A Comparison of Logistic Regression to Decision-Tree Induction in a Medical Domain. Compt Bio Res 1993:74–97.
15. Lucas PJF and Abu-Hanna A. Prognostic Methods in Medicine (Lucas PJF and Abu-Hanna A. eds.). Special issue of Artificial Intelligence in Medicine. 1999; 15(2):105–119.
16. Marshall RJ. The use of Classification and Regression Trees in Clinical Epidemiology, Journal of Clinical Epidemiology 54 (6) (2001) pp. 603-609.
17. Miller ME and Hui SL. Validation Techniques for Logistic Regression Models. Statistics in Medicine, 1991, Vol 10, pp. 1213–1226.
18. Rowan K, Kerr J, Major E, McPherson K, Short A, Vessey M. Intensive Care Society's APACHE II Study in Britain and Ireland-II. BMJ 1993; 307:977–981.

Space Partitioning for Instance Reduction in Lazy Learning Algorithms *

M.C. Ainslie and J.S. Sánchez

Dept. de Llenguatges i Sistemes Informàtics, Universitat Jaume I,
Campus Riu Sec, E-12071 Castelló, Spain

Abstract. Lazy learning methods suffer from the indiscriminate storage of all training instances, resulting in large memory requirements and slow execution speed. This paper focuses on the problem of reducing the training set size by presenting new instance selection schemes.

1 Introduction

Many supervised learning algorithms use a collection of training instances, typically called training set (TS), to estimate the class label of new input vectors. Each instance in the TS has an attribute vector and a class label (the output value). After learning from the TS, the algorithm is presented with additional input vectors and must use some inductive bias to decide the output value. Methods that employ this technique are also known as lazy learners.

The nearest neighbour (NN) algorithm [5] is one of the most widely studied examples of lazy learning methods. During classification, the NN scheme employs an appropriate distance metric defined on the feature space to determine how close a new input vector x is to each instance in the TS, and uses the nearest instance to predict the class of x. An improved version of this corresponds to the k-NN rule, which consists in assigning a new input vector to the class most frequently represented among the k closest instances stored in the TS.

In general, lazy learning algorithms must decide which instances to store in the TS for use during classification in order to avoid excessive storage and time complexity, and possibly to improve classification accuracy by avoiding noise and overfitting. For example, the instances used to train the NN classifier are stored indiscriminately. This means that the NN approach has to search through all available cases to classify a new input vector, so it can become too slow during classification. On the other hand, since it stores every instance in the TS, noisy instances are also stored, possibly degrading significantly the accuracy.

Among the many proposals to reduce the storage requirements and time complexity of the NN rule, it is worth mentioning those that try to obtain a more efficient scheme by removing some instances from the TS. In this context, it is possible to differentiate between two main types of TS reduction techniques: those that retain a subset of the original instances [1,6,8] and, those that modify the training instances using a new representation [3,4].

* Partially supported by grant No. TIC2000-1703-C03-03 from the Spanish CICYT.

2 Training Set Size Reduction Techniques

In lazy learning, the problem of instance selection is primarily related to instance deletion as irrelevant and harmful cases are removed while retaining only critical instances. Others modify the instances themselves to reduce the TS size.

2.1 Reduction by Eliminating Instances

Hart's algorithm [6] is the earliest attempt at minimizing the number of stored instances by retaining only a *consistent* subset of the original TS. A consistent subset, S, of a TS, T, is some subset that correctly classifies every instance in T using the 1-NN rule. One generally is interested in the *minimal* consistent subset to minimize the cost of storage and computing time. Hart's condensing does not guarantee finding the minimal subset and furthermore, different subsets are given when the TS order is changed.

Aha et al. [1, 2] presented the incremental learning schemes IB1-IB4. With IB2, if a new case to be added can already be correctly classified by the current training instances, then the case is discarded and not stored at all. IB3 addresses the problem of keeping noisy instances by retaining only acceptable misclassified cases. IB4 is thought to handle irrelevant attributes by building a set of attribute weights for each class

Wilson [8] introduced the first editing proposal. Briefly, this consists in using the k-NN rule to estimate the class of each instance in the TS, and removing those whose class label does not agree with that of the majority of its k neighbours. This algorithm tries to eliminate mislabelled instances from the TS as well as close border instances, smoothing the decision boundaries.

2.2 Reduction by Generating Prototypes

Some algorithms try to modify the instances in order to reduce the TS size, instead of deciding which ones to retain. Some of them artificially generate prototypes in locations accurately determined.

Chang [3] introduced an algorithm in which each instance in the TS is initially considered as a prototype. Then, it consists in repeatedly attempting to merge the nearest two existing instances into a new single prototype. Two instances p and q are merged only if they are from the same class and, after replacing them with prototype z, the consistency property can be guaranteed.

Chen and Józwik [4] proposed an algorithm which consists in dividing the TS into subsets using the concept of *diameter of a set* (i.e., the distance between its two farthest points). The algorithm starts by partitioning the TS into two subsets by the middle point between the two farthest cases. The next division is performed for the subset that contains a mixture of instances from different classes. If more than one subset satisfies this condition, then that with the largest diameter is divided. The number of partitions will be equal to the number of instances initially defined. Finally, the subsets are replaced by their centroids, which will assume the same class label as the majority of cases in each subset.

3 Instance Reduction by Space Partitioning

The algorithms developed in this section are directed towards defining prototypes by partitioning the feature space. They will be here called IRSP1-IRSP4 (Instance Reduction by Space Partitioning). From now on, the original TS will be denoted by T, while S will refer to the resulting reduced set.

3.1 IRSP1

One problem associated with the heuristic introduced by Chen and Józwik refers to the fact that, in some cases, all instances from one of the classes can be eliminated from the TS. IRSP1 overcomes this by computing one centroid for each different class existing in the subset. We will obtain c prototypes per subset, being c the number of different classes present in it.

1. IRSP1(T, b):S
2. Let $b_c = 1$ (b_c is the current number of subsets in T), and $i = 1$.
3. Let $B = T$.
4. Find the two farthest points, p_1 and p_2, in B.
5. Divide the set B into two subsets B_1 and B_2, where
   $B_1 = \{p \in B : d(p, p_1) \leq d(p, p_2)\}$
   $B_2 = \{p \in B : d(p, p_2) < d(p, p_1)\}$
6. Let $b_c = b_c + 1$, $C(i) = B_1$, and $C(b_c) = B_2$.
7. Let $I_1 = \{i : C(i)$ contains instances from two different classes at least$\}$, and let $I_2 = \{i : i \leq b_c\} - I_1$.
8. Let $I = I_1$ if $I_1 \neq \emptyset$ else $I = I_2$.
9. Find the two farthest points, $q_1(i)$ and $q_2(i)$, in each $C(i)$ for $i \in I$.
10. Find the set $C(j)$ with the largest diameter.
11. Let $B = C(j)$, $p_1 = q_1(j)$, and $p_2 = q_2(j)$.
12. If $b_c < b$ then go to Step 5.
13. Find the centroids $c(l, i)$ for each class l in subset $C(i)$, $i = 1, 2, \ldots, b$.
14. Put the $c(l, i)$ in the resulting set S.

For this approach, the number of partitions is given, which results in a number of instances greater or equal to the number of subsets and less or equal to the number of partitions multiplied by the number of classes, that is, $b \leq |S| \leq b \cdot m$, where b is the number of subsets and m is the number of classes in the TS.

3.2 IRSP2

Theory dictates that cases from a class would be as close to each other as possible, while instances from different classes would be located as far as possible. Therefore, it seems more appropriate to split the subset by the highest degree of overlapping among instances from different classes. Another difference regarding IRSP1 is that IRSP2 generates only one prototype for each subset and assigns it to the class with a majority of cases.

The overlapping degree of a set A, say O_A, can be defined as the ratio of the average distance between instances belonging to different classes, D_d, and the average distance between instances being from the same class, D_s.

1. IRSP2(T, n):S
2. Let $n_c = 1$ (n_c is the current number of subsets in T), and $i = 1$.
3. Let $B = T$.
4. Find the two farthest points, p_1 and p_2, in B.
5. Divide the set B into two subsets B_1 and B_2, where
   $B_1 = \{p \in B : d(p, p_1) \leq d(p, p_2)\}$
   $B_2 = \{p \in B : d(p, p_2) < d(p, p_1)\}$
6. Let $n_c = n_c + 1$, $C(i) = B_1$, and $C(n_c) = B_2$.
7. Let $I_1 = \{i : C(i)$ contains instances from two different classes at least$\}$, and let $I_2 = \{i : i \leq n_c\} - I_1$.
8. Let $I = I_1$ if $I_1 \neq \emptyset$ else $I = I_2$.
9. Find the set $C(j)$ with the highest overlapping degree, $O_j = \frac{D_d}{D_s}$.
10. Find the two farthest points, $q_1(j)$ and $q_2(j)$, in $C(j)$.
11. Let $B = C(j)$.
12. If $n_c < n$ then go to Step 5.
13. Find the centroids $c(i)$ for each subset $C(i)$, $i = 1, 2, \ldots, n$.
14. Put the $c(i)$ in the resulting set S.

By applying the IRSP2 algorithm, the number of instances in S will be bounded by the number of problem classes and the number of instances given in advance, say n: $m \leq |S| \leq n$.

3.3 IRSP3

IRSP3 corresponds to a combination of IRSP1 and IRSP2. The subset divided at each stage is that with the highest overlapping degree and, one prototype for each different class existing in the resulting subsets is computed. This helps to avoid the possible excessive displacement of the decision boundaries when replacing subsets with a high overlapping degree by only one centroid. The input to the algorithm is the number of partitions to perform and consequently, the number of resulting cases will be the same as that of IRSP1.

3.4 IRSP4

The last reduction heuristic consists in performing partitions until all subsets are homogeneous (no subset contains a mixture of instances from different classes). It is not necessary to provide any tuning parameter (number of partitions or number of instances) to the algorithm. IRSP4 can use both division criteria defined previously. In fact, the partition criterion is not important here because all heterogeneous subsets have to be finally divided.

4 Experimental Results

The schemes introduced in Sect. 3 have been empirically compared with Wilson's, Hart's, and Chen's algorithms. The basic 1-NN rule with 100% of training cases has also been included here for comparison purposes. We have utilized six data sets from the UCI Repository [7], and three from the ELENA Project. We have worked only with domains where all attributes are continuous.

Holdout averaged over five random partitions has been used for each database. A percentage of the instances has been used as the TS and the rest of cases for the test set. Experiments consist in applying the 1-NN rule to the test sets, where the training portion has been preprocessed by some reduction algorithm.

Table 1. Classification accuracy and reduction percentage for each data set.

Data set	1-NN	Wilson's	Hart's	Chen's	IRSP1	IRSP2	IRSP3	IRSP4
Cancer	94.53	94.89	91.24	95.01	94.53	95.13	94.28	94.28
		3.02	96.70	99.01	98.72	99.01	98.63	96.89
Clouds	84.60	88.42	88.08	58.22	65.61	58.87	68.35	85.52
		12.20	94.76	99.79	99.73	99.79	99.68	74.58
Glass	72.50	66.25	67.50	37.50	50.83	41.67	54.17	63.75
		35.63	87.36	96.93	95.89	96.93	95.08	69.83
Heart	59.26	64.81	66.67	63.27	66.67	65.74	64.81	62.04
		36.11	86.11	97.53	96.72	97.53	96.29	74.07
Liver	68.12	70.29	63.77	57.00	61.11	57.00	60.63	67.39
		36.23	80.80	97.74	96.68	97.74	96.56	64.86
Phoneme	76.08	72.95	70.14	65.81	65.68	68.37	68.07	72.19
		43.55	79.12	99.68	99.66	99.68	99.69	82.93
Pima	63.40	68.96	67.05	65.47	67.87	66.23	70.59	69.94
		29.84	84.36	99.13	98.43	99.13	98.59	81.14
Satimage	79.98	79.67	78.24	64.15	75.41	64.00	76.01	78.82
		6.06	77.38	99.53	99.55	99.53	99.59	75.02
Wine	73.53	73.54	71.39	65.69	67.16	65.69	66.18	73.53
		30.21	83.91	96.20	95.75	96.20	95.63	87.15
Average	74.67	75.53	73.79	63.57	68.32	64.74	69.23	74.16
		25.87	85.61	98.39	97.90	98.39	97.75	78.49

Several comments can be made from the results in Table 1. As expected, 1-NN and Wilson's algorithms present the highest average accuracy, but it is mainly due to retaining all or most of the instances (in average, Wilson's algorithm only removes 25.87% of the cases). However, in general, all IRSP methods achieve higher accuracy than Chen's scheme, with practically the same reduction rate. Hart's condensing results are quite similar to those of IRSP4: a sufficiently high reduction degree without an important degradation in accuracy.

Among the algorithms proposed in this paper, IRSP1, IRSP3 and IRSP4 seem to be the best in terms of balancing accuracy with storage reduction. IRSP4

shows the highest classification accuracy and it still removes many instances (78.49% in average). When comparing this approach with Wilson's editing, we can see that IRSP4 achieves an accuracy close enough to that of Wilson's, but with a very important difference in the reduction percentage (52.62% higher in average). On the other hand, IRSP1 and IRSP3 eliminate close to 98% of the cases and the average accuracy is over 68%.

It can be observed that IRSP2 and Chen's algorithms achieve worse accuracy results than the other schemes proposed. It seems that the number of centroids computed for each subset can become more important than the partition criterion used by the algorithm. In fact, in many cases, when applying Chen's method or IRSP2 (i.e., those schemes that compute only one prototype in each subset), all the instances belonging to some classes have been removed from the TS.

5 Concluding Remarks

This paper has focused on reduction techniques for lazy learning algorithms based on generating new prototypes from the original training cases. Four new schemes have been introduced by using the concept of space partitioning. Two different division criteria have been analyzed, along with two distinct manners to create the new prototypes in each subset.

From the experiments, it seems that generating only one prototype for each subset gives lower accuracy than creating one for each class present in each partition. The split criterion affects accuracy, but much less than the method used for generating prototypes. Accordingly, IRSP1 and IRPS3 show the highest accuracy with a very small number of instances. Secondly, IRSP2 has the advantage that it allows to control the number of resulting cases and, it still obtains higher accuracy than Chen's algorithm. Finally, IRSP4 achieves the highest accuracy, although it also creates more prototypes than the other schemes.

References

1. Aha, D.W., Kibler, D., Albert, M.K.: Instance-based learning algorithms. Machine Learning **6** (1991) 37-66.
2. Aha, D.W.: Tolerating noisy, irrelevant and novel attirbutes in instance-based learning algorithms. Int. Journal of Man-Machine Studies **36** (1992) 267-287.
3. Chang, C.-L.: Finding prototypes for nearest neighbor classifiers. IEEE Trans. on Computers **23** (1974) 1179-1184.
4. Chen, C.H., Józwik, A.: A sample set condensation algorithm for the class sensitive artificial neural network. Pattern Recognition Letters **17** (1996) 819-823.
5. Cover, T.M., Hart, P.E.: Nearest neighbor pattern classification. IEEE Trans. on Information Theory **13** (1967) 21-27.
6. Hart, P.: The condensed nearest neighbor rule. IEEE Trans. on Information Theory **14** (1968) 505-516.
7. Merz, C.J., Murphy, P.M.: UCI Repository of Machine Learning Databases. Univ. of California, Irvine. http://www.ics.uci.edu/mlearn (1998).
8. Wilson, D.L: Asymptotic properties of nearest neighbor rules using edited data sets. IEEE Trans. on Systems, Man and Cybernetics **2** (1972) 408-421.

Describing Decision Support, Data Mining, and Text/Web Mining Studies in SolEuNet

Marko Bohanec, Bojan Cestnik, Marko Grobelnik, Dunja Mladenić
Jožef Stefan Institute, Ljubljana, Slovenia

Mário Alves, Alípio Jorge
LIACC, Oporto, Portugal

Steve Moyle
Oxford University, Oxford, UK

Abstract. We present a schema for documenting and classifying completed Data Mining, Decision Support and Text and Web Mining cases. Project descriptions from these areas are unified in a hierachically structured relational database. The main objectives and benefits of the repository are presented and discussed.

1 Introduction

Working with end-user problems usually implies that most of the results cannot be published even though the experts performing the data analysis have learned general lessons that can be potentially useful when approaching other end-user problems. That kind of experience is usually related to specific information about the problem characteristics and the used methodology; it can be shared without revealing confidential information about the problem and the customer. In our work on developing prototype solutions for customer problems within project SolEuNet (Mladenić, 2001), we aim not only at solving end-user problems (e.g., Cestnik and Bohanec, 2001), but also at developing new methods for collaborative data mining (Jorge, et al., 2002), combining problem solutions as well as combining data mining and decision support with information systems. The idea is to work on prototype solutions that have a potential for later commercial exploitation, and also to analyse failed and successful approaches using a joint infrastructure, education and dissemination. So, one of the main objectives is, based on the experience and lessons learned from practical cases, to propose a compact description of the cases in the form of a repository.

Among several benefits that are expected as a result of having the past projects stored in a repository, we emphasize the following ones:

- Unified project documentation;
- Stored knowledge and experience that could facilitate learning about the stored cases as well as replicating the successful solutions on similar new problems;

- Fast search among the end-user projects by using descriptive criteria (when the repository gets implemented in the form of a database);
- Summarized lessons learned from similar end-user problems, which might help avoiding obstacles when facing new problems.

The following section describes typical categories and examples of projects approached within SolEuNet. Section 4 presents a unified project description schema, designed as a flexible relational data base.

2 SolEuNet end-user projects

End-user projects, approached within SolEuNet, belong to three different areas: (1) Decision Support, (2) Data Mining, and (3) Text and Web Mining.

Decision Support (DS). In SolEuNet, DS has been mostly based on qualitative hierarchical multi-attribute modeling, using the supporting computer programs DEX and DEXi (Bohanec and Rajkovič, 1990; Bohanec, 2002). Six different DS projects have been approached and completed. One of them, *Housing* (Bohanec et al., 2002), was aimed at supporting the task of housing loan allocation for the reconstruction of denationalised buildings in the city of Ljubljana. Two multi-attribute models have been developed and used for this purpose. The characteristics of this project – already using the unified schema proposed in section 3 – are shown in Table 1.

Prior to SolEuNet, the completed DS projects had been documented in various ways. While some of them produced a written text report and/or some form of schematic description (Urbančič, et al., 1998), others were mostly documented with printouts from DEX and DEXi, and some outstanding projects were described as practical cases in scientific papers (e.g., Bohanec, et al., 1996).

Data Mining (DM). An example of a SolEuNet DM project is *Mediana* (Škrjanc, et al., 2001), where different data mining methods were used for the analysis of the media space in Slovenia. A media space consists of many different factors fighting for the attention of the customer population in some environment. We have analyzed data describing the entire media space of the whole country (Slovenia) with the population of 2 million people. The data were collected by the private research institute Mediana. The database consists of 8000 questionnaires, each containing 1200 questions, gathered in 1998.

Text and Web Mining. The problem of this kind comes from the Portuguese Institute of Statistics (*INE*), the governmental agency which is the keeper of national statistics. INE has the task of monitoring inflation, cost-of-living, demographic trends, and other important indicators. Its goal was to get information and on this basis provide better services on Infoline (www.ine.pt), a web site that makes statistical data available to the Portuguese citizens. The specific task was to extract knowledge from the web site's access data log, using DM techniques such as association rules, clustering and classification (Jorge and Moyle, 2002; Alves and Jorge, 2002). Association rules, for instance, can tell what is the next page a user would like to see, and help them finding the information they are looking for. This ability of "guessing" the user's wishes can

be provided to the site by analyzing the usage of the site by other users, and discovering their own preferences. Also, the technique of clustering can, from the same stream of data, discover natural groups of users with similar preferences and behavior. This knowledge can help improve the usability of the site. Data collection is nearly costless, but the patterns found in the data can help the Portuguese save thousands of hours in their quest for statistical data.

Initially, several project description schemas for these specific areas have been designed by different SolEuNet workpackages (Mladenić, 2002). For instance, a description schema for DS projects was proposed in (Cestnik and Bohanec, 2002). A different schema was used for INE (Jorge and Moyle, 2002). Almost independently, the SolEuNet Information Collector (SENIC) database has been developed as a web system designed to support the task of collecting information about tools and case studies in SolEuNet. SENIC was engineered with the reliable web technologies described in (Alves, 2001). Although designed as a general repository, SENIC has been found more appropriate for describing DM than DS projects, clearly exposing the need for a unified project description schema.

3 Unified project description

The unified approach to describing Data, Text, and Web Mining and Decision Support solutions of completed end-user projects draws on two facts. First, these projects share a considerable number of common characteristics, which can be used for all of them. Second, project descriptors can be layered in order to cope with the specifics of approaches and applied methods in different areas.

This leads to a hierarchically organized relational database in which, at the top level, a project description is divided into three categories: (A) general description, (B) problem description and (C) method-specific parameters. This division is rather natural: first, a project is described in general, regardless of the specific type of the project and applied methods. Then, the specific problem is elaborated in more detail, using descriptors that are specific for the taken approach, such as DM or DS. Finally, method-specific parameters are presented on the third level.

Each higher-level category can contain one or more lower-level categories. For example, consider a hypothetical project, whose general characteristic can be described by descriptors of the category A. Suppose it is a DS project; in this case, the description can be supplemented by DS-specific parameters B. The problem can be approached by one or more different DS methods (C), for instance by two qualitative multi-attribute models (C1 and C2), a quantitative multi-attribute model (C3) and decision trees (C4). In addition, the same project (A) may have some data available, which can be analyzed by DM techniques. So, this is also a DM project and can be described by DM-specific parameters (say, B2). Again, several methods can be used for DM, such as association rules (B2.C1) and clustering (B2.C2).

Table 1. Project Housing described by the unified schema.

A. General			
Project acronym			Housing
Project title			Loan allocation for the Housing Fund of Ljubljana
Keywords			Loan allocation, housing
Business sector			Finance
End-user mission			Housing, mortgage market
Customer institution			The Housing Fund of Ljubljana Municipality
Location			Ljubljana, Slovenia
Involved SolEuNet partners			Temida, IJS
Other partners			None
Start date			January 2000
End date			September 2001
Time span			9 months
Expert team size			5
Expert resources			14 MM
Press release			*text describing the project (omitted)*
Summary			Decision support of a tender for renovating old Denationalized blocks of flats in Ljubljana
B. DS Problem Description			
Background	Problem acronym		Housing
	Problem title		Loan allocation
	Business success criteria		Undefined
	Internal champion		Not available
	Problem owner(s) accessible		Yes
Problem style	Problem type		Two-time
	Problem structure		Semi-structured
	Problem definition		Medium
	Organizational level		Tactical/strategic, management involved
	Supporting methods		Modelling, qualitative ranking/evaluation models, computational models, database, what-if analysis
	Primary DS elements		Data, models
	Group decision problem		No (no different interests)
Team members	Problem owner		1
	Additional experts		1
	Decision analysts		3
	Users		0
	Others		0
C. Method-specific parameters			
		C1.	**C2.**
Method type		Qualitative multi-attribute model	Qualitative multi-attribute model
Model name		A	B
Model description		Priority ranking of applicants that own only one flat in which they reside (the flat must be in a denationalised block)	Priority ranking of applicants that own also some other flats (in the denationalised block) that are rented non-profitably
Tools used		DEX	DEX
Size	Basic attributes	10	6
	Aggregate attributes	7	4
	Ranks	5	5
Number of options		109	258

Thus, this hypothetical project can be described by the following instance of the unified schema:

A: General description (Project acronym, Title, Keywords, …)

- B1: DS Problem description: Background, Problem style, Evaluation
  - C1: First DS qualitative multi-attribute model
  - C2: First DS qualitative multi-attribute model
  - C3: DS quantitative multi-attribute model
  - C4: DS decision tree
- B2: DM Problem description: Background, Problem style, Evaluation
  - C1: DM association rules
  - C2: DM clustering

Organized in this way, the schema is highly flexible. First, it facilitates the description of projects that are approached by a variety of different approaches and methods. Second, it can be easily extended by new sets of descriptors corresponding to new types of problems (B) or new methods (C).

For the illustration of specific descriptors, the DS project *Housing* is described by this schema in Table 1. Notice that the descriptors in section A are standardized and equal for all projects. Section B is specific to DS projects, but equal for all of them. Section C contains two descriptions, C1 and C2, each corresponding to one of the multi-attribute models developed in the project.

4 Conclusion and further work

The main goal of this work was to propose a unified schematic description of completed end-user cases that can serve as a basis for the repository. The repository is one of the prerequisites for promoting and extending exploitation of Data Mining, Decision Support and Web/Text Mining technology into practice.

There are several benefits of having the past projects stored in a repository. First, the stored projects are documented in a similar formal way; as a result, it is relatively easy to get information about a single project as well as to mutually compare two or more projects. Second, stored knowledge and experience in the repository facilitate the discovery and learning about the recorded cases as well as replicating the successful solutions in similar new problems. Next, when the repository gets implemented in the form of a database, it will facilitate fast searching among the stored projects by using descriptive criteria. Last but not least, one can gain access to summarised lessons learned from similar problems, which might help avoiding obstacles when facing new problems.

The proposed project description schema is highly flexible. Its hierarchical structure facilitates the description of problems that are of different types and that are approached by a variety of methods. Also, it can be easily extended to new types of problems and methods used.

For further work we plan to implement the resulting repository schema as an object-oriented computer database, accessible through WWW, and include additional completed projects in the repository.

Acknowledgment

The work reported here was in part supported by EU project SolEuNet, IST-11495, and by the Slovenian Ministry of Education, Science and Sport.

References

Alves, M.A.: Safe Web Forms and XML Processing in Ada. In: Reliable Software Technologies: Ada-Europe 2001: Leuven, Belgium, May 14-18. Springer, LNCS 2043. 349–358.

Alves, M.A., and Jorge, A.: INE's Infoline Website Access Analysis (www.ine.pt) : CRIPS-DM Report. February 2002. SolEuNet working document published on Zeno (http://zeno.gmd.de/login) /SolEuNet /WP5 /RAMSYS /INE_Infoline /Phases (restricted).

Bohanec, M., Rajkovič, V.: DEX: An expert system shell for decision support. Sistemica, 1(1), (1990) 145–157.

Bohanec, M., Cestnik, B., Rajkovič, V.: A management decision support system for allocating housing loans. In: Humphreys, P, Bannon, L., McCosh, A., Migliarese, P., Pomerol, J.-C.(eds.): *Implementing Systems for Supporting Management Decisions*.London:Chapman and Hall (1996).

Bohanec, M.: DEX: An expert system shell for decision support. `http://www-ai.ijs.si/MarkoBohanec/dex.html` (2002).

Bohanec, M., Rajkovič, V., Cestnik, B.: *Report on Decision Support Practical Cases, Phase II*, Jožef Stefan Institute, Ljubljana, Report DP-8512 (2002).

Cestnik, B., Bohanec, M.: Decision support in housing loan allocation: A case study. In: Giraud-Carrier, C., Lavrač, N., Moyle, S., Kavšek, B. (eds.): *IDDM-2001: ECML/PKDD-2001 Workshop Integrating Aspects of Data Mining, Decision Support and Meta-Learning: Positions, Developments and Future Directions*, Freiburg (2001) 21–30.

Cestnik, B., Bohanec, M.: *SolEuNet Report on the repository of problem descriptions:D6.6* (2002).

Jorge, A., Moyle, S.: *Sol-Eu-Net WP5 Data Mining midterm* report: D5.3.2 (2002).

Jorge, A., Moyle, S., Voss, A.: Remote Collaborative Data Mining Through Online Knowledge Sharing. In: Camarinha-Matos, L.M. (ed.): Collaborative Business Ecosystems and Virtual Enterprises. Kluwer Academic Publishers, 2002.

Mladenić, D.: EU project: Data mining and decision support for business competitiveness: a European virtual enterprise (Sol-Eu-Net). In: D'Atri, A., Solvberg, A., Willcocks, L. (eds.). *OES-SEO 2001: Open enterprise solutions: Systems, experiences and organizations*. Rome, 14-15 September 2001. Roma: LUISS (2001) 172–173.

Mladenić, D.: Describing Data Mining and Decision Support Studies in SolEuNet. *Technical Report IJS-DP 8622*, J.Stefan Institute, Ljubljana, Slovenia, May 2002.

Škrjanc, M., Grobelnik, M., Zupanič, D.: Insights offered by data-mining when analyzing media space data. *Informatica* 25(3), (2001) 357–363.

Urbančič, T., Križman, V., Kononenko, I: *Review of AI Applications*, Jožef Stefan Institute, Ljubljana, Report DP-7806 (1998).

Data Mining for Decision Support in Marketing: A Case Study in Targeting a Marketing Campaign

Bojan Cestnik[1, 5], Nada Lavrač[1], Filip Železný [2],

Dragan Gamberger[3], Ljupčo Todorovski[1], Miro Kline[4]

[1] Jožef Stefan Institute, Ljubljana, Slovenia
[2] Czech Technical University, Prague, Czech Republic
[3] Rudjer Bošković Institute, Zagreb, Croatia
[4] Kline&Kline, Ljubljana, Slovenia
[5] Temida, Ljubljana, Slovenia

Abstract. The paper presents a case study in targeting a marketing campaign for a specific non-alcoholic beverage brand name, based on the results of completed questionnaires about brand name recognition in Slovenia. First, we give a motivation for the task. Then, we briefly describe the data and explain the preprocessing steps. In the main part of the paper we highlight the major steps in actionable knowledge generation form the studied task. The paper concludes with lessons learned and directions for further work.

1 Introduction

One of the most important tasks of a marketing expert is how to efficiently target a population for advertising a specific product or service (e.g. Berry and Linoff, 2000). Usually, there is a trade-off between the cost of communicating a message to everybody, on one hand, and a loss due to selecting a too narrow population segment, which may result in missing some of the potential customers, on the other hand. Therefore, the area is particularly suitable for applications of Data Mining (Škrjanc, et al., 2001) and Decision Support methods (Bohanec, et al., 2002; Cestnik and Bohanec, 2002).

Among the available tools that can help marketing experts accomplish their tasks, Data Mining tools have, since recently, gained significant importance. Namely, in most large companies the quantity as well as the quality of data about customers and orders has increased dramatically in the last decade. In addition, there are several global studies and reports about customer behavior published each year. Statistical tools that were successfully applied in the past to study global phenomena can no longer provide sufficient answers to specific, individually focused questions (Berry and Linoff, 2000).

In this paper we present a case study in targeting a marketing campaign for a Slovenian natural non-alcoholic sparkling beverage brand (since we are not allowed to uncover the identity of this brand, the brand name is labeled *X*). The study is based on the data that were collected from a public survey done by the marketing agency

Kline&Kline using dedicated software for questionnaires *QA* developed by the Temida company. More specifically, the task was to identify the characteristics of those consumers that do not yet know and/or use brand *X*. In reality, the target population is further segmented to (A) drinkers of other non-alcoholic sparkling beverages and (B) others that do not drink any beverages of this kind. The latter can, by all means, be excluded from our target since it is fair to assume that they are not inclined to use the product generically. In another words, in our campaign we would like to contact the users of competitive brands and present them with the qualities of new product *X*. However, the marketing expert in our team pointed out one additional constraint. There seems to be one particular brand (labeled brand *Y*) that is so firmly positioned on the market that it would be a wise idea to exclude also the users of this particular brand from our target population. In fact, it is reasonable to expect that the regular drinkers of brand *Y* are very unlikely to change their prevalent behavior no matter what our arguments about the brand *X* are.

The rest of the paper is organized as follows. The next section describes the steps required to accomplish the case study in a more detailed manner. A special emphasis is dedicated to data description, data preprocessing, and actionable knowledge generation. In conclusion we present some lessons learned and give directions for further work.

2 Data Mining for targeting a marketing campaign

This section first describes the data used for the study. Then it presents two major tasks that were accomplished within the study: data preprocessing and subgroup discovery aimed at actionable knowledge generation for the development of a marketing campaign. The marketing problem addressed in this paper is how to target a marketing campaign for a Slovenian natural non-alcoholic sparkling beverage brand. The starting point is a relational database obtained by interviewing potential customers. The targeting task is the problem of selecting potential customer subgroups that can be targeted by advertising campaigns.

Business to be successful have to view their markets as consisting of distinct groups of consumers, each with their own distinct set of requirements. True consumer segmentation has such a profound impact on a business that getting it right cannot be left to chance (McDonald and Dunbar, 2000). Literature on market segmentation underlines the view that markets and their segments are clusters of potential customers (Kotler 1991; Tynan and Drayton 1987). Only some of them are suitable to be selected and approached or targeted to pursue with right offers from a particular marketing agent.

Every segmentation yield several segments and the key question is how to help marketing planners to decide which ones are likely to be most promising. The rule of a thumb is to target the segment with the heaviest users. On surface this makes the most sense; however, there are some strong indications that such approach is not the most promising one (Myers 1996, p.21-22):

- One or more major competitors may have already targeted this group successfully;

- The company product line is not well designed for this group;
- In reality there are no heavy users;
- The company is to small to go after the heavy-user segment;
- The company and its agency want to develop different marketing campaign for its brand for all usage groups.

2.1 Data description

The input data for the targeting task was gathered as a result of a survey done by the Kline&Kline marketing agency about brand name recognition and reputation. The data consist of three relational tables: (1) general customer responses and demographic facts, (2) responses about specific brand names, (3) verification of brand names recognition.

The first table contains customer responses to general questions and demographic facts. Each customer is identified by unique key *Q*. There are 2013 rows (customers) in the table. The customers are described by their answers concerning age, level of education, occupation, area of living, consumer preferences and habits like what TV programs they watch and what newspapers they read regularly.

The second table contains responses about specific brand names. There are 300 different brand names analyzed in this survey; to avoid an overkill each customer is given a subset of 15 brand names to evaluate with respect to their recognition, reputation and usage. Therefore, the second table contains *Q* as a foreign key and *D* as a key for a specific brand. There are in total 30195 lines, representing 2013 * 15 answers.

The third table contains control questions that can be used to estimate the quality of answers in the second table. Here, additional questions about each brand name with respect to its product category are asked. For example, if a customer responds that he knows and uses a specific brand, and at the same time categorizes the brand in a wrong product category, one can reasonably conclude that the customer's knowledge about the brand is questionable. Either the brand is mistakenly mixed-up with some other brand or he simply made a mistake. For the specific task described in this paper we did not take the third table into account.

The most important attribute for our study is the frequency of consumption of a particular brand *D*. It is included in the second table and can have values from 1 to 5, 1 meaning that the customer does not know the brand (and therefore does not use it), and 5 meaning that he regularly uses it. For further analysis we took answers 4 and 5 as positive (uses the brand) and 1, 2 and 3 as negative.

The need for data preprocessing arose from the fact that the concept "drinker of brand *X*" can only be determined for a limited number of respondents: only to those that were originally asked this question. The same holds for two other concepts: "drinkers of brand *Y*" and "drinker of non-alcoholic sparkling beverage brands". Note that when combing the sparse concepts with logical operator AND one can quite easily end up with an empty set. Therefore, the concepts need to be stretched to all respondents in order to obtain a set large enough to produce reliable actionable descriptions.

2.2 Data preprocessing

Every respondent got to evaluate only 15 brands out of 300. This means that only one customer out of 20 was asked about the recognition and reputation of a specific brand. In order to combine several different concepts it is necessary to construct a classifier that can be used to fill in the data about using/non-using a specific brand for the rest of 19 customers out of 20. The process of data preprocessing is described in full detail in (Železný, et al., 2002).

In order to fill in the missing concept assignment, we used the CN2 algorithm (Clark and Niblett, 1989) to learn from the known training cases to produce classification rules that can be used to assign a class to other instances that could not initially be evaluated weather they belong to a given concept. If, for example, the concept is "drinker of brand *X*", then only the respondents that were asked about brand *X* can originally be included in or excluded from the concept. The learned CN2 rules were used to classify all the other respondents.

A common experience of data-mining efforts applied as well in achieving the mentioned goal: several tools had to be applied to obtain a useful result. Besides CN2, we employed the Sumatra Transormation Tool (Aubrecht, et al., 2002) and two Prolog programs, in the sequence described bellow.

Firstly, the transformation of responders' grading (1-5) into positive and negative examples of particular concepts was dictated by the principles given by the domain-expert and in some cases it was not trivial. For example, although the concepts of "X – drinker" and "Y – drinker" are straightforward (satisfied by persons who gave a response equal or greater than 3 to the question regarding the consumption of X or Y, respectively), the concept of "drinker of *other* sparkling drinks" was defined in a more complicated manner. Positive examples were people not asked about brand Y, who responded with confidence equal or grater than 3 for at least one sparkling drink. Negative were those asked about at least one sparkling drink excluding X and Y and the maximum of the answers to the questions on sparkling drinks was equal or smaller than 3. Positives and negatives are thus disjoint, but their union is a proper subset of all responders (not all responders qualify to be examples). Furthermore, the required notion of "other sparkling drink" is itself a pre-defined concept (among all drinks) as well.

To preserve clarity of such transformations, we decided to encode them declaratively in Prolog. The input data thus had to be transformed into Prolog facts and Sumatra TT provided this service.

The CN2 algorithm was then applied on the example file that was generated by the Prolog program. It then produced a set of rules of the form

```
Body → Class_Assignement
[N1 N2]
```

where `the Body` is a conjunction of attribute value assignments, where attributes are mostly demographic parameters of the responders, their preferred TV channels, journals etc. The numbers N1 (N2) define the quantity of positive (negative) examples complying with the rule's body conditions. An example of the rules is e.g.

```
IF journal_15 = 0 AND tv_1 = 1 AND tv_8 = 1 THEN class = 1
[13 0]
```

Such rules were easily retranslated into a format interpretable by a Prolog machine. Subsequently, to predict a response of a person in the range of 1 to 5, the distributions (*N1*'s and *N2*'s) are summed up for all rules whose bodies are satisfied by the attribute assignments of the given person into a cumulative distribution ($N1_{sum}$ and $N2_{sum}$). Then the probability P of that person satisfying the given concept (being a consumer of the drink) can be estimated as

$$P = N1_{sum} / (N1_{sum} + N2_{sum})$$

and this value was used to calculate the most likely response R in the original range of 1 – 5, by a linear projection

$$R = 1 + 4\,P\,.$$

In order to assess the expected error of the CN2 induced concepts, we primarily (for evaluation purposes only) split each of the three concepts into 70% of training and 30% of testing data. For obtaining the final concept descriptions, we, however, used 100% of the data for training. The results are presented in Table 1. Although the accuracies on the testing sets are only slightly higher than the majority vote accuracy, it should be noted that this kind of measurement treats the prediction task as pure binary classification, whereas in fact CN2 algorithm produces a probabilistic classification. By measuring the average squared error on probabilities, we obtained the figures in Table 2, which are more favorable for the induced models. Note that the average square error is measured on training examples with known outcomes.

Table 1: Classification accuracy of the induced concepts

Method	Brand X	Brand Y	Other sparkling
Majority vote	65.8	81.0	57.5
Training accuracy	95.1	92.4	83.2
Testing accuracy	66.7	84.8	59.5

Table 2: Average squared error of the induced concepts on the training examples

Method	Brand X	Brand Y	Other sparkling
Random guess	3.64	3.26	3.55
Majority vote	2.76	1.78	3.03
Induced concept	1.02	0.76	1.29

2.3 Discovering important factors for Decision Support

By data preprocessing we managed to obtain the data set that is suitable for further investigation of the concept under study. From Tables 1 and 2 one can observe that the classifications are not performed with high statistical significance. This is mostly due to the fact that the inherent nature of the domain (targeting a population in marketing) is inexact and probabilistic. For example, it makes no sense saying that all the readers of a specific newspaper will buy a certain product; however, it might be fair to conclude that the probability of them buying the product is higher than average.

Indeed, instead of trying to describe the final concept with a rule, we find it more beneficial to present it by listing its supporting factors as well as its opposing factors, which follows the basic principles of Bayesian analysis (e.g. Berger, 1985). These factors were found in such way that they respectively maximize or minimize the conditional probability of the concept. Only the factors with statistical significance higher than 99% were selected as influential and were included in the listings. The marketing expert found such descriptions very intuitive and easy to apply in practice, especially in the cases where the corresponding group can be named with a suitable metaphor. It seems that such a disjunctive approach is particularly suitable in marketing (and possibly related domains), where the task is to increase the probability of a certain event (order, buy, reply) in a target population and not to accurately describe a portion of the target population.

In our case, we first have the concept of "drinker of brand *X*". This group can be characterized by the following supporting factors:

- The customers come from a central Slovenian region,
- Label "Monitored food" is neither important nor unimportant,
- They regularly read *Dnevnik* and/or *Mladina*, and
- Their education degree is higher or equal to university degree.

On the other hand, the non-users of brand *X* can be characterized as follows:

- Availability of a product in different quantities is not important, and
- Product price is not so important.

The next concept is "user of non-alcoholic sparkling drinks". The members of this concept can be characterized with the following descriptions:

- They read regularly *Finance* and *Večer,*
- Their education level if higher or equal to university degree,
- Healthy food is important,
- They watch regularly *Gajba-TV* (local TV station),
- Availability of a product with different tastes is not important, and
- Nice product outfit is important.

In contrast, the opposing factors for the above concept are the following:

- The read regularly *Naš dom*, *Mag*, *Nedeljski dnevnik* or *Gea,*
- They do not watch *POP-TV,*
- Good commercials are not important,
- Their age is 61 and over.

The last simple concept is "user of brand *Y*". We found the following supporting factors:

- They are younger than 20 years,
- They read regularly *Nedeljski dnevnik,*
- Adequate brand name is very important,
- Availability of a product in different quantities is important,
- They have a free profession (lawyer, architect, artist, …),
- Healthy food is important,
- They watch *TV3*, and
- Good commercials are very important.

The non-users of brand *Y* are characterized by the following factor:

- Good commercials are not important.

Here, let us restate our target population: they are the ones that do not yet know or use the brand *X*, but do drink other non-alcoholic drinks, with the exception of those that regularly drink brand *Y*. To describe it, one might use the combination of the above supporting factors; however, since we stretched the concept in the data pre-processing phase (by learning labels for missing concepts), we can find the following supporting factors for the combined concept:

- Availability of a product in different quantities is not important,
- Good commercials are not important,
- Different tastes of a product are not important,
- Good name of a product is not important,
- Popularity of a product is not important, and
- They read *Večer* regularly.

Here is the list of factors against the combined concept:

- Good commercials are important,
- They read *Dnevnik*, *Nedeljski dnevnik*, *Mladina* and/or *Naš dom*,
- Good name of a product is important,
- They regularly read more than 4 newspapers,
- They are from central Slovenian region, and
- Their education level is higher or equal to university degree.

One important observation to be made is that the supporting factors of the combined concept are not necessarily included in the basic concepts. For example, the concept of reading more than 4 newspapers did not appear in any of the basic concept descriptions. However, there are also some strong factors that can be traced from the combined concept to the basic one. For instance, the consumers from the central Slovenian region tend to be excluded from the combined concept, because they tend to be more then average consumers of brand *X*.

Note that the descriptive factors differ in how actionable they really are. If the description includes readers of a specific newspaper, the information can be used for targeting whereas there is not much that can be changed about the target audience of the newspaper, provided that you are not the editor in chief. Also, if one of the characteristics of the target population is that they don't value good commercials, you can't

reach them by making bad commercials. On the other hand, if they think that healthy food is important or that the nice product outfit is important, you can address their need by stating the healthy ingredients of your product or introducing its nicer outfit.

When describing subgroups with a set of influential factors it is important to be able to substitute the factors with a proper metaphor. For example, the first five factors from the description of the target population can be, according to the marketing expert, formulated as store-brand consumers. These consumers do not buy established popular brands. They settle for no-brand products that are usually sold under the store brand name, are packed in simple packages and offer good quality for a reasonable price. Such consumers can be addressed by low-profile advertising. According to the marketing expert the discovery of this piece of knowledge is substantial for the marketing analyst when planning and directing a marketing campaign.

3 Conclusions and lessons learned

In symbolic predictive induction, two most common approaches are rule learning and decision tree learning. The goal of rule learning is to generate separate models, one for each class, inducing class characteristics in terms of class properties occurring in the description of examples. Classification rule learning produces characteristic descriptions that are usually generated for each class by repeatedly applying the covering algorithm. In decision tree learning, on the other hand, the rules that can be formed of paths leading from the root node to class labels in the leaves represent discriminating descriptions, formed of properties that best discriminate between the classes. Therefore, classification rules serve two different purposes: characterization and discrimination. They form actionable knowledge, when the action to be performed is classification and/or prediction. This means actionability just in terms of determining class membership of individual non-labeled instances, and not necessarily uncovering the properties of population that can guide a decision maker in directing a targeting campaign.

In a marketing campaign targeting potential clients of a natural non-alcoholic sparkling drink, the target class are people who do not use or know this brand, but do drink other non-alcoholic drinks, except of those who regularly drink beverages of world-famous brands. Why should consumers of world-famous brands be excluded from the target? According to the marketing expert, these consumers are very unlikely to change their habits; therefore it makes no sense to direct a marketing campaign at these consumers. Moreover, in the discussion with a marketing expert it became clear that the negative class should not be formed of all the other consumers. Limiting the population to non-alcohol drinkers makes more sense in uncovering specific properties of the target population. If, for example, alcohol drinkers were to be included in class negative, the subtle differences between people who don't use the Slovenian brand, but do drink other non-alcoholic drinks would be hidden by much stronger regularities discriminating non-alcohol drinkers to those drinking alcohol drinks. Note that even subtler properties could be uncovered if the entire population were limited to

consumers of non-alcoholic dark-colored sparkling drinks, since the color of the analyzed brand is dark.

In the marketing problems where the task is to find significant characteristics of customer subgroups who do not know a brand compared to the characteristics of the population that recognizes the brand, one of the lessons learned is that the ROC space (Flach and Gamberger, 2001) is very appropriate for the comparison of induced models. Only subgroups lying on the convex hull may be optimal solutions and all other subgroups can be immediately discarded. When concrete parameters of the mailing campaign are known, like marginal cost per mailing and the size of the population, they define the slope of the lines with equal profit in the ROC space. Movements in the ROC space along these lines will not change the amount of the total profit, while movements upward or downward will increase of decrease the profit, respectively. The optimal subgroup in a concrete marketing situation is the point on the convex hull that has an equal profit line as its tangent. Additionally, in the direct marketing problem it was detected those optimal subgroups may be combinations of induced subgroups. In order to make use of this possibility, we have induced many potentially good solutions by changing the generalization parameters. In the problem of targeting a marketing campaign for a Slovenian natural non-alcoholic sparkling drink brand, most of already described techniques have been used; additionally, much effort was spent on data preparation.

One of the main requirements for successful application of Data Mining methods in marketing is that the learned concepts are actionable. This is, however, in most cases hard to achieve. If, for example, the learned concept includes customers of a certain age and living in a certain area, there is not much to act about. The only thing one can do is to take it into account when targeting the commercial message. On the other hand, if the learned concept includes customers that were sent a promotional material, then we can actively enlarge the coverage of the concept by sending some additional catalogs.

When describing subgroups with a set of influential factors it is important to be able to substitute the factors with proper a metaphor. For example, the first five factors from the description of the last target concept in section 2.3 can be formulated as store-brand consumers. Those are the consumers that do not buy established popular brands; instead, they settle for no-brand products that are usually sold under the store brand name, are packed in simple packages and offer good quality for a reasonable price. In marketing such consumers can be addressed by low-profile advertising. Although this conclusion is relatively simple and seems rather strait forward, it offers a crucial leverage to the marketing analyst in planning a marketing campaign.

For further work we envisage some more replications of the principle described in this paper on different marketing problem areas. In these new cases we plan to specifically monitor relations between different levels of actionable knowledge and its influence to potential use in practice.

Acknowledgment

The work reported here was in part supported by EU project SolEuNet, IST-11495, and by the Slovenian Ministry of Education, Science and Sport.

References

Aubrecht P., Železný F., Mikšovský P., Štěpánková O.: SumatraTT: Towards a Universal Data Preprocessor. *Proc. 16th European Meeting on Cybernetics and System Research*, vol. 2, p. 818-823, Austrian Society for Cybernetics Studies, ISBN 3-85206-160-1, 2002

Berger J.O., *Statistical Decision Theory and Bayesian Analysis*, Springer-Verlag, 1985.

Berry M.J.A., Linoff G.S.: *Mastering Data Mining, The Art and Science of Customer Relationship Management*, Wiley, 2000.

Bohanec M., Rajkovič V., Cestnik B.: *Report on Decision Support Practical Cases, Phase II*, Jožef Stefan Institute, Ljubljana, Report DP-8512 (2002).

Cestnik B., Bohanec M.: *SolEuNet Report on the repository of problem descriptions*. SolEuNet Report D6.6 (2002).

Clark P., Niblett T.: The CN2 induction algorithm. *Machine learning*, 3(4):261-283, 1989.

Flach P., Gamberger D.: Subgroup evaluation and decision support for direct mailing problem. *Integrating Aspects of Data Mining, Decision Support and Meta-Learning* Workshop at ECML/PKDD 2001 Conference.

Kotler P.: *Marketing Management: Analysis, planning and control*, Prentice-Hall, Englewood Cliffs, 1991.

McDonald M., Dunbar I.: Using structured processes and systems to help managers develop strategic segmentation, *Journal of Targeting, Measurement and Analysis for Marketing*, Vol. 9, 2, 109-127, 2000.

Myers J.H.: Segmentation and Positioning for Strategic Marketing Decisions, American Marketing Association, Chicago, 1996.

Škrjanc M., Grobelnik M., Zupanič D.: Insights offered by data-mining when analyzing media space data. *Informatica* 25(3), (2001) 357–363.

Tynan A.C., Drayton J.: Market segmentation, *Journal of Marketing Management*, Vol. 2, 3, 301-335, 1987.

Urbančič T., Križman V., Kononenko I.: *Review of AI Applications*, Jožef Stefan Institute, Ljubljana, Report DP-7806 (1998).

Železný F., Gamberger D., Todorovski L.: *Data Processing for a Marketing Application*, Jožef Stefan Institute, Ljubljana, Report DP-8576 (2002).

Subgroup Visualization: A Method and Application to Population Screening

Dragan Gamberger[1], Nada Lavrač[2], Dietrich Wettschereck[3]

[1] Rudjer Bošković Institute, Bijenička 54, 10000 Zagreb, Croatia
dragan.gamberger@irb.hr
[2] Jožef Stefan Institute, Jamova 39, 1000 Ljubljana, Slovenia
nada.lavrac@ijs.si
[3] University of Applied Sciences, Bonn-Rhein-Sieg, 53757 Sankt Augustin, Germany
dietrich.wettschereck@fh-bonn-rhein-sieg.de

Abstract. The paper presents a method for the visualization of subgroups, detected by a subgroup discovery algorithm. The main advantage and novelty of the method is that the visualized models can be used to illustrate the distributions of detected groups in terms of the percentages of true positive and false positive cases covered by the model.

1 INTRODUCTION

A subgroup discovery task can be defined as follows: given a population of individuals and a property of those individuals we are interested in, find population subgroups that are statistically 'most interesting', e.g., are as large as possible and have the most unusual statistical (distributional) characteristics with respect to the property of interest. An example subgroup discovery system is MIDOS [10].

Some approaches to association rule induction can be used for subgroup discovery. For instance, the APRIORI-C algorithm [4], adapting the association rule induction algorithm to classification rule induction, outputs classification rules with guaranteed high support and confidence. As such, each APRIORI-C rule represents a 'chunk' of knowledge about the problem, which is very important for knowledge discovery. In this paper, subgroups were discovered by a new heuristic rule learning algorithm [3]. The actual subgroup discovery algorithm is implemented in the on-line Data Mining Server, available at http://dms.irb.hr, whose description is out of the scope of this paper. The problem of population screening for early detection of atherosclerotic coronary heart disease (CHD) risk groups is used to illustrate the visualization of results obtained by applying our subgroup discovery methodology. To this end, the application of our subgroup discovery algorithm resulted in five models of patients with CHD risk which can be used for population screening.

The paper presents the detected risk groups and discusses approaches for their visualization (Section 2). A novel method for visualization of distributions of subgroups is described in Section 3, including a short review of related work.

2 VISUALIZATION OF CHD RISK GROUPS

Some interesting models of groups of CHD patients were constructed using the methodology of descriptive induction, using the available patient data, collected at the Institute for Cardiovascular Prevention and Rehabilitation, Zagreb, Croatia. There are three typical stages in the risk factor screening process, denoted by A, B, and C [7]. Our goal was to construct at least one model for every stage. Table 1 presents five induced models.

	Principal Factors	**Supporting Factors**
A1	positive family history age over 46 year	psychosocial stress cigarette smoking hypertension overweigth
A2	body mass index over 25 kgm^{-2} age over 63 years	positive family history hypertension slightly increased LDL cholesterol normal but decreased HDL cholesterol
B1	total cholesterol over 6.1 $mmolL^{-1}$ age over 53 years	increased triglycerides value
B2	total cholesterol over 5.6 $mmolL^{-1}$ fibrinogen over 3.7 $mmolL^{-1}$	positive family history
C1	left ventricular hypertrophy	positive family history hypertension diabetes mellitus

Table 1. Induced subgroup descriptions (principal factors) and their statistical characterizations (supporting factors). Subgroup A1 is for males, subgroup A2 for females, while subgroups B1, B2, and C1 are for both genders.

Figure 1 displays the respective coverages of subgroups A1, A2, B1, B2, and C1 in box plots. The figure shows the following information: the size of each subgroup, how it compares to the entire population and the distribution of the target values within each subgroup. Experience gained from working with non-technical end-users has shown that a pie chart visualization is more appealing to these users because they more closely resemble business charts. Pie charts, however, often mislead the perception of the user due to difficulties with relating the size of pie slices to actual values. Hence, the visualization with boxes is preferred. While these figures are more difficult to understand when first encountered, they allow for better comparison of the different subgroups and clearly display the size of each subgroup. This visualization technique can serve as an entry point to the more in depth visualization technique introduced in this paper.

3 VISUALIZATION OF SUBGROUP DISTRIBUTIONS

Data visualization methods have been part of statistics and data analysis research for many years. This research concentrated primarily on plotting one or

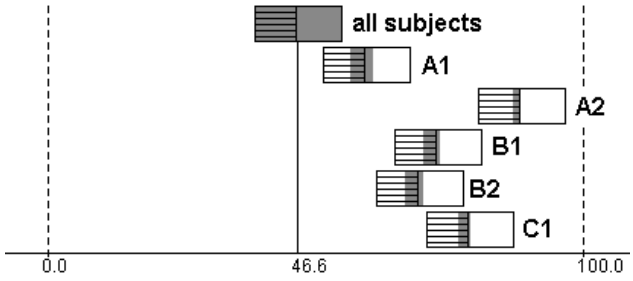


Fig. 1. Visualization by box plots. Each subgroup is represented in one box plot (all studied subjects are also considered one subgroup and are displayed in the top box). Each box shows the entire population. The gray area within each box indicates the respective subgroup. The overlap of the gray area with the hatched area shows the overlap of the group with the target (CHD). Hence, the farther to the left a gray area extends, the larger the overlap with the target (coverage). The lesser the gray area extends to the right of the hatched area, the more specific a subgroup is (less overlap with the non-target subjects). Finally, the location of the box along the X-axis indicates the relative share of the target CHD within each subgroup: the farther to the right a box is placed, the higher is the share of the target value within this subgroup. The line at 46.6% indicates default accuracy, i.e. the number of patients with CHD in the entire population.

more independent variables against a dependent variable in support of explorative data analysis [6, 8]. The visualization of analysis results, however, gained only recently some attention with the proliferation of data mining [1, 2, 5, 9]. This recent interest was spawned by the often overwhelming number and complexity of data mining results.

The visualization of analysis results primarily serves four purposes:

- better illustrate the model to the end user,
- utilize comparison of models,
- increase model acceptance, and
- enable model editing and support for "what-if questions".

The proposed novel visualization method can be used to visualize the output of any subgroup discovery algorithm, provided that the output has the form of rules with a target class in their consequent. It can also be used as a tool for visualizing standard classification rules. Its unique property is that it allows us to compare distributions of different subgroups.

The approach assumes the existence of at least one numeric (or ordered discrete) attribute of expert's interest for subgroup analysis. The selected attribute is plotted on the X-axis of the diagram. The Y-axis usually represents a class, or more precisely, the number of instances belonging to some target class. It must be noted that both directions of the Y-axis (Y^+ and Y^-) are used to indicate the number of instances. In Figure 2, for instance, the X-axis represents *age*, the Y^+-axis denotes class coronary heart disease (CHD) and Y^- denotes class

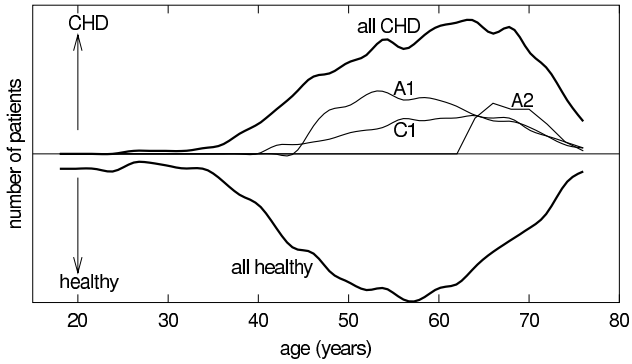


Fig. 2. Distribution of CHD patients and healthy subjects with respect to age in years. Graphs A1, A2, and C1 present corresponding model properties. Model A1 is for men, model A2 is for women, and model C1 represents patients with left ventricular hypertrophy. Healthy persons covered by models A1, A2, and C1 are not displayed.

non-CHD (or 'healthy'). Out of four graphs at the Y^+ side, three represent induced subgroups ($A1$, $A2$ and $C1$) of CHD patients, and the fourth shows the age distribution of the entire population of CHD (all CHD) patients. The graph at the Y^- side shows only the distribution of non-CHD (all healthy) patients in the training set. Note that the subgroups $A1$, $A2$ and $C1$ also cover some non-CHD patients, but the coverage of negative cases is not displayed for better viewing.

In general, it is not necessary that Y^+ and Y^- denote two opposite classes. If appropriate, they may denote any two classes, or even any two different attribute values, which the expert would like to compare.

Figures of this type can be drawn for any available numeric attribute and they are very valuable in the expert interpretation of the obtained results. For example, from Figure 2 it can be seen that there is no significant difference between CHD patients and healthy subjects in respect to their age, but that there are significant differences among detected models. From Figure 3 it can be noticed that there is a similar effect for the total cholesterol values although it is known that total cholesterol is an important risk factor for the CHD disease. This effect shows that the problem of CHD risk group detection typically can not be solved on the level of one feature and it demonstrates the importance of the descriptive induction methods which tries to describe models by a logical combination of a few correlated features. The advantage of the suggested visualization approach is that it makes such relations obvious. In this context, Figure 4 is interesting because it is different from the previous one. At first, it clearly demonstrates significant differences between all CHD and all healthy subjects in respect to ECG ST segment depression values, demonstrating that this measurement is an excellent disease indicator. But also, it shows that, although it is known that models A1-C1 cover various disease subpopulations, they behave very similarly in respect to the ECG ST segment depression property.

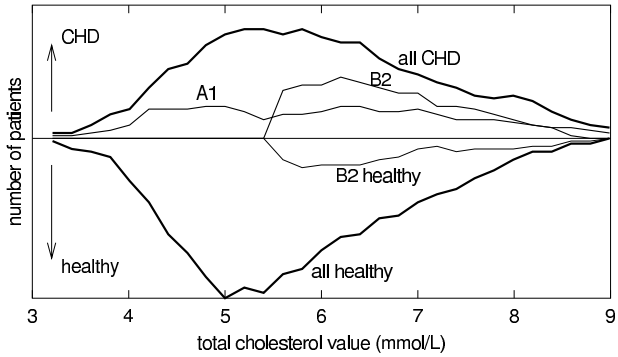


Fig. 3. Distribution of all CHD patients, CHD cases described by models A1 and B2, all healthy subjects, and healthy subjects erroneously included into model B2, with respect to total cholesterol value in $mmolL^{-1}$.

4 CONCLUSIONS

Subgroup visualization, described in this paper, allows us to compare distributions of different subgroups in terms of the selected attribute, plotted on the X-axis of the diagram. In medical domains we typically use the Y^+ side to represent the number of positive cases in order to reveal properties of induced models for subgroups of these patients. On the other hand, the Y^- side is reserved to reveal properties of these same models (or other models) for the negative cases. One of the advantages of using Y^+ and Y^- as proposed above is that in binary classification problems the comparison of the area under the graph of a subgroup and the graph of the entire population visualizes the fractions of $\frac{TP}{Pos} = \frac{TP}{TP+FN}$ at the Y^+ side (sensitivity TPr), and $\frac{FP}{Neg} = \frac{FP}{TN+FP}$ at the Y^- side (false alarm FPr), where Pos and Neg stand for the numbers of positive and negative cases in the entire population, respectively. For instance, in the visualization of subgroup $C1$ in Figure 4 the area under the thin line on the Y^- side represents the numbers of misclassified training instances of subgroup $C1$.

Acknowledgment

This work has been supported in part by the Croatian Ministry of Science and Technology, the Slovenian Ministry of Education, Science and Sport, and the EU funded project Data Mining and Decision Support for Business Competitiveness: A European Virtual Enterprize (IST-1999-11495). We are grateful to Goran Krstačić from the Institute for Cardiovascular Prevention Rehabilitation, Zagreb, Croatia for his involvement in the experiments in the CHD risk domain. The visualization presented in Figures 1 was developed by A. and G. Andrienko, AIS, FhG, Sankt Augustin, Germany.

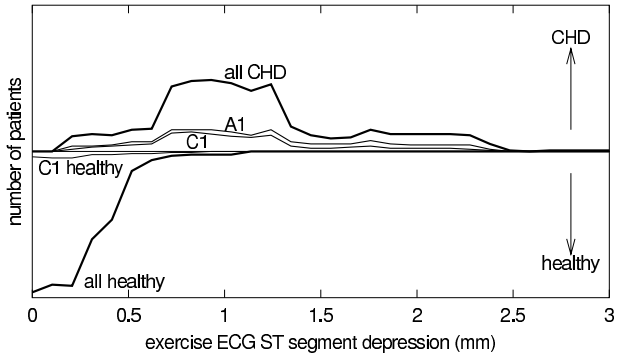


Fig. 4. Distribution of CHD patients and healthy subjects with respect to exercise ECG ST segment depression in millimeters (1mm corresponds to 0.1 mV). Large difference between total healthy and ill populations can be noticed, but differences among models are very small. Models A1 and C1 are selected as extreme cases. The thin line on the Y^- side represents the misclassified cases by subgroup $C1$.

References

1. Card, S.K., Mackinlay, J.D., & B. Shneidermann, B. (1999) Readings in information visualization. Morgan Kaufmann.
2. Fayyad, U.M., Grinstein, G.G., & Wierse, A. (2002) Information visualization in data mining and knowledge discovery. Morgan Kaufmann.
3. Gamberger, D. & Lavrač, N. (2002) Descriptive induction through subgroup discovery: a case study in a medical domain. In *Proc. of 19th International Conference on Machine Learning (ICML2002)*, Morgan Kaufmann, in press.
4. Jovanoski, V. & Lavrač, N. (2001) Classification Rule Learning with APRIORI-C. In *Proceedings of the Tenth Portuguese Conference on Artificial Intelligence, EPIA-2001*, Porto, Portugal, pp.44–51.
5. Keim, D.A & Kriegel, H.P. (1996) Visualization techniques for mining large databases: a comparison. In *IEEE Transactions on Knowledge and Data Engineering*, Vol. 8:6, pp. 923–938.
6. Lee, H.Y, Ong, H.L., and Quek, L.H. (1995) Exploiting visualization in knowledge discovery. In *Proc. of the First Inter. Conference on Knowledge Discovery and Data Mining*, pp. 198–203.
7. Maron, D, Ridker, P.M., & Pearson, A.T. (1998) Risk factors and the prevention of coronary heart disease. In *A.R. Wayne, R.C. Schlant, V. Fuster : HURST'S: The Heart*, 1175-1195. McGrawc Hill, NY.
8. Unwin, A. (2000) Visualisation for data mining, http://www1.math.uni-augsburg.de/ unwin/
9. Workshop on visual data mining, PKDD 2001, Freiburg, Germany. http://www-staff.it.uts.edu.au/~simeon/vdm_pkdd2001/
10. Wrobel, S. (1997) An algorithm for multi-relational discovery of subgroups. In *Proc. First European Symposium on Principles of Data Mining and Knowledge Discovery*, pp.78–87, Springer.

Combined Data Mining and Decision Support Approach to the Prediction of Academic Achievement

Silvana Gasar [1], Marko Bohanec [2,3], Vladislav Rajkovič [4,2]

[1]High School Jesenice, Ruparjeva 2, SI-4270 Jesenice, Slovenia
silvana.gasar@telesat.si
[2]Institute Jožef Stefan, Jamova 39, SI-1000 Ljubljana, Slovenia
[3]University of Ljubljana, School of Public Administration Ljubljana, Slovenia
marko.bohanec@ijs.si
[4]University of Maribor, Faculty of Organisational Sciences, Kranj, Slovenia
vladislav.rajkovic@fov.uni-mb.si

Abstract. We present the development of multi-attribute hierarchical models for the prediction of final academic achievement in a particular high-school educational program. The models were developed by a sequential application of data mining (DM) and decision support (DS) techniques. A database of pupils' achievements was first analyzed by DM methods: statistical analysis, clustering, decision trees and hierarchical multi-attribute models. The findings were incorporated into expert-developed DS models. Predictive accuracy of these models is comparable to that of experienced human experts.

1 Introduction

Data Mining (DM) and Decision Support (DS) are complementary modeling disciplines; DS [1] tends to rely on knowledge acquired from experts, while DM [2] attempts to extract it from data. Recently, Bohanec and Zupan [3] proposed an approach that combines DM and DS for the development of qualitative hierarchical multi-attribute models. The approach combines two methods: DEX [4] as a DS method for model development based on expert knowledge, and HINT [5] as a DM method that discovers concepts and models from data. Since both methods share a common model representation, they can be combined in a number of ways, such as supervised, serial, or parallel. These modes of operations were demonstrated on a real-life case of housing loan allocation [3], indicating a considerable improvement of classification accuracy and comprehensibility of models developed in a combined way. However, that study had a weak point: it was based on a case that had been originally approached only by DEX, and for the integration of DEX and HINT, the case was revisited several years later in a somewhat hypothetical setting. Thus, the study explicitly recommended further practical evaluation of the approach.

In this paper, we present such a real-life case in which the combination of DM and DS methods has taken place from the beginning. The case is in the area of education: the aim was to develop a hierarchical multi-attribute decision model for the prediction

of final academic achievement in a particular high-school educational program. We used a database of pupils collected in one of Slovenian high schools. In order to discover the patterns and indicators that determine academic success or failure, this database was analyzed by a number of DM methods, including basic statistical analysis, clustering, and machine learning of decision trees and hierarchical multi-attribute models. These findings were then taken into account in developing a predictive multi-attribute model, which was done in a DS way by involving an expert and using DEX. In the final stage, the model was thoroughly evaluated from the viewpoint of its predictive accuracy and suitability for practice.

This paper is organized as follows. Section 2 describes the problem of academic achievement prediction, and formulates research questions, goals and methodology. The results of data mining are presented in section 3. These results were combined with expert knowledge to develop two models, which are presented in section 4. The quality of the models is assessed in section 5. The paper is concluded by a summary and recommendations for further research.

2 Problem

Academic achievement depends on the consistency between individual's features and demands of school. Therefore, the problem of high school failure has its roots mostly in an inappropriate choice of school. The choice of school or profession is a multi-attribute decision-making process in which the choice takes place at both sides. The goals of pupils and schools are often in conflict: pupils wish to choose the most appropriate school for themselves, and schools want to select only the best candidates. Because of uncertainty involved in the process, there is always a risk of inadequate choice of school with negative consequences. Furthermore, when passing from primary to high school, pupils are still immature; they do not know themselves and they are not fully aware of different opportunities and demands of further education. They need professional advice for choosing schools and educational programs [6].

Educational and professional counseling in our country, Slovenia, is provided mostly by schools' counseling services. A study of their work [7] revealed a number of problems. Counselors are too busy for educational counseling of high quality. Their suggestions are usually intuitive and based on very few and often incomplete data. All schools try to reject "bad" pupils. There is little time and there is a lack of alternatives. At the time of high schools enrolment it is already too late for appropriate activities to prevent academic failure. Because of the conflict situation and subjective advice, pupils and their parents rarely consider counselors' warnings.

Therefore, counselors need a tool for the prediction of final academic achievement in an easy understanding way with high accuracy. Such a tool would provide an opportunity to predict school failure and react properly on time. Indirectly, it would also help to decrease some undesirable social phenomena such as unemployment, delinquency, drug addiction, violence, etc. [8]. In general, academic achievement depends on the interaction of physical, physiological, social and psychological factors [9], which provide a basis for multi-attribute modeling.

Our research goal was to develop a multi attribute model for the prediction of final academic achievement on an individual high school educational program. We wanted to validate the model, to show its strengths and weaknesses, and give recommendations for its application. In particular, we were looking for the answers on the following questions:

- Is it possible to discover general patterns and rules of academic achievements from a database of pupils, such as the one commonly used in Slovenian primary and high schools?
- On this ground, is it possible to build a multi-attribute model for the prediction of final academic achievement on an individual high school educational program?
- How accurate is the prediction of such models?
- If and how can models contribute to the quality of achievements' estimations and quality of school choices?
- Could they improve achievements in general?
- How to implement the approach in practice?

The methodology involved is a combination of DM and DS methods, which were used sequentially as presented in the following two sections.

3 Data Mining

The DM stage was carried out using statistical methods, visualization, clustering, and machine learning. Among machine learning methods we used the development of classification decision trees and development of multi-attribute hierarchical models. We used the tools SPSS [10], Weka [11] and Orange [12]. In accordance with a general DM methodology [2], the analysis was preceded by data preparation and preprocessing, and followed by the interpretation and evaluation of results.

3.1 Data Preparation and Pre-Processing

The analysis was based on a pupils' database that was created in one of Slovenian high schools using a computer program Evidenca [13]. The database was exported to Microsoft SQL Server 2000 [14], which was used as a tool for data preparation. After normalization, the database was integrated into a single table, in which each record contained all data available about one pupil. In total, there are 96 attributes. A part, 19 attributes, is known before enrolment in high school, while the remaining 77 attributes represent school marks and other data obtained in successive grades of the high school. The main groups of attributes are:

- Pupil's personal and demographic data: gender, date and town of birth, citizenship, primary school name, etc.
- Data on academic achievements in primary school: individual subject marks and general achievement marks in the last two years of primary education.

- Data on achievements and behavior in the first, second, third and fourth high school grade: individual subject marks, general achievement mark, discipline sanctions, hours of excused and unexcused absence from school, etc.

For each pupil in the database, his or her academic achievement is already known. It is represented by the following five categories

- 5: graduates with general achievement mark 4 or 5 (B or A) after four years;
- 4: graduates with general achievement mark 3 or 2 (C or D) after four years;
- 3: graduates after five or six years (prolonged time of education);
- 2: fails and stops educating after one or two years;
- 1: fails and stops educating after tree or more years.

This database–hereafter referred to as DB1–contained data about $N = 1794$ pupils. All the records contained complete data known before the enrolment and data about final academic achievements. However, a considerable proportion of data for all school years was incomplete. Therefore, we also created a smaller database, DB2, of $N = 889$ pupils for which complete data was available for all school years.

3.2 Basic Statistical Analysis

We started the analysis with establishing general statistics, measures of correlations between variables and visualization in SPSS. Descriptive statistics and frequency distributions of variables were assessed for both databases, and Spearman coefficients of rank correlation were computed between numerical variables and final academic achievement. The relation between nominal variables and final achievement was established using chi-square test and contingency coefficient.

Table 1: Frequency distribution of achievement categories in DB1.

Category	**1**	**2**	**3**	**4**	**5**
Frequency [%]	12.3	11.9	14.6	51.2	9.9

The distribution of achievement categories is shown in Table 1. It turned out that academic failure (categories 1 and 2) was quite common – more than 20% pupils left the school and never graduated. Such level of failure would be considered a disaster in every work organization [8], but surprisingly not in schools. High percentage of failure in our high schools confirms a poor performance of professional counseling.

The majority of pupils graduate in time with general achievement mark 3 or 2 (C or D in American system). In our analysis, they represent the majority category 4 with apriori classification accuracy of 51.2 %. General achievement marks in the first high school grade are on average one or two marks lower than general achievement marks in primary school. Most marks are between 1 and 3 (E and C), and pupils with 4 and 5 (B and A) are very rare.

Absences from school and discipline sanctions are in negative, while different marks in primary and high school are in positive correlation with achievements. Achievements most strongly correlate with general achievement marks at the end of

grades. Pupils' age at high school enrolment negatively correlates with achievements. Obviously, predictions based on "later" data are more accurate and valid, but also less useful in practice, as they come too late for a successful corrective action.

Statistical analyses also revealed some interesting and unexpected findings, such that particular subject marks from a particular teacher have no validity at all, what deserves serious consideration and appropriate actions of school's management.

3.3 Machine Learning of Decision Trees

Classification decision trees were built with the program Weka, using the algorithm J4.8 [11], a version of well-known Quinlan's algorithm C4.5 [15]. An initial decision tree was built from all attributes. Different decision trees were then developed on the basis of different selections of attributes. Classification accuracy was estimated by 10-fold cross validation.

When using all the attributes, the classification accuracy turned out as high as 99.39%. This means that at the end of high-school education the prediction of final achievement is almost certain. However, this is not practical as we wish to make predictions several years earlier, possibly at the enrolment in high school or at least after the first high school grade. When we limit the prediction to attributes that are known at that time, the classification accuracy of corresponding decision trees degrades considerably. Figure 1 shows that the classification accuracy on the basis of data known before enrolment (16 attributes) only slightly exceeds 50%. At the end of the first grade (30 attributes), the classification accuracy improves to about 60%, and one year later (46 attributes) to slightly below 70%.

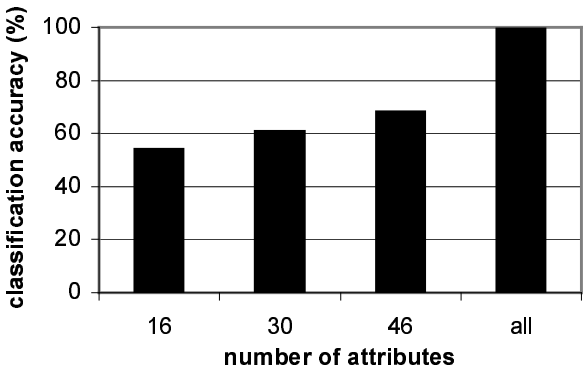


Fig. 1. Classification accuracy of decision trees developed from different attributes

In order to improve classification accuracy, we conducted a number of additional experiments: (1) developing trees on data corresponding to specific educational programs, (2) using different achievement classifications, (3) using cost-sensitive classification, and (4) developing trees on DB2 instead of DB1. The results revealed that developing trees with respect to educational program with attributes known at least at the end of first high school grade seem the most rational. Different achievement classi-

fications and cost-sensitive classification did not improve classification accuracy. Developing trees on DB2 resulted in smaller trees with significantly better classification accuracy – often over 10% better than with DB1.

We concluded that for developing multi-attribute decision models, the best basis provide the trees developed on DB2, corresponding to specific educational program, and using 16 or 30 expert-selected attributes. Despite considerable differences between trees, it seems that the best predictor among the attributes known before enrolment is the final general achievement mark of primary school, and among attributes known at the end of first high school grade the final general achievement mark of the first high school grade.

Other good predictors depend on the educational program. For example, the best tree using 30 attributes for educational program referred to as "L" is shown in Figure 2. Among attributes known at the end of the first grade the most important are the first grade general achievement mark, subject marks in Slovene language, History and Physics in the first high school grade, age at the high school enrolment and unexcused absence in the third semester. These attributes indicate that the program "L" demands general intelligence, verbal abilities, work habits, memory and logical abilities and that the absence diminishes the possibility of success.

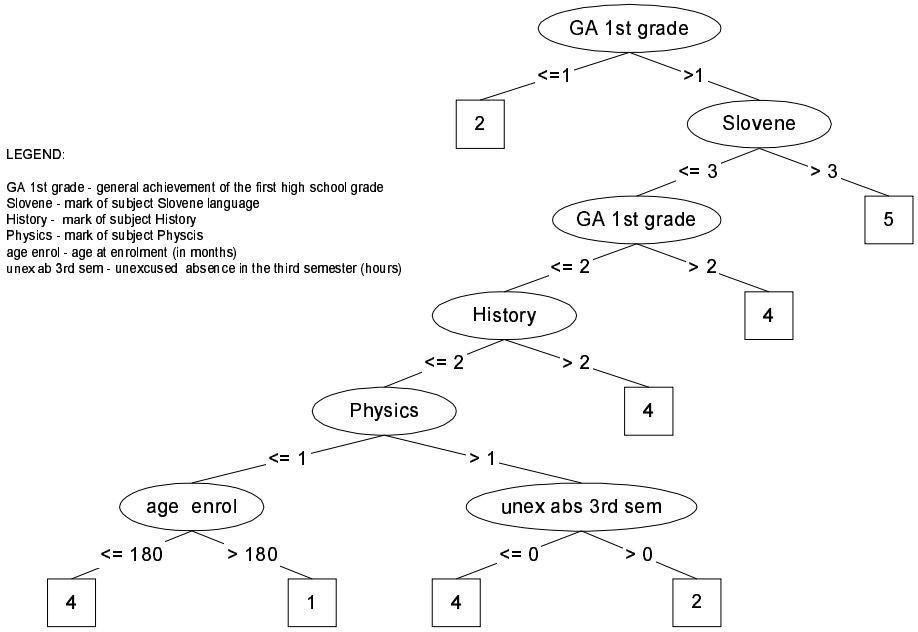


Fig.2. The best tree for educational program "L" (n = 468, classification accuracy is 69.7%)

In summary, classification accuracy of trees using attributes known before enrolment is generally low (about 60%) and slightly exceeds the apriori accuracy (51.2%). Somewhat better, about 70%, is the classification accuracy of decision trees developed from attributes known at the end of first high school grade.

3.4 Clustering

In Weka, pupils were clustered into three, four and five groups using the *k*-means algorithm [16]. Two sets of attributes were used: (1) 16 expert-selected attributes, all known before enrolment in high school, and (2) 30 attributes, known at the end of first high school grade.

Clustering reflected different abilities and motivation of pupils. The best results were achieved by clustering into five groups on the basis of attributes known at the end of first high school grade, because some differences manifest only on the higher and more demanding level of education. The five clusters differ mostly in subject grades and general achievement grades. It is interesting that "worst" pupils in primary school often chose a less demanding educational program, indicating they had at least partially considered advice of educational counselors. But pupils from large cities seem to have higher educational aspirations and despite the same achievement tend to choose more demanding programs, resulting in the lowest achievements of the first high school grade. When abilities are low, high aspirations are not enough for success.

3.5 Machine Learning of Multi-Attribute Hierarchical Models with HINT

Finally, we used HINT to construct multi-attribute models from data. We used its implementation within the DM suite Orange [12], where HINT is limited to discrete data without missing values. Thus, continuous attributes were first discretized and missing values were replaced with majority values. Models were built using 16 and 30 expert-selected attributes, using unsupervised minimal-error decomposition with the default bound set size of two. Again, the classification accuracy was estimated by 10-fold cross validation.

The classification accuracy of HINT models was quite low and unsatisfactory – most often it was close to the apriori accuracy. Such results are partly due to a relatively small learning sample ($N = 889$) and complexity of data.

Although the models were not appropriate for direct use, they provided a number of important guidelines for composing attributes and defining decision rules in the forthcoming manually development of DEX models. For example, among the attributes known before the enrolment, the marks of Math and Physics turned out to be very good predictors. The analysis also revealed the so-called "rule of chain": the marks of Math and Physics mostly limit the highest general achievement. Even more obviously than decision trees, HINT models revealed interesting patterns of absence: pupils are not absent coincidentally or because of health reasons, but systematically and related to exams taking place in school.

4 Development of Decision Support Models

The final multi-attribute decision models were developed manually using DEX and following the three typical DS development stages [4]: (1) acquisition of attributes, (2) development of the hierarchy of attributes, and (3) defining decision rules. The expert

tried to incorporate her previous knowledge about academic achievement prediction and supplement it by the results of the DM stage.

First, a list of attributes was created on the basis of expert opinion; the best decision trees and best models developed by HINT. Beside demographical variables, which represent extenuating circumstances or difficulties in reaching a high achievement, the list mostly included different school marks, hours of absence and discipline sanctions, which reflect pupil's knowledge, abilities, motivation and other personality traits. Continuous attributes were discretized in the same way as with HINT.

Next, attributes were organized into a hierarchical structure according to their dependence and interaction, introducing new aggregate attributes (internal nodes) whenever necessary. This turned out to be the most difficult task, since each new attribute is a result of complex interaction of multiple basic attributes. Sometimes it is difficult to accurately estimate the contribution of each single factor, because it is changeable and may depend on an individual pupil.

Eventually, two attribute hierarchies were developed in this way (Figure 3). The first one uses only attributes that are known before the enrolment in high school. The second hierarchy contains an extended set of attributes for the prediction at the end of first high school grade. Both structures are meant to be general in the sense that they do not address any particular educational program.

In the final stage, the expert defined decision rules, i.e., rules that determine the aggregation of values from the leaves towards the root of the hierarchy. The expert judged the importance of attributes and determined their weights. The results of data mining were taken into account; for example, due to their important influence to the final achievement, relatively high weights have been given to school-marks of Math and Physics.

Finally, we fine-tuned the models to the requirements of a single educational program "L". Decision rules were appropriately modified; all rules in decision tables were reviewed in detail and changed, if necessary. We considered the accuracy of used predictors and the "rule of chain" by which one single weak link or sub-criteria is enough for failure. The structure of model 2 was simplified by excluding 8 less relevant attributes.

Table 2 shows an example of decision rules for model 1 and program "L". The rules predict pupils' final academic achievement on the basis of their abilities and previous knowledge, motivation, and circumstances. Only rules for the achievement categories 1 and 2 are shown in Table 2. Rule 1, for example, states that if pupils' abilities and previous knowledge are inappropriate and their motivation is lowered, regardless on circumstances, they will achieve the category 1. According to rule 4, the category 2 will be achieved if their abilities are at least appropriate, motivation is lowered and circumstances are negative.

Table 2: An example of decision rules for model 1(program "L").

	Abilities and p. knowledge	**Motivation**	**Circumstances**	**Fin. achievement**
1	Inappropriate	Lowered	*	**1**
2	Inappropriate	*	<= Appropriate	**1**
3	Inappropriate	Appropriate	Positive	**2**
4	>= Appropriate	Lowered	Negative	**2**

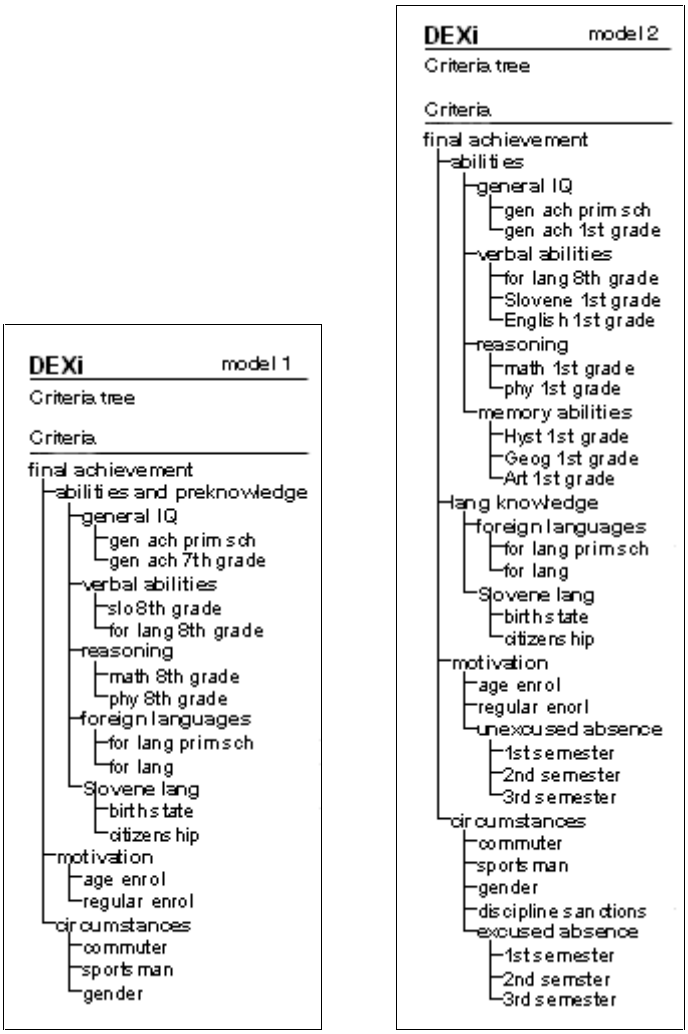


Fig. 3. Structure of the two multi-attribute models

5 Evaluation of Models

A 10% stratified sample (N = 47) was extracted from the database DB1 and used for the evaluation of developed models. We compared the classification accuracy of: (1) both DEX models, (2) decision trees developed in the DM stage, and (3) the classification of the 47 cases, which was provided by an expert counselor.

All the three groups of models achieved almost the same classification accuracy: about 60% using data known at the time of enrolment in high school, and about 70% after one year of study. The models almost never predict failure to really successful

pupils, while pupils with bad prediction usually really fail. Occasionally (i.e., in 12.8% and 8.5% of cases, respectively) the models wrongly predict success to unsuccessful pupils. They recognize typically successful and typically unsuccessful pupils relatively well. Wrong predictions, which are more than one category apart from real achievements, are relatively rare, occurring in 12.8% and 6% of cases, respectively.

In particular, the models perform best in recognizing pupils from the achievement category 4, and worst in recognizing pupils from the category 3. They often intermix cases from the categories 1 and 3; this distinction is indeed difficult because both categories correspond to pupils who educate a long time and progress slowly. But almost until the end we cannot say if they will successfully pass the program (category 3) or fail (category 1).

In summary, with regard to a relatively small number and low quality of used attributes, which are not "pure" neither accurate measures of pupil's features, the predictive accuracy of the models is surprisingly high. We must notice that experts in practice also take into consideration other data, obtained in personal meetings with pupils and their parents, what usually increases the accuracy of their predictions. Our expert counselor was very experienced, but his predictive accuracy does not tell much about average prediction accuracy of school counselors, which differ a lot in experience, abilities, knowledge, intuition, and motivation. As highly experienced and capable experts are rare, the use of predictive models for educational counseling seems meaningful. It may also provide a step toward more consistent, objective and systematic estimation, evidence and evaluation of predictions.

For further work, it is interesting to compare DM and DS models developed in this study and notice an important difference in the treatment of preference-ordered attributes: while DS models do take into account the ordering information, DM ones do not. For example, the achievement categories themselves are preference ordered, because 5 (graduates as A or B) is better than 4 (graduates as C or D), which is better than 3, etc. The rule acquisition mechanism of DEX ensures their consistency, so that if object x is better than or equal to y on all considered preferentially-ordered attributes, then x is not assigned a worse class than y. Neither decision trees nor HINT can ensure this kind of consistency. In further integration of DM and DS, it is thus important to include DM methods that can deal with preference-ordered attributes, for example methods based on rough sets [17].

6 Conclusion

Using a combination of data mining and decision support techniques, we developed and evaluated multi-attribute models for the prediction of final academic achievement in high schools. Their predictive accuracy is close–although in practice probably lower than–the accuracy of an experienced human expert, but may be better than the accuracy of an inexperienced expert. Thus, we believe that at this stage the models are appropriate for experimental use by school counselors, but not yet by pupils and their parents. We also believe that their application would increase the accuracy of predic-

tions and quality of educational counseling, helping to prevent inappropriate choices of school and improve academic achievements.

Both DEX models have a number of strong points. They work with data that are practically always available or easy to get. They facilitate a consistent and systematical prediction and evaluation of estimates. Re-estimation at the end of first high school grade increases the reliability of prediction. The use of models would probably improve pupils' and parents' trust in estimates and consequently contribute to more serious selection of schools and educational programs.

On the weak side, the models occasionally wrongly predict good achievement to pupils who will really fail. By that they encourage them to persist in school that is too demanding for them. Also, the models have not been validated on data other than that used in the analysis, so their general prediction accuracy in practice is unknown, but it is probably lower than established. The predictive accuracy of models can easily degrade due to changes of school system and generational changes of pupils. Thus, the models should be adapted and validated perpetually.

Methodologically, the models were developed by a combination of data mining and decision support techniques. From the viewpoint of DS modeling, which is usually carried out without extensive data analysis and relies mainly on expert knowledge, the preceding DM stage brought considerable benefits. Although difficult to quantify, it is clear that the DM stage helped the expert to better understand the characteristics of the population and to discover some important rules and patterns that affect academic achievement. In particular, important contributions of DM to DS were the following:

- Machine learning of decision trees (section 3.3) provided evidence of what was achievable with the available attributes and data, and established the target classification accuracies to be achieved by DS models under various conditions.
- At the intersection of DM and DS, we used HINT to develop parts of DS models from data (section 3.5). Although HINT's models themselves exhibited poor classification accuracy, some developed subtrees turned out extremely useful and revealed important patterns, such as the "rule of chain". Some concepts discovered by HINT were used almost unchanged in the final model. Thus, HINT has been particularly useful as a knowledge exploration and feature discovery tool.
- After the DS models have been developed, the available DM database facilitated a relatively easy evaluation of the quality of DS models (section 5), which is usually very difficult and thus too often omitted from DS modeling.

From the DM viewpoint, the benefits of combining it with DS in this project are somewhat less clear. Using only DM, we would still obtain a number of models (decision trees) of sufficient classification accuracy. Bringing in the expert and developing a DS model clearly did not improve the accuracy in this case. What we did obtain using DS, however, is a general hierarchy of attributes (Figure 3) that can be easily adapted to various educational programs and used for typical DS tasks, such as what-if and sensitivity analysis, and generation of alternatives. Probably its greatest advantage is that it can be relatively easily extended further using DS tools such as DEX. The extension can introduce new attributes that have not been available in the current database, but are achievable by counselors in schools and may potentially contribute

to the prediction of academic achievement, for example measures of different intellectual abilities, interests, motivation and personality traits of pupil. Such an extension of our models remains a challenge for the future.

Acknowledgment

The work reported here was in part supported by the Slovenian Ministry of Education, Science and Sport, and by the EU project SolEuNet, IST-11495.

References

1. Mallach, E.G.: Understanding Decision Support Systems and Expert Systems. Irwin, Burr Ridge (1994)
2. Han, J., Kamber, M.: Data Mining: Concepts and Techniques. Morgan Kaufman, San Francisco (2001)
3. Bohanec, M., Zupan, B.: Integrating decision support and data mining by hierarchical multi-attribute decision models. ECML/PKDD-2001 Workshop Integrating Aspects of Data Mining, Decision Support and Meta-Learning (IDDM-2001) (eds. Giraud-Carrier, C., Lavrač, N., Moyle, S., Kavšek, B.), Freiburg, (2001) 25–36
4. Bohanec, M., Rajkovič, V.: DEX: An expert system shell for decision support. Sistemica 1(1) (1990) 145–157
5. Zupan, B., Bohanec, M., Demšar, J., Bratko, I.: Learning by discovering concept hierarchies. Artificial Intelligence 109 (1999) 211–242
6. Hughes, M., Wikeley, F., Nash, T.: Parents and their children's schools. Oxford, Cambridge, Blackwell (1994)
7. Resman, M., Bečaj, J., Bezić, T., Čačinovič-Vogrinčič, G., Musek, J.: Svetovalno delo v vrtcih, osnovnih in srednjih šolah (Educational counseling in nursery, primary and high schools). The National Education Institute of Slovenia, Ljubljana (1999)
8. Dryden, G., Vos, J.: Revolucija učenja (Revolution of the learning). Educy, Ljubljana (2001)
9. Hayes, N., Orell, S.: Psychology: an Introduction. Longman, Harlow (1993)
10. Einstein, G., Abernethy, K.: SPSS Tutorial: Statistical Package for the Social Science: SPSS Version 10.0. Furman University. http://s9000.furman.edu/mellonj/spss1.htm (2000)
11. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations. Morgan Kaufmann, San Francisco (2000)
12. Zupan, B., Demšar, J.: Orange. http://magix.fri.uni-lj.si/orange/ (2002)
13. Galle, R.: Priročnik za uporabo programa: vodenje evidence učencev (Evidenca 3) (User's manual for the program Evidenca 3). High School of Electrotechnical and Computer Science, Ljubljana (1996)
14. Vieira, R.: Professional SQL Server 2000 Programming. Wrox Press, Birmingham (2000)
15. Quinlan, R.J.: C4.5: Programs for Machine Learning. Morgan Kaufman, San Francisco (1993)
16. Kaufman, L., Rousseeuw, P.J.: Finding groups in data: an introduction to cluster analysis. New York: J. Wiley & Sons (1990).
17. Greco, S., Matarazzo, B., Slowinski, R.: Rough sets methodology for sorting problems in presence of multiple attributes and criteria. European Journal of Operational Research 138(2) (2002) 247–259.

A Post-processing Environment for Browsing Large Sets of Association Rules [1]

Alipio Jorge[1] , João Poças[2], Paulo Azevedo[3]

[1] LIACC/FEP, Universidade do Porto, Portugal
amjorge@liacc.up.pt
[2] Instituto Nacional de Estatística, Portugal
joao.pocas@ine.pt
[3] Universidade do Minho, Portugal
pja@di.uminho.pt

Abstract. Association rule engines typically output a very large set of rules. Despite the fact that association rules are regarded as highly comprehensible and useful for data mining and decision support in fields such as marketing, retail, medicine, demographics, among others, lengthy outputs may discourage users from using the technique. In this paper we propose a post-processing methodology and tool for browsing/ visualizing large sets of association rules. The method is based on a set of operators that transform sets of rules into sets of rules, allowing focusing on interesting regions of the rule space. Each set of rules can be then depicted with different graphical representations. The tool is web-based and uses SVG. The input set of association rules is given in PMML.

Keywords: Data mining, association rules, post processing, decision support, visualization.

1 Introduction

Association Rule (AR) discovery (Agrawal et al. 96) is many times used, for decision support, in data mining applications like market basket analysis, marketing, retail, study of census data, analysis of medical data, among others. This type of knowledge discovery is adequate when the data mining task has no single concrete objective to fulfil (such as how to discriminate good clients from bad ones), contrarily to what happens in classification or regression. Instead, the use of AR allows the decision maker/ knowledge seeker to have many different views on the data. There may be a set of general goals possibly not measurable (like "what characterizes a good client?", "which important groups of clients do I have?", "which products do which clients

[1] This work is supported by the European Union grant IST-1999-11.495 Sol-Eu-Net and the POSI/2001/Class Project sponsored by FCT

typically buy?"). Moreover, the decision maker may even find relevant patterns that do not correspond to any question formulated beforehand. This style of data mining is sometimes called "fishing" (for knowledge).

Due to the data characterization objectives of the association rule discovery task, AR discovery algorithms produce a complete set of rules above user-provided thresholds (typically minimal support and minimal confidence, defined in Section 2). This implies that the output of such an algorithm is a very large set of rules, which can easily get to the thousands, overwhelming the user. To make things worse, the typical association rule algorithm outputs the list of rules as a long text (even in the case of commercial tools like SPSS Clementine), and lacks post processing (sometimes also called rule mining) facilities for inspecting the set of produced rules.

In this paper we propose a method and tool for the browsing and visualization of association rules. The tool reads sets of rules represented in the proposed standard for predictive models, PMML (Data Mining Group). The complete set of rules can then be browsed by applying rule set operators based on the generality relation between itemsets. The set of rules resulting from each operation can be viewed as a list or can be graphically summarized through a number of techniques.

This paper is organized as follows: we start by introducing the basic notions related to association rule discovery, and association rule space. We then describe PEAR, the post processing environment for association rules and its implementation. We describe the set of operators in more detail, show one example of the application of PEAR, compare with related work and conclude, also suggesting the next steps of our work.

2 Association Rules

An association rule $A \rightarrow B$ represents a relationship between the sets of items A and B. Each item I is an atom representing the presence of a particular object. The relation is characterized by two measures: support and confidence of the rule. The support of a rule R within a dataset D, where D itself is a collection of sets of items (or itemsets), is the number of transactions in D that contain all the elements in $A \cup B$. The confidence of the rule is the proportion of transactions that contain $A \cup B$ with respect to the number of transactions that contain A. Each rule represents a pattern captured in a dataset. The support of the rule is the commonness of that pattern, while the confidence measures its predictive ability.

The most common algorithm for discovering AR from a dataset D is APRIORI (Agrawal et al. 96). This algorithm produces all the association rules that can be found from a dataset D above given values of support and confidence, usually referred to as *minsup* and *minconf*. APRIORI has many variants with more appealing computational properties, but that should produce exactly the same set of rules since the exact set of rules to produce is determined by the problem definition and the data.

2.1 The Association Rule space

The space of itemsets I can be structured in a lattice with the $\subseteq$ relation between sets. The empty itemset $\varnothing$ is at the bottom of the lattice and the set of all itemsets at the top. The $\subseteq$ relation also corresponds to the generality relation between itemsets.

To structure the set of rules, we need a number of lattices, corresponding each lattice to one particular itemset that appears as the antecedent, or to one itemset that occurs as a consequent. For example, the rule $\{a,b,c\} \rightarrow \{d,e\}$, belongs to two lattices: the one of the rules with antecedent $\{a,b,c\}$, structured by the generality relation over the consequent, and the lattice of rules with $\{d,e\}$ as a consequent, structured by the generality relation over the antecedents of the rules.

We can view this collection of lattices as a grid, where each rule belongs to one intersection of two lattices. The idea behind the rule browsing approach we present, is that the user can visit one of these lattices (or part of it) at a time, and take one particular intersection to move into another lattice (set of rules).

3 PEAR: a web-based AR browser

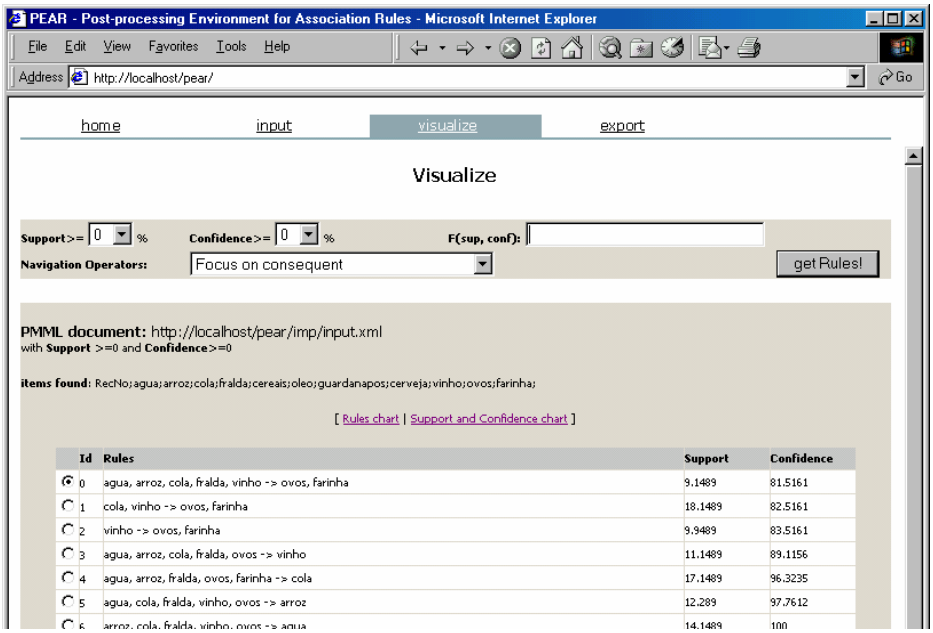


Figure 1: PEAR screen showing some rules.

To help the user browsing a large set of rules and ultimately find the subset of interesting rules, we developed PEAR (Post processing Environment for Association Rules). PEAR implements the set of operators described below that transform one set of rules into another, and allows a number of visualization techniques. PEAR's server is run

under an http server. A PEAR client is run on a web browser. Although not currently implemented, multiple clients can potentially run concurrently.

PEAR operates by loading a PMML representation of the rule set. This initial set is displayed as a web page (Figure 1). From this page the user can go to other pages containing ordered lists of rules with support and confidence.

To move from page (set of rules) to page, the user applies restrictions and operators. The restrictions can be done on the minimum confidence, minimum support, or on functions of the support and confidence of the itemsets in the rule. Operators can be selected from a list. If it is a $\{Rule\} \rightarrow \{Sets\ of\ Rules\}$ operator, the input rule must also be selected.

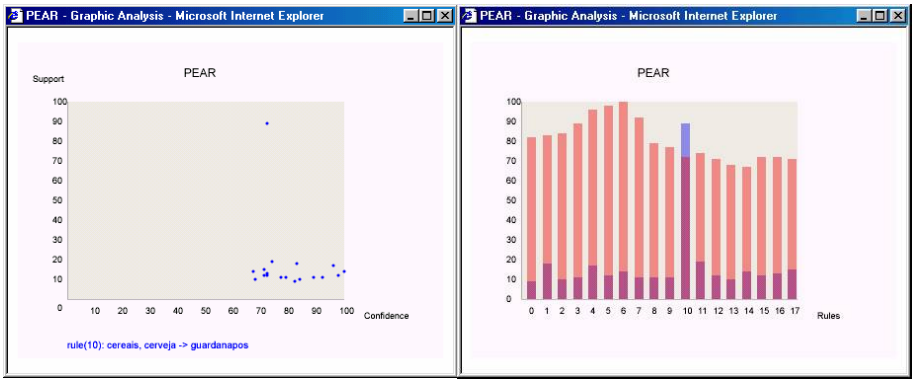


Figure 2: PEAR plotting support x confidence points for a subset of rules, and showing a multi-bar histogram.

For each page, the user can also select a graphical visualization that summarizes the set of rules on the page. Currently, the available visualizations are Confidence × Support plot and Confidence / support histograms (Figure 2). The produced charts are interactive and indicate the rule that corresponds to the point under the mouse.

4 Operators for sets of Association Rules

The association rule browser helps the user to navigate through the space of rules by viewing one set of rules at a time. Each set of rules corresponds to one page. From one given page the user moves to the following by applying a selected operator to all or some of the rules viewed on the current page. In this section we define the set of operators to apply to sets of association rules.

The operators we describe here transform one single rule $R \in \{Rules\}$ into a set of rules $RS \in \{Sets\ of\ Rules\}$ and correspond to the currently implemented ones. Other interesting operators may transform one set of rules into another. In the following we describe the operators of the former class.

Antecedent generalization (AntG)

$AntG(A \rightarrow B) = \{A' \rightarrow B \mid A'$ is obtained by deleting one or more atoms in $A\}$

This operator produces rules similar to the given one but with a syntactically simpler antecedent. This allows the identification of relevant or irrelevant items in the current rule. The support and confidence lines of the resulting set of rules allow the visual identification of items to prune in the antecedent. In terms of the antecedent lattice, it gives all the rules below the current one with the same consequent.

Antecedent least general generalization (AntLGG)

$AntLGG(A \rightarrow B) = \{A' \rightarrow B \mid A'$ is obtained by deleting one atom in $A\}$

This operator is a stricter version of the *AntG*. It gives only the rules on the level of the antecedent lattice immediately below the current rule.

Consequent generalization (ConsG)

$ConsG(A \rightarrow B) = \{A \rightarrow B' \mid B'$ is obtained by deleting atoms in $B\}$

Consequent least general generalization (ConsLGG)

$ConsLGG(A \rightarrow B) = \{A \rightarrow B' \mid B'$ is obtained by deleting one atom in $B\}$

Similar to *AntG* and *AntLGG* respectively, but the simplification is done on the consequent instead of on the antecedent.

Antecedent specialization (AntS)

$AntS(A \rightarrow B) = \{A' \rightarrow B \mid A' \supseteq A\}$

This produces rules with lower support but higher confidence than the current one.

Antecedent least specific specialization (AntLSS)

$AntLSS(A \rightarrow B) = \{A' \rightarrow B \mid A'$ is obtained by adding one (any) atom to $A\}$

As *AntS*, but only for the immediate level above the current rule on the antecedent lattice.

Consequent specialization (ConsS)

$ConsS(A \rightarrow B) = \{A \rightarrow B' \mid B' \supseteq B\}$

Consequent least specific specialization (ConsLSS)

$ConsLSS(A \rightarrow B) = \{A \rightarrow B' \mid$ B' is obtained by adding one (any) atom to $B\}$

Similar to *AntS* and *AntSS*, but on the consequent.

Focus on antecedent (FAnt)

$FAnt(A \rightarrow B) = \{A \rightarrow C \mid C$ is any$\}$

Gives all the rules with exactly the same antecedent. $FAnt(R) = AntG(R) \cup AntS(R)$.

Focus on consequent (FCons)

$FCons(A \rightarrow B) = \{C \rightarrow B \mid C \text{ is any}\}$

Gives all the rules with the same consequent. $FCons(R) = ConsG(R) \cup ConsS(R)$.

5 The Index Page

Our methodology is based on the philosophy of web browsing, page by page following hyperlinks. The operators implement the hyperlinks between two pages. To start browsing, the user needs an index page. This should include a subset of the rules that summarize the whole set. In terms of web browsing, it should be a small set of rules that allows getting to any page in a limited number of clicks. A candidate for such a set could be the, for example, the smallest rule for each consequent. Each of these rules would represent the lattice on the antecedents of the rules with the same consequent. Since the lattices intersect, we can change to a focus on the antecedent on any rule by applying an appropriate operator.

Similarly, we could start with the set of smallest rules for each antecedent. Alternatively, instead of the size, we could consider the support, confidence, or other measure. All these possibilities must be studied and some of them implemented in our system, which currently shows, as the initial page, the set of all rules.

6 One Example

We now describe how the method being proposed can be applied to browse through a set of association rules. The domain considered is the analysis of downloads done from the site of the Portuguese National Institute of Statistics (INE). This site (www.ine.pt/infoline) functions like an electronic store, where the products are tables in digital format with statistics about Portugal.

From the web access logs of the site's http server we produced a set of association rules relating the main thematic categories of the downloaded tables. This is a relatively small set set of rules (211) involving 9 items that serves as an illustrative example. The aims of INE are to improve the usability of the site by discovering which items are typically combined by the same user. The results obtained can be used in the restructuring of the site or in the inclusion of recommendation links on some pages. Although we show here how rules at the highest level of the products taxonomy, a similar study could be carried out for lower levels.

Rule	Sup	Conf
Economics_and_Finance <= Population_and_Social_Conditions & Industry_and_Energy & External_Comme	0,038	0,94
Commerce_Tourism_and_Services <= Economics_and_Finance & Industry_and_Energy & General_Statistic	0,036	0,93
Industry_and_Energy <= Economics_and_Finance & Commerce_Tourism_and_Services & General_Statistic	0,043	0,77
Territory_and_Environment <= Population_and_Social_Conditions & Industry_and_Energy & General_Statist	0,043	0,77
General_Statistics <= Commerce_Tourism_and_Services & Industry_and_Energy & Territory_and_Environm	0,040	0,73
External_Commerce <= Economics_and_Finance & Industry_and_Energy & General_Statistics	0,036	0,62
Agriculture_and_Fishing <= Commerce_Tourism_and_Services & Territory_and_Environment & General_Sta	0,043	0,51

Figure 3: First page (index)

The rules in Figure 3 show the contents of one index page, with one rule for each consequent (from the 9 items, only 7 appear). The user then finds the rule on "Territory_an_Environment" relevant for structuring the categories on the site. By applying an operator, she can drill down the lattice around that rule, obtaining all the rules with a generalized antecedent.

Rule			Sup	Conf
Territory_and_Environment	<=	Population_and_Social_Conditions & Industry_and_Energy & General_Statisti	0,043	0,77
Territory_and_Environment	<=	Population_and_Social_Conditions & Industry_and_Energy	0,130	0,41
Territory_and_Environment	<=	Population_and_Social_Conditions & General_Statistics	0,100	0,63
Territory_and_Environment	<=	Industry_and_Energy & General_Statistics	0,048	0,77
Territory_and_Environment	<=	General_Statistics	0,140	0,54

Figure 4: Applying the operator ConsG (consequent generalization).

From here, we can see that "Population_and_Social_Conditions" is not relevantly associated to "Territory_and_Environment". The user can now look into rules with "Population_and_Social_Conditions" by applying the FAnt (focus on antecedent) operator. From there she could see what the main associations to this item are.

The process would then iterate, allowing the user to follow particular interesting threads in the rule space. Plots and bar charts summarize the rules in one particular page. The user can always return to an index page. The objective is to gain insight on the rule set (and on the data) by examining digestible chunks of rules. What is an interesting or uninteresting rule depends on the application and the knowledge of the user.

7 Implementation

To develop this web environment we chose a Microsoft platform, due to the development background of the team, and also because of the possibilities offered in terms of XML development. This option does not compromise our goal of having a browser-free tool. Currently, all PEAR's features are supported in both Netscape and Internet Explorer. In the following sections we describe the main technologies involved in PEAR. The interactions are summarized in Figure 5.

7.1 Microsoft Internet Information Server

We use the Microsoft Internet Information Server (IIS) as PEAR's http server to run the Active Server Pages (ASP) for server-side programming, allowing database and XML manipulation and form data submitted by the user. PEAR also runs offline with no limitation under Microsoft Personal WebServer (Windows 95/98/2000/Me) or under Microsoft Peer Web Services (Windows NT Workstation). This means it can be installed in any PC with a Microsoft system in it (Windows 98/Me/NT/2000/XP).

7.2 Active Server Pages and VbScript

Active Server Pages (ASP) (Microsoft) are dynamic and interactive web pages processed on the server-side, thus useful to manipulate data submitted by users (for in-

stance, selecting a set of association rules given certain restrictions by the users) as well as manipulating database requests.

An ASP page integrates HTML tags with script commands. These scripts can be either VbScript or Jscript (JavaScript similar). Microsoft JScript is an open implementation of Netscape's JavaScript which are both compliant with the European Computer Manufacturing Association's ECMAScript Language Specification (ECMA-262 standard[2]) When the page is downloaded, these scripts are executed on the Active Server Page environment thus producing the final HTML code to the requesting browser.

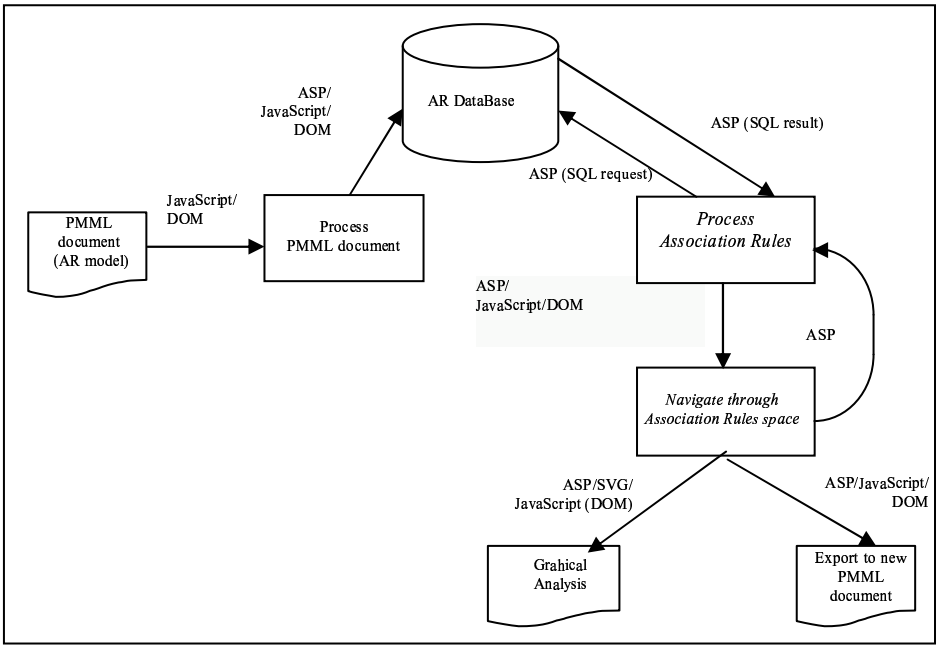


Figure 5: General architecture of PEAR.

PEAR uses VbScript (in Active Server Pages) to process the set of association rules represented in a PMML document (using Document Object Model), to allow the user to browse through it, and to store the rules in a relational database. VbScript is used at the server-side only.

7.3 JavaScript

JavaScript, (Microsoft Web Site) is used for data manipulation on the client-side, for its portability. We try to use only commands that are compliant with ECMAScript. This way PEAR may rub under Netscape and Internet Explorer. With JavaScript we create and manipulate PMML documents or SVG (both XML documents) using

[2] ECMA is an international industry association founded in 1961 and dedicated to the standardization of information and communication systems.

Document Object Model. JavaScript is also important for data validation and for interaction with the user (event handling).

7.4 Document Object Model

The Document Object Model (DOM) is a tree structure-based application program interface (API) for HTML and XML documents issued as a W3C Recommendation in October 1998 (W3C DOM Level 1 specification). It is used to process and manipulate an XML document by accessing its internal structure. DOM represents an XML document as a tree. Its nodes are elements, text, and so on. DOM makes it convenient for application programs to traverse the tree and access the contents of the tree. The Document Object Model provides a standard programming model for working with XML.

PEAR uses the Microsoft XML parser provided in Microsoft Internet Explorer 5 (and above), which implements the W3C DOM specification. With this parser we can easily access and manipulate the internal tree-structure of an XML document. In particular, we use the DOM to read and manipulate the original PMML document (XML document that represents a data mining model), to export a new PMML document and also to create and manipulate the graphical visualization (SVG documents).

7.5 Scalable Vector Graphics

Scalable Vector Graphics (SVG) is an XML-based language that specifies and defines vector graphics that can be visualized by a web browser. [W3C Recommendation]. «…defines the features and syntax for SVG, a language for describing two-dimensional vector and mixed vector/raster graphics in XML». So, using SVG is very similar to working with any other normal XML document. An SVG document must also follow the DTD (Data Type Definition) that specifies the graphic elements that can be produced.

Again, we can manipulate SVG graphics with VbScript or JavaScript (using Document Object Model). With SVG, it is easy to produce a data visualization and even make it interactive (controlling keyboard or mouse events). PEAR gets data from PMML and presents it using VbScript and SVG graphics.

7.6 Database and SQL

We use a relational database (Microsoft Access) to store the PMML model and take advantage of using Structured Query Language (SQL) to obtain sets of association rules. Compared to using DOM directly to manipulate the original PMML document, SQL provides a faster and easier access. In PEAR, all database connections and requests are done with Active Server Pages on the server side.

7.7 Representing Associations Rules with PMML

Predictive Model Markup Language (PMML) is an XML-based language. A PMML document provides a non-procedural definition of fully trained data mining models with sufficient information for an application to deploy them. It provides a way for

people to share models between different applications. Like any XML document, also a PMML document must follow a Data Type Definition (DTD) that defines the entities and attributes for documenting a specific data mining model. For instance, there is one DTD to specify a Regression model; another DTD to represent a Naive Bayes model; other to define an AR model and so on. Any AR model written in PMML by different entities must follow the same AR specific DTD.

A model described using PMML has the following structure:

1) A header,
2) A data schema,
3) A data mining schema,
4) A predictive model schema,
5) Definitions for predictive models,
6) Definitions for ensembles of models,
7) Rules for selecting and combining models and ensembles of models,
8) Rules for exception handling.

Component (5) is required. The other components are optional.

The main reasons that drove the formulation of the PMML for predictive models were that it must be universal, extensible, portable and human readable. It allows users to develop models within one vendor's application, and use other vendors' applications to visualize, analyze, evaluate or otherwise use the models. Previously, this was virtually impossible, but with PMML, the exchange of models between compliant applications now will be seamless. At this moment, only a few data mining tools and applications allows to export their models to PMML, but is urgent to implement it in other software tools to satisfy dramatically increasing requirements for statistical and data mining models in business systems.

PEAR can read an AR model specified in a PMML document. The user will be able to manipulate the AR model, creating a new rule space based on a set of operators, and export a subset of selected rules to a new PMML document.

8 Related Work

There is some work on the visualization and summarization of association rules. In this section we refer to selected work on theme.

The system DS-WEB (Ma et al.) uses the same sort of approach as the one we propose here. In common, DS-WEB and PEAR have the aim of post processing a large set of AR through web browsing and visualization. DS-WEB relies on the presentation of a reduced set of rules, called direction setting or DS rules, and then the user can explore the variations of each one of these DS rules. In our approach, however, we rely on a set of general operators that can be applied to any rule, including DS rules as defined for DS-WEB. The set of operators we define is based on simple mathematical properties of the itemsets and have a clear and intuitive semantics. PEAR also has the additional possibility of reading AR models as PMML.

VizWiz is the non-official name for a PMML interactive model visualizer implemented in Java (Wettshereck). It graphically displays, not only association rules, but

many other data mining models. The philosophy of WizWiz for displaying AR relies on the presentation of the list of rules, allowing the user to set the minimal support and confidence through very intuitive gauges. VizWiz also accompanies the display of each rule by color bars representing support and confidence. This visualizer can be used directly in a web browser as a java plug-in.

(Lent et al 97) describe an approach to the clustering of association rules. The aim is to derive information about the grouping of rules obtained from clustering. As a consequence one can replace clustered rules by one more general rule. For a given attribute in the consequent, the proposed algorithm constructs a 2D grid where each axis corresponds to an attribute in the antecedent. The algorithm tries to find "the best" clustering of rules for non-overlapping areas of the 2D grid. The approach only considers rules with numeric attributes in the antecedents.

9 Future Work and Conclusions

Association rule engines are often rightly accused of overloading the user with very large sets of rules. This applies to any software package, commercial or non-commercial, that we know.

In this paper we describe a rule post processing environment that allows the user to browse the rule space, organized by generality, by viewing one relevant set of rules at a time. A set of simple operators allows the user to move from one set of rules to another. Each set of rules is presented in a page and can be graphically summarized. In the following we summarize the main advantages, limitations and future work of the proposed approach.

The main advantages are:

- PEAR enables selection and browsing across the set of derived AR.
- It enables plotting numeric properties of each subset of rules found.
- Browsing is done by a set of well-defined operators with a clear and intuitive semantics.
- Selection of AR rules by an user is an implicit form of providing background knowledge, that can be later used, for example, in selecting rules for a a classifier made out of a subset of rules.
- PEAR presents an open philosophy by reading the set of rules as a PMML model.

The main limitations are:

- Visualization techniques are always difficult to evaluate. This one is no exception.
- The current implementation requires, on the server-side, the use of an operating system from one specific vendor.
- The entry point (the index page) is still relatively weak.

Future work:

- Develop metrics to measure the gains of this approach.
- Develop mechanisms that allow the incorporation of user defined visualizations and rule selection criteria, such as for example, the combination of primitive operators.

- Evaluate the current implementation against other alternatives such as java, as well as an alternative to client-server, such as plug-in.
- Implement other visual representations of subsets of rules.
- Allow the definition of rule selection criteria based on the support and confidence of the rule, its antecedent and its consequent.

References

1. Agrawal, R., Mannila, H., Srikant, R., Toivonen, H., Verkamo, A. I., Fast Discovery of Association Rules. *Advances in Knowledge Discovery and Data Mining*: 307-328. 1996.
2. Data Mining Group (PMML development), http://www.dmg.org/
3. ECMA-262 standard http://www.ecma.ch/ecma1/STAND/ECMA-262.HTM
4. Lent, B., Swami, A., Widom, J.: Clustering Association Rules, in Alex Gray, Per-Åke Larson (Eds.): *Proc. of the Thirteenth International Conference on Data Engineering, ICDE 97* Birmingham U.K. IEEE Computer Society 1997
5. Ma, Yiming, Liu, Bing, Wong, Kian (2000), Web for Data Mininig: Organizing and Interpreting the Discovered Rules Using the Web, School, *SIGKDD Explorations*, ACM SIGKDD, Volume 2, Issue 1, July 2000.
6. Microsoft Web Site (Active Server Pages) http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnbegvb/html/activeserverpages.asp
7. Microsoft Web Site (Descriptions of Java, JScript, and JavaScript) http://support.microsoft.com/default.aspx?scid=kb;EN-US;q154585
8. W3C, Scalable Vector Graphics (SVG) 1.0 Specification, W3C Recommendation, September 2001, http://www.w3.org/TR/SVG/
9. W3C DOM Level 1 specification http://www.w3.org/DOM/
10. Wettshereck, D., A KDDSE-independent PMML Visualizer, in *Proc. of IDDM-02, workshop on Integration aspects of Decision Support and Data Mining*, (Eds.) Bohanec, M., Mladenic, D., Lavrac, N., associated to the conferences ECML/PKDD 02, Helsinki, Finland, 2002.

Combination of Task Description Strategies and Case Base Properties for Meta-Learning

Christian Köpf and Ioannis Iglezakis

DaimlerChrysler AG, Research & Technology, RIC/AM, P.O.-Box 2360,
D-89013 Ulm, Germany,
{christian.koepf,ioannis.iglezakis}@daimlerchrysler.com

Abstract. Describing a learning task is crucial, not only for meta-learning but also to gain insight in this learning task. The paper evaluates the performance of a recent method for assessing quality standards for case bases when used for a supervised meta-learning. Empirical results on real-world data show this approach in combination with others as a promising one.

1 Introduction

The problem of selecting an appropriate model for a given learning task is a crucial one. Often, there is neither enough time nor space to select learning algorithms from a given pool by simply trying them out. Thus, as users, we want to relate to past experiences of and with learners in the pool to predict which one is most suitable for a given task. This might be in terms of measures such as predictive accuracy, time, or comprehensibility. In this work, we limit ourselves to predictive accuracy.

How can we relate to a past experience with a learner, how can we describe a new learning task adequately? We will concentrate on two already known strategies for task description here, namely landmarking ([15], [2]) and data characterization (which we will refer to as DCT due to the name of a software used to compute the characteristics) ([7], [12], [11]). Additionally, we will experiment with a new approach which has recently been in use in the field of case base reasoning to assess the quality of case bases. This approach will be used for meta-learning and will also be combined with other already used measures.

We begin by introducing the already mentioned strategies for task description, landmarking and DCT. Afterwards, we will dwell on the case properties extracted from case bases to evaluate their quality by possible conflicts between items within a case base. This is followed by empirical results with real-world data. Eventually, the last section summarizes the paper and points at future work.

2 Task Description Strategies

Probably the most common way is to use data characteristics to describe a given learning task for either classification or prediction. At first, basic information is

computed such as number of classes, number of attributes, both symbolic and numeric, number of observations and number of missing values. These measurements are supposed to give a first estimation of the learning problem.

They are extended by statistical measures which are supposed to inform about the distribution of the numeric attributes. For instance, several measures might be computed to check if the given learning task meets the assumption of a discriminant analysis. Measures such as eigenvalues and discriminant functions are computed from the data where the relative proportion of the first discriminant function is given by

$$Fract1 = \frac{\lambda_{max}}{d} \tag{1}$$

where λ_{max} denotes the largest eigenvalue. In addition, the canonical correlation of the best linear combination of attributes is given by [10]

$$CanCor1 = \sqrt{\frac{\lambda_{max}}{1 + \lambda_{max}}}. \tag{2}$$

Eventually, information theoretic measurements are computed to test how much the symbolic attributes contribute to correctly classifying the labeled objects. The entropy of an attribute A as a realization of a discrete random variable $\mathbf{X}$ with k characteristics is given by [19]

$$H_A = H(A) = -\sum_{i=1}^{k} p_i \log_2(p_i) \tag{3}$$

where p_i $(1 \leq i \leq k)$ is the probability for A taking the i^{th} value $(\sum_{i=1}^{k} p_i = 1)$. The entropy of a symbolic target variable with q characteristics, referred to as the class entropy, is given by

$$H_C = H(C) = -\sum_{i=1}^{q} p_i \log_2(p_i). \tag{4}$$

If p_{ij} denotes the joint probability of observing class C_i and the j^{th} value of attribute A, the joint entropy defined as

$$H_{CA} = H(C, A) = -\sum_{i,j} p_{ij} \log_2(p_{ij}) \tag{5}$$

is a measure of the total entropy of the joint system of these attributes, i.e. of all combinations (C, A). Then, the information gain or mutual information is given by [16][17]

$$I_{gain}(C, A) = H_C + H_A - H_{CA}. \tag{6}$$

These measurements have been explored for a meta-learning approach within the StatLog project (1991–1994) and a detailed description can be found in [14].

A completely different approach was chosen by [15] and [2]. Instead of using measurements for describing a given learning problem statistically which is not

directly related to the performance of algorithms, fast learning algorithms are used to describe the problem adequately. There is little point in measuring data characteristics to predict the performance of a classifier if this measurement process takes longer to execute than running the classifier(s) in question. Rooted in the StatLog project under the term "yard stick methods", this approach is known as landmarking. The above mentioned authors proposed the use and motivation of the following landmarkers.

1. *Decision Node:* Using $C5.0$'s information gain-ratio [17] a single decision node is chosen which is then to be used for classifying test observations. The goal of this landmark learner is to establish closeness to linear separability.
2. *Randomly Chosen Node:* An attribute is chosen randomly and then used for splitting the training set and classifying new observations. The goal of this landmark learner is to inform about irrelevant attributes.
3. *Worst Node:* By using the information gain ratio again, the least informative attribute is used to make the single split. Together with the first landmark learner, this landmarker is supposed to inform about linear separability.
4. *Naïve Bayes*: The necessary probabilities for using Bayes' theorem [6] are computed on the training set in order to classify observations in the test set. The goal of this landmark learner is in measuring the extent to which attributes are conditionally independent given the class.
5. *1-Nearest Neighbor:* According to the closest observation in the training set, a new observation in the test set is classified [6]. The goal of this landmark learner is in determining how close instances belonging to the same class are.
6. *Elite 1-Nearest Neighbor:* This landmarker works like the previous one, although it is computed on a subset of all attributes. This subset is determined by the most informative attributes.[1]
7. *Linear Discriminant:* Using the training set, a linear target function is computed which is then used to classify observations from the test set [10].

Landmarking proved to be a competitive method for task description since the results of the landmarkers are directly related to more "sophisticated" algorithms instead of the indirect data characteristics. However, we might also want to consider the meaning and interpretation of possible outcomes. At the end of a meta-learning experiment, we might like to discover some useful insights into when algorithms perform well. Thus, data characteristics are still of importance for meta-learning.

Finally, a meta-data set comprises a number of meta-observations each of which represents an actual data set. The above described data characteristics (basic, statistical, and information theoretic measures) and landmarkers are used as meta-attributes trying to adequately describe the original data sets with the final aim of model selection. This can be either done by a classification approach trying to predict the algorithm out of a given learners pool that will yield the

[1] Here, only attributes are taken into account for which the information gain ratio was smaller than 1. This threshold is due to results obtained by [1]. This algorithm is part of a set of algorithms called *Edited 1-Nearest Neighbor*.

lowest misclassification rate or by a regression approach where the error rate of each learner from the pool is to be predicted [3] [11]. In the latter case, it is up to the user's experience which of the learners is eventually chosen. In the section following the next one, we will use different sets of meta-attributes for meta-learning. Their formation will then be explained in more detail.

3 Case Base Properties

One major drawback of the data characterization scenario is that information theoretic and statistical measures take either numeric or symbolic attributes into account. Often though, the measures describing the basic properties of a learning task, say, contain nearly equally as much information. A possible way of taking information contained in all attributes into account is to compare observations with each other. This might be helpful in various ways. A data set may contain two observations with similar or equal attribute values, but with different labels which might cause a classifier to get "confused". Analogously, there might be two or more observations which are identical. In such a case, the observation might be given more weight, however, the information contained in it might be redundant for the classifier. Also, in this very case, attribute values might be missing, so that one observation would actually be a subset of another observation. Such an approach is described in detail in [9]. There, case base properties are used to assess the quality of given case bases in terms of measures such as redundancy or incoherency. Following and using the notation given in [9], we will briefly introduce some necessary requirements for the implementation of the case based properties which is followed by an example demonstrating the approach. To begin with, we have to settle on the notation. For a more thorough description, however, see [9] and [18].

Definition 1 (Cases and Case Base).

1. *An attribute a_j is a name accompanied by a set $V_j := \{v_{j1}, \ldots, v_{jk}, \ldots, v_{jN_j}\}$ of values. We denote the set of attributes as $A := \{a_1, \ldots, a_j, \ldots, a_N\}$.*
2. *A problem is a set $p_i := \{p_{i1}, \ldots, p_{ij'}, \ldots, p_{iN_i}\}$ with $\forall j' \in [1; N_i]\ \exists a_j \in A$ and $\exists v_{jk} \in V_j : p_{ij'} = v_{jk}$, and $\forall j \in [1; N] : |(p_i \cap V_j)| \leq 1$. We denote the set of problems as $P := \{p_1, \ldots, p_i, \ldots, p_M\}$.*
3. *A solution s_i is any item.*
4. *A case is a tuple $c_i := (p_i, s_i)$ with a problem p_i and a solution s_i. A case base is a set of cases $C := \{c_1, \ldots, c_i, \ldots, c_M\}$.*
5. *We further assume a separation of C into a training set T and a test set (or query set) Q with $C = T \cup Q$ and $T \cap Q = \emptyset$.*

Additionally, we have to define functions to be able to determine the similarity between two given cases, that is to say two observations.

Definition 2 (Auxiliary Functions). *Assume a local similarity measure $sim_j : V_j \times V_j \mapsto [0; 1]$.*

1. $S_{\leftrightarrow} : P \times P \mapsto \{1..N\}$,
   $S_{\leftrightarrow}(p_i, p_{i'}) := \left|\{j \in \{1..N\} : |p_i \cap V_j| = |p_{i'} \cap V_j| = 1 \ \wedge \ sim_j(p_{ij}, p_{i'j}) = 1\}\right|$
2. $S_{\leftrightsquigarrow} : P \times P \mapsto \{1..N\}$,
   $S_{\leftrightsquigarrow}(p_i, p_{i'}) := \left|\{j \in \{1..N\} : |p_i \cap V_j| = |p_{i'} \cap V_j| = 1 \ \wedge \ sim_j(p_{ij}, p_{i'j}) \neq 1\}\right|$
3. $S_{\leftarrow} : P \times P \mapsto \{1..N\}$,
   $S_{\leftarrow}(p_i, p_{i'}) := \left|\{j \in \{1..N\} : |p_i \cap V_j| > |p_{i'} \cap V_j|\}\right|$
4. $S_{\rightarrow} : P \times P \mapsto \{1..N\}$,
   $S_{\rightarrow}(p_i, p_{i'}) := \left|\{j \in \{1..N\} : |p_i \cap V_j| < |p_{i'} \cap V_j|\}\right|$

The overall similarity in the following definition is the normalized weighted sum of the above introduced and computed auxiliary values. Values coinciding for the same attribute as positive are considered. Different values, however, do not contribute positive to local similarity values. Note also that for all other values ($S_{\leftarrow}(p_i, p_{i'}), S_{\rightarrow}(p_i, p_{i'})$, and $S_{-}(p_i, p_{i'})$), weights $w_{\leftarrow}$, $w_{\rightarrow}$, and w_{-} decide whether we consider their relations as positive ($w = 1$) or negative ($w = 0$).

Definition 3 (Similarity Measure). *Assume* $w_{\leftarrow}, w_{\rightarrow}, w_{-} \in \{0, 1\}$.
$sim : P \times P \mapsto [0; 1]$,

$$sim(p_i, p_{i'}) := N^{-1} \cdot \Big(S_{\leftrightarrow}(p_i, p_{i'}) + w_{\leftarrow} \cdot S_{\leftarrow}(p_i, p_{i'}) + w_{\rightarrow} \cdot S_{\rightarrow}(p_i, p_{i'}) + w_{-} \cdot S_{-}(p_i, p_{i'}) \Big).$$

Eventually, the case base properties are defined as follows.

Definition 4. *Assume* $G \subseteq C$, $c_i \in G$, *and* $1 \leq \Delta \in \mathbb{N}$.

1. c_i *consistent within* $G :\Longleftrightarrow \nexists c_{i'} \in G : s_i \neq s_{i'} \ \wedge \ \mathcal{S}_{\leftrightarrow}(p_i, p_{i'}) + \mathcal{S}_{\leftarrow}(p_i, p_{i'}) = N_i \geq N_{i'} \ \wedge \ \mathcal{S}_{\leftrightarrow}(p_i, p_{i'}) > 0 \ \wedge \ \mathcal{S}_{\leftarrow}(p_i, p_{i'}) \geq 0 \ \wedge \ \mathcal{S}_{\rightarrow}(p_i, p_{i'}) = 0$.
2. c_i *unique within* $G :\Longleftrightarrow \nexists c_{i'} \in G, c_{i'} \neq c_i : s_i = s_{i'} \ \wedge \ \mathcal{S}_{\leftrightarrow}(p_i, p_{i'}) = N_i = N_{i'} \ \wedge \ \mathcal{S}_{\leftrightarrow}(p_i, p_{i'}) > 0$.
3. c_i *minimal within* $G :\Longleftrightarrow \nexists c_{i'} \in G : s_i = s_{i'} \ \wedge \ \mathcal{S}_{\leftrightarrow}(p_i, p_{i'}) + \mathcal{S}_{\leftarrow}(p_i, p_{i'}) = N_i > N_{i'} \ \wedge \ \mathcal{S}_{\leftrightarrow}(p_i, p_{i'}) > 0 \ \wedge \ \mathcal{S}_{\leftarrow}(p_i, p_{i'}) > 0 \ \wedge \ \mathcal{S}_{\rightarrow}(p_i, p_{i'}) = 0$.
4. c_i *incoherent*$_{\Delta}$ *within* $G :\Longleftrightarrow \nexists c_{i'} \in G : s_i = s_{i'} \ \wedge \ \mathcal{S}_{\leftrightarrow}(p_i, p_{i'}) + \mathcal{S}_{\leftrightsquigarrow}(p_i, p_{i'}) + \mathcal{S}_{\leftarrow}(p_i, p_{i'}) = N_i = N_{i'} \ \wedge \ \mathcal{S}_{\leftrightarrow}(p_i, p_{i'}) > 0 \ \wedge \ \mathcal{S}_{\leftrightsquigarrow}(p_i, p_{i'}) \geq 0 \ \wedge \ \mathcal{S}_{\leftarrow}(p_i, p_{i'}) \geq 0 \ \wedge \ \mathcal{S}_{\rightarrow}(p_i, p_{i'}) \geq 0 \ \wedge \ \mathcal{S}_{\leftarrow}(p_i, p_{i'}) = \mathcal{S}_{\rightarrow}(p_i, p_{i'}) \ \wedge \ \mathcal{S}_{\leftrightsquigarrow}(p_i, p_{i'}) + \mathcal{S}_{\leftarrow}(p_i, p_{i'}) = \Delta$.

To illustrate the given definitions, the examples in Table 1 will be helpful. Pairs of cases, their conflict to each other and the resulting values for the auxiliary functions in Definition 2 are shown. Note that the symbol $\neg$ denotes the negation of a proposition. Note as well that by using these case base properties, suspicious observations which might impair the results of learning algorithms can be removed which was the original intention behind this approach. This, however, is more of a preprocessing task which is beyond the scope of our work. Instead, we use the computed measurements as meta-attributes to add more information to the meta-learners.

	p_i	s_i	$p_{i\prime}$	$s_{i\prime}$	Proposition	$S_{\leftrightarrow}$	$S_{\rightsquigarrow}$	$S_{\leftarrow}$	$S_{\rightarrow}$	S_{-}	Δ
1	v_{11} v_{21} v_{31}	s_1	v_{11} v_{21}	s_2	¬ consistent	2	0	1	0	2	-
2	v_{11} v_{21} v_{31}	s_1	v_{11} v_{21} v_{31}	s_1	¬ unique	3	0	0	0	2	-
3	v_{11} v_{21} v_{31}	s_1	v_{11} v_{21}	s_1	¬ mininmal	2	0	1	0	2	-
4	v_{11} v_{21} v_{31} v_{41}	s_1	v_{11} v_{21} v_{42} v_{51}	s_1	¬ incoherent$_2$	2	1	1	1	0	2

Table 1. Examples for Pairs of Cases and the corresponding propositions with respect to general case properties

4 Results

A meta-data set was constructed using 78 data sets from the UCI repository [4]. The number of observations did not exceed 1066, and the number of attributes ranged from 4 to 69. 32 data sets contained only symbolic attributes, 20 data sets contained only numeric attributes. The remaining sets were mixed. The data contained up to 25% missing values. Error rates for ten different classification algorithms from the Metal project [13] were determined for different subsets of data characteristics by a ten-fold cross validation, viz. c50boost, tree, and rules [17], the neural networks clemMLP, clemRBFN, both implemented in Clementine, the discriminant tree learner Ltree [8], the rule learner RIPPER [5], a linear discriminant learner, a naive Bayes learner and an instance-based learner. In all cases, the default settings were used.

To begin with, we tried to evaluate various ways on how to represent the data adequately. By adequately, we mean a representation that would give an error rate as small as possible for each algorithm. As a basic set of data characteristics to be used for meta-learning, denoted by DCT_b, we computed the number of attributes, both symbolic and numeric, the number of observations and the number of classes. Additionally, this basic set was either amended by the accuracy and standard deviation of the default class, denoted by DCT_{bd}, or the number of missing values and tuples containing missing values, denoted by DCT_{bm}. Consequently, DCT_{bmd} represents the combination of all measurements. Additionally, an often proposed strategy is to use seven features given by the proportion of both symbolic attributes, attributes with outliers and missing values, the number of observations, the class entropy and mutual information as well as CanCor1, denoted by DCT_{com}. Note that we restricted ourselves here to three base learners, namely Ltree, Naive Bayes, and c50rules. In case, the learners performed equally, the meta-observation was labelled as "TIE". Our goal was to predict the algorithm with the lowest error rate. The corresponding error rates for a ten-fold cross-validation are given in Table 2. Obviously, DCT_{bd} performs best, being significantly better than most other approaches. The information contained within the missing values contributes poorly to predicting the correct class labels whereas the default accuracy seems much more appropriate.

As previously mentioned, we followed various ways to describe a given learning task. First, we computed data characteristics for a given data set. This was

Meta-learner	DCT_b	DCT_{bd}	DCT_{bm}	DCT_{bdm}	DCT_{com}
Default Class	63.16	63.16	63.16	63.16	63.16
c50boost	43.42	25.00	44.73	34.21	35.52
c50tree	50.00	43.42	53.94	44.73	40.79
IB	42.11	36.84	43.42	36.84	46.05
Ripper	52.63	52.63	59.21	51.32	39.47
Average	47.04	39.47	50.32	41.78	40.46

Table 2. Percentage error rates for DCT strategies and different meta-learners

followed by computing error rates using the landmarking algorithms as meta-attributes. Ext-Land is based on the seven landmarkers given in [2] whereas Landmarking goes without those learners being both in the learners and landmarkers pool, viz. LinDiscr, NB, and IB. Eventually, we computed the case base properties for each of the data sets for different values of ε, $\varepsilon = 0.01, 0.05, 0.1$, which indicates the possible distance between observations. As can be seen from table 3, both Landmarking and Ext-Land perform on average significantly better than the approach using case base properties, in particular than $CBR_{0.1}$.

Meta-learner	Landmarking	Ext-Land	$CBP_{0.01}$	$CBP_{0.05}$	$CBP_{0.1}$
Default Class	63.16	63.16	63.16	63.16	63.16
c50boost	56.57	52.68	69.74	60.52	63.16
c50tree	56.58	53.94	59.21	67.11	71.05
IB	55.26	47.36	50.00	57.89	64.47
Ripper	57.89	52.63	59.00	69.73	67.11
Average	56.25	52.30	59.48	63.81	66.45

Table 3. Error rates for different meta-learners and task description strategies

Meta-learner	DCT_b	DCT_{bd}	DCT_{bm}	DCT_{bmd}	DCT_{com}
Default Class	63.16	63.16	63.16	63.16	63.16
c50boost	38.15	36.82	46.05	42.11	34.12
c50tree	47.36	44.73	51.31	47.37	40.29
IB	47.36	39.47	51.31	44.73	44.39
Ripper	57.89	55.26	55.26	48.68	42.32
Average	47.69	44.07	50.98	45.72	40.28

Table 4. Error rates for different meta-learners combining case base properties with $\varepsilon = 0.01$ and various DCT strategies

Meta-learner	DCT_b	DCT_{bd}	DCT_{bm}	DCT_{bmd}	DCT_{com}
Default Class	63.16	63.16	63.16	63.16	63.16
c50boost	40.78	35.52	42.11	34.21	36.21
c50tree	38.15	39.47	40.79	40.79	42.11
IB	51.31	43.32	55.26	47.36	42.11
Ripper	61.84	56.57	53.94	47.36	44.32
Average	48.02	43.72	48.03	42.43	41.19

Table 5. Error rates for different meta-learners combining case base properties with $\varepsilon = 0.05$ and various DCT strategies

Meta-learner	DCT_b	DCT_{bd}	DCT_{bm}	DCT_{bmd}	DCT_{com}
Default Class	63.16	63.16	63.16	63.16	63.16
c50boost	48.68	39.47	43.42	38.15	37.24
c50tree	40.79	43.42	44.73	43.42	43.16
IB	57.89	46.05	59.21	48.68	44.73
Ripper	56.57	59.21	57.89	55.26	43.42
Average	50.98	47.03	51.31	46.38	42.13

Table 6. Error rates for different meta-learners combining case base properties with $\varepsilon = 0.1$ and various DCT strategies

Tables 4 through 6 show error rates of meta-learners on combined measures from DCT and the case base approach. Although results on the average deteriorate, they are still quite similar when compared to table 2. This seems in particular interesting, since we added a total of ten variables from the case base approach to the different DCT approaches and the meta-data set consists only of 78 observations. It is our believe that by choosing the right mixture of DCT and case base measures, we might improve meta-learning, although maybe not significantly. Encouraged by our results, we tried to evaluate them using all learners as base and as meta-learners. The results for DCT strategies are given in table 7. On average, all methods perform better than the default. The missing values could not be computed. Table 8 shows the results for the landmarking and case-based reasoning approaches. Again, methods perform better than the default on average, though sometimes close to it. Tables 9 through 11 show various combinations of DCT strategies and case base measures. Again, the results of the learners on average are not much different from the case when using only DCT. This is particularly true for $\varepsilon = 0.1$.

5 Conclusions and Future Work

We have presented a new approach for task description as a means of model selection in meta-learning. Tasks are described by their similarity, consistency, incoherency, uniqueness and minimality. While this method does not outperform any of the existing approaches on its own, combinations of methods seem very

Meta-learner	DCT_b	DCT_{bd}	DCT_{bm}	DCT_{bdm}	DCT_{com}
Default Class	77.63	77.63	77.63	77.63	77.63
C5.0boost	67.11	**61.84**	64.47	63.16	63.16
C5.0rules	65.79	67.11	**61.84**	64.47	64.48
C5.0tree	67.11	69.74	**64.48**	65.79	**64.48**
ClemMLP	77.61	**73.68**	80.26	77.63	77.63
ClemRBFN	68.08	?	**62.91**	?	78.19
LinDiscr	**68.42**	76.32	75.00	78.95	78.95
Ltree	**67.11**	68.42	**67.11**	**67.11**	72.37
IB	**64.47**	68.42	65.79	69.74	68.42
NB	73.68	**69.74**	77.63	76.32	86.84
Ripper	**64.47**	69.74	69.73	68.43	68.42
Average	**68.38**	69.44	68.92	70.18	72.29

Table 7. Percentage error rates for DCT strategies using all learners as base and meta-learners

Meta-learner	Landmarking	Ext-Land	$CBP_{0.01}$	$CBP_{0.05}$	$CBP_{0.1}$
Default Class	77.63	77.63	77.63	77.63	77.63
C5.0boost	**61.84**	78.95	68.42	75.00	71.05
C5.0rules	**65.79**	78.95	68.42	75.00	71.05
C5.0trees	**64.47**	77.63	69.73	73.68	71.05
MLP	80.26	**77.63**	80.26	80.26	78.94
RBFN	70.58	80.26	?	84.21	75.00
LinDiscr	84.21	75.00	72.37	73.68	72.37
Ltree	**64.47**	76.31	73.68	77.68	71.05
IB	**64.47**	68.42	71.05	80.26	77.63
NB	73.68	77.63	80.26	71.05	76.31
Ripper	71.05	75.00	76.31	78.95	73.68
Average	70.08	76.58	73.39	76.97	73.82

Table 8. Error rates for different meta-learners and task description strategies using all learners as base and meta-learners

Meta-learner	DCT_b	DCT_{bd}	DCT_{bmd}
Default Class	77.63	77.63	77.63
C5.0boost	61.84	61.84	65.79
C5.0rules	68.42	67.11	68.42
C5.0trees	68.42	67.11	67.11
MLP	80.26	81.58	78.95
RBFN	81.58	81.58	78.95
LinDiscr	72.37	73.68	69.74
Ltree	71.05	71.05	69.74
IB	68.42	71.05	75.00
NB	73.68	73.68	75.00
Ripper	68.42	71.05	72.37
Average	71.44	71.97	72.11

Table 9. Case base properties using $\varepsilon = 0.01$ and various DCT strategies using all learners as base and meta-learners

Meta-learner	DCT_b	DCT_{bd}	DCT_{bmd}
Default Class	77.63	77.63	77.63
C5.0boost	60.52	61.84	60.53
C5.0rules	63.16	64.47	65.79
C5.0trees	67.11	67.11	67.11
MLP	80.26	77.63	81.58
RBFN	76.31	76.31	76.32
LinDiscr	73.68	75.00	73.61
Ltree	68.42	69.73	69.74
IB	67.11	68.42	73.68
NB	68.42	67.11	67.11
Ripper	73.68	71.05	73.68
Average	69.87	69.87	70.92

Table 10. Case base properties using $\varepsilon = 0.05$ and various DCT strategies using all learners as base and meta-learners

Meta-learner	DCT_b	DCT_{bd}	DCT_{bmd}
Default Class	77.63	77.63	77.63
C5.0boost	59.21	63.16	63.16
C5.0rules	63.16	63.16	63.16
C5.0trees	63.16	64.47	63.16
MLP	78.95	78.95	77.63
RBFN	85.52	82.83	84.21
LinDiscr	73.68	77.63	77.63
Ltree	60.53	61.84	61.82
IB	69.73	69.73	69.73
NB	71.05	69.73	67.11
Ripper	71.05	69.73	76.31
Average	69.61	70.13	70.39

Table 11. Case base properties using $\varepsilon = 0.1$ and various DCT strategies using all learners as base and meta-learners

promising, in particular for real-world data. Using case base properties might also help in understanding why methods perform differently. However, this serves as an outlook for future work. We also intend to use larger data sets for creating our meta-data set and to eventually use a larger meta-data set itself. Additionally, we want to evaluate useful combinations including landmarkers as well as testing which distance measure is most appropriate for meta-learning. One problem to overcome is the computational complexity of the case base properties. Since the complexity is quadratic, we think about drawing samples of smaller sizes, as the size of data sets increases. In general, this seems to be an interesting field of research.

Acknowledgements

The authors would like to thank Thomas Reinartz from DaimlerChrysler and the members of the METAL consortium for fruitful discussions. The research was supported financially by EC METAL project (ESPRIT # 26.357) and DaimlerChrysler.

References

1. Hilan Bensusan. *Automatic Bias Learning: An Inquiry Into The Inductive Basis of Induction*. PhD thesis, School of Cognitive and Computing Sciences, University of Sussex, UK, 1999.
2. Hilan Bensusan and Christophe Giraud-Carrier. Casa Batló is in Passeign de Gràcia or landmarking the expertise space. In *Proceedings of the Meta-Learning Workshop at the ECML-2000*, 2000.

3. Hilan Bensusan and Alexandros Kalousis. Estimating the predictive accuracy of a classifier. In Luc De Raedt and Peter Flach, editors, *Proceedings of the Twelfth European Conference on Machine Learning ECML-2001*. Springer, New York, NY, 2001.
4. C.L. Blake and C.J. Merz. UCI repository of machine learning databases. University of California, Irvine, Dept. of Information and Computer Sciences, 1998. http://www.ics.uci.edu/∼mlearn/MLRepository.html.
5. William W. Cohen. Fast effective rule induction. In *Proceedings of the 12th International Conference on Machine Learning*, pages 115–123. Morgan Kaufman, San Mateo, CA, 1995.
6. Richard O. Duda and Peter E. Hart. *Pattern Classification and Scene Analysis*. John Wiley & Sons, New York, NY, 1973.
7. Robert Engels and Christiane Theusinger. Using a data metric for offering preprocessing advice in data mining applications. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence*, pages 430–434, 1998.
8. João Gama. Discriminant trees. In *Proceedings of the Sixteenth International Conference on Machine Learning – ICML'99*, 1999.
9. Ioannis Iglezakis and Thomas Reinartz. Relation between customer requirements, performance measures, and general case properties for case base maintenance. In *Proceedings of the Sixth European Workshop on Case-Base Maintenance*, 2002.
10. William R. Klecka. *Discriminant Analysis*. Sage Publications, Newbury Park, London, UK, 1980.
11. Christian R. Köpf, Charles C. Taylor, and Jörg Keller. Meta-learning: From data characterisation for meta-learning to meta-regression. In *Proceedings of the Workshop on "Data Mining, Decision Support, Meta-Learning and ILP: Forum for Practical Problems" at the PKDD-2000*, 2000.
12. Guido Lindner and Rudi Studer. AST: Support for algorithm selection with a cBR approach. In *Proceedings of the Third International Conference on Princinples and Practice of Knowledge Discovery in Databases*, pages 418–423, 1999.
13. MetaL. EC ESPRIT MetaL Project #26.357. http://www.cs.bris.ac.uk/cgc/METAL, 1998–2001.
14. Donald Michie, David J. Spiegelhalter, and Charles C. Taylor, editors. *Machine Learning, Neural and Statistical Classification*. Ellis Horwood, New York, NY, 1994.
15. Bernhard Pfahringer, Hilan Bensusan, and Christophe Giraud-Carrier. Meta-learning by landmarking various learning algorithms. In *Proceedings of the 17th International Conference on Machine Learning*, 2000.
16. J. Ross Quinlan. Induction of decision trees. In *Proceedings of the First International Conference on Machine Learning*, pages 81–106. Morgan Kaufman, San Mateo, CA, 1986.
17. J. Ross Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufman, San Mateo, CA, 1992.
18. Thomas Reinartz, Ioannis Iglezakis, and Thomas Roth-Berghofer. On quality measures for case base maintenance. In *Proceedings of the Fifth European Workshop on Case-Base Maintenance*, pages 247–259. Springer, New York, NY, 2001.
19. Claude E. Shannon and Warren Weaver. *The Mathematical Theory of Communication*. University of Illinois Press, Urbana and Chicago, IL, 1963.

Rule induction for subgroup discovery with CN2-SD

Nada Lavrač[1], Peter Flach[2], Branko Kavšek[1], and Ljupčo Todorovski[1]

[1] Institute Jožef Stefan, Jamova 39, 1000 Ljubljana, Slovenia
{Nada.Lavrac,Branko.Kavsek,Ljupco.Todorovski}@ijs.si
[2] University of Bristol, Bristol, UK
Peter.Flach@bristol.ac.uk

Abstract. Rule learning is typically used in solving classification and prediction tasks. However, learning of classification rules can be adapted also to subgroup discovery. This paper shows how this can be achieved by modifying the CN2 rule learning algorithm. Modifications include a new covering algorithm (weighted covering algorithm), a new search heuristic (weighted relative accuracy), probabilistic classification of instances, and a new measure for evaluating the results of subgroup discovery (area under ROC curve). The main advantage of the proposed approach is that each rule with high weighted accuracy represents a 'chunk' of knowledge about the problem, due to the appropriate tradeoff between accuracy and coverage, achieved through the use of the weighted relative accuracy heuristic. Moreover, unlike the classical covering algorithm, in which only the first few induced rules may be of interest as subgroup descriptors with sufficient coverage (since subsequently induced rules are induced from biased example subsets), the subsequent rules induced by the weighted covering algorithm allow for discovering interesting subgroup properties of the entire population. Experimental results on 17 UCI datasets are very promising, demonstrating big improvements in number of induced rules, rule coverage and rule significance, as well as smaller improvements in rule accuracy and area under ROC curve.

1 Introduction

Classical rule learning algorithms were designed to construct classification and prediction rules [5, 11]. In addition to this area of machine learning, referred to as *predictive induction*, developments in *descriptive induction* have recently gained much attention. These involve mining of association rules (e.g., the APRIORI association rule learning algorithm [1]), subgroup discovery (e.g., the MIDOS subgroup discovery algorithm [17]), and other approaches to non-classificatory induction.

The methodology presented in this paper can be applied to subgroup discovery. As in the MIDOS approach, a subgroup discovery task can be defined as follows: given a population of individuals and a property of those individuals we are interested in, find population subgroups that are statistically 'most

interesting', e.g., are as large as possible and have the most unusual statistical (distributional) characteristics with respect to the property of interest.

This paper investigates how to adapt classical classification rule learning approaches to subgroup discovery, by exploiting the information about class membership in training examples. This paper shows how this can be achieved by appropriately modifying the well-known CN2 rule learning algorithm [4, 5, 3], which we have implemented in Java and incorporated in the WEKA data mining environment [16]. The modified CN2 algorithm and its experimental evaluation in selected domains of the UCI Repository of Machine Learning Databases [12] are outlined. The experimental results are very promising, demonstrating big improvements in number of induced rules, rule coverage and rule significance, as well as smaller improvements in rule accuracy.

This paper is organized as follows. In Section 2 the background for this work is explained: the standard CN2 rule induction algorithm, including the covering algorithm and standard CN2 heuristics, as well as the weighted relative accuracy heuristic and probabilistic classification. Section 3 presents the modified CN2 algorithm, called CN2-SD, adapting the CN2 algorithm for subgroup discovery. Section 4 presents the experimental evaluation in selected UCI domains. Section 5 concludes by summarizing the results and presenting plans for further work.

2 Background

This section presents the backgrounds: classical CN2 rule induction algorithm, including the covering algorithm and standard CN2 heuristics, as well as the weighted relative accuracy heuristic, probabilistic classification and rule evaluation in the ROC space.

The CN2 Rule Induction Algorithm. CN2 is an algorithm for inducing propositional classification rules [4, 5]. CN2 consists of two main procedures: the search procedure that performs beam search in order to find a single rule and the control procedure that repeatedly executes the search.

The search procedure performs beam search using classification accuracy of the rule as a heuristic function. The accuracy of the propositional classification rule `if` *Cond* `then` *Class* is equal to the conditional probability of class *Class*, given that the condition *Cond* is satisfied: $Acc(\texttt{if}\ Cond\ \texttt{then}\ Class) = p(Class|Cond)$.

We replaced the accuracy measure with the weighted relative accuracy, defined in Equation 1 below. Furthermore, different probability estimates, like the Laplace [3] or the m-estimate [2, 6], can be used in CN2 for estimating the above probability and the probabilities in Equation 1. The standard CN2 algorithm used in this work uses the Laplace estimate.

Additionally, CN2 can apply a significance test to the induced rule. The rule is considered to be significant, if it locates regularity unlikely to have occurred by chance. To test significance, CN2 uses the likelihood ratio statistic [5] that

measures the difference between the class probability distribution in the set of examples covered by the rule and the class probability distribution in the set of all training examples. Empirical evaluation in [3] shows that applying a significance test reduces the number of induced rules (and also slightly reduces the predictive accuracy).

Two different control procedures are used in CN2: one for inducing an ordered list of rules and the other for the unordered case. When inducing an ordered list of rules, the search procedure looks for the best rule, according to the heuristic measure, in the current set of training examples. The rule predicts the most frequent class in the set of examples, covered by the induced rule. Before starting another search iteration, all examples covered by the induced rule are removed. The control procedure invokes a new search, until all the examples are covered.

In the unordered case, the control procedure is iterated, inducing rules for each class in turn. For each induced rule, only covered examples belonging to that class are removed, instead of removing all covered examples, like in the ordered case. The negative training examples (i.e., examples that belong to other classes) remain and positives are removed in order to prevent CN2 finding the same rule again.

The Weighted Relative Accuracy Heuristic. Weighted relative accuracy can be meaningfully applied both in the descriptive and predictive induction framework; in this paper we apply this heuristic for subgroup discovery.

We use the following notation. Let $n(Cond)$ stand for the number of instances covered by a rule $Class \leftarrow Cond$, $n(Class)$ stand for the number of examples of class $Class$, and $n(Class.Cond)$ stand for the number of correctly classified examples (true positives). We use $p(Class.Cond)$ etc. for the corresponding probabilities. We then have that rule accuracy can be expressed as $Acc(Class \leftarrow Cond) = p(Class|Cond) = \frac{p(Class.Cond)}{p(Cond)}$. Weighted relative accuracy [10, 15] is defined as follows.

$$WRAcc(Class \leftarrow Cond) = p(Cond).(p(Class|Cond) - p(Class)). \quad (1)$$

Weighted relative accuracy consists of two components: generality $p(Cond)$, and relative accuracy $p(Class|Cond) - p(Class)$. The second term, relative accuracy, is the accuracy gain relative to the fixed rule $Class \leftarrow true$. The latter rule predicts all instances to satisfy $Class$; a rule is only interesting if it improves upon this 'default' accuracy. Another way of viewing relative accuracy is that it measures the utility of connecting rule body $Cond$ with a given rule head $Class$. However, it is easy to obtain high relative accuracy with highly specific rules, i.e., rules with low generality $p(Cond)$. To this end, generality is used as a 'weight', so that weighted relative accuracy trades off generality of the rule ($p(Cond)$, i.e., rule coverage) and relative accuracy ($p(Class|Cond) - p(Class)$).

Probabilistic Classification. The induced rules can be ordered or unordered. Ordered rules are interpreted as a decision list [14] in a straight-forward manner:

when classifying a new example, the rules are sequentially tried and the first rule that covers the example is used for prediction.

In the case of unordered rule sets, the distribution of covered training examples among classes is attached to each rule. Rules of the form:

$$\text{if } Cond \text{ then } Class \ [ClassDistribution]$$

are induced, where numbers in the *ClassDistribution* list denote, for each individual class, how many training examples of this class are covered by the rule. When classifying a new example, all rules are tried and those covering the example are collected. If a clash occurs (several rules with different class predictions cover the example), a voting mechanism is used to obtain the final prediction: the class distributions attached to the rules are summed to determine the most probable class. If no rule fires, a default rule is invoked which predicts the majority class of uncovered training instances.

3 The CN2-SD Algorithm for Subgroup Discovery

The main modifications of the CN2 algorithm, making it appropriate for subgroup discovery, involve the implementation of the weighted covering algorithm, incorporation of example weights into the weighted relative accuracy heuristic, probabilistic classification also in the case of the 'ordered' induction algorithm, and area under ROC curve rule set evaluation.

The Weighted Covering Algorithm. In the classical covering algorithm only the first few induced rules may be of interest as subgroup descriptors with sufficient coverage, since subsequently induced rules are induced from biased example subsets, i.e., subsets including only positive examples not covered by previously induced rules. This bias constrains the population for subgroup discovery in a way that is unnatural for the subgroup discovery process which is, in general, aimed at discovering interesting properties of subgroups of the entire population. In contrast, the subsequent rules induced by the weighted covering algorithm allow for discovering interesting subgroup properties of the entire population.

The weighted covering algorithm is modified in such a way that covered positive examples are not deleted from the current training set. Instead, in each run of the covering loop, the algorithm stores with each example a count how many times (with how many rules induced so far) the example has been covered. Weights derived from these example counts then appear in the computation of *WRAcc*. We have implemented two approaches.

Multiplicative weights. In the first approach, weights decrease multiplicatively. For a given parameter $\gamma < 1$, weights of covered examples decrease as follows: $e(i) = \gamma^i$, where $e(i)$ is the weight of an example being covered i times. Note that the weighted covering algorithm with $\gamma = 1$ would result in finding the same rule over and over again, whereas with $\gamma = 0$ the algorithm would perform the same as the standard CN2 algorithm.

Additive weights. In the second approach, weights of covered examples are modified as follows: $e(i) = \frac{1}{i+1}$.

Modified WRAcc Heuristic with Example Weights. The modification of CN2 reported in [15] affected only the heuristic function: weighted relative accuracy was used as search heuristic, instead of the accuracy heuristic of the original CN2, while everything else stayed the same. In this work, the heuristic function was further modified to enable handling example weights, which provide the means to consider different parts of the instance space in each iteration of the weighted covering algorithm.

In the *WRAcc* computation (Equation 1) all probabilities are computed by relative frequencies. An example weight measures how important it is to cover this example in the next iteration. The initial example weight $e(0) = 1$ means that the example hasn't been covered by any rule, meaning 'please cover this example, it hasn't been covered before', while lower weights mean 'don't try too hard on this example'. The modified *WRAcc* measure is then defined as follows

$$WRAcc(Class \leftarrow Cond) = \frac{n'(Cond)}{N'}(\frac{n'(Class.Cond)}{n'(Cond)} - \frac{n'(Class)}{N'}). \quad (2)$$

where N' is the sum of the weights of all examples, $n'(Cond)$ is the sum of the weights of all covered examples, and $n'(Class.Cond)$ is the sum of the weights of all correctly covered examples.

Probabilistic classification. Each CN2 rule returns a class distribution in terms of numbers of examples covered, as distributed over classes. The CN2 algorithm uses class distribution in classifying unseen instances only in the case of unordered rule sets, where rules are induced separately for each class. In the case of ordered decision lists, the first rule that fires provides the classification. In our modified CN2-SD algorithm, the same probabilistic classification is used in both classifiers, due to overlapping rules. This means that the terminology 'ordered' and 'unordered', which in CN2 distinguished between decision list and rule set induction, has a different meaning in our setting: the 'unordered' algorithm refers to learning classes one by one, while the 'ordered' algorithm refers to finding best rule conditions and assigning the majority class in the head.

4 Experimental evaluation

We experimentally evaluated our approach on 17 data sets from the UCI Repository of Machine Learning Databases [12]. In Table 1, the selected data sets are summarised in terms of the number of attributes, the number of examples, and the percentage of examples of the majority class. These data sets have been widely used in other comparative studies. Since our re-implementation of CN2 currently does not support continuous attributes and can not handle missing

values, all continuous attributes have been discretised and data sets that contain no missing values have been chosen. The discretisation described in [8] was performed using the WEKA tool [16]. Moreover, all of the data sets have two classes, either originally or by selecting one class as 'positive' and joining all the other in a 'negative' class (in Table 1, the selected positive class is indicated by [{ClassName}]); this was done for the purpose of enabling the area under ROC curve evaluation.

Table 1. Characteristics of data sets used in the experiments.

#	Data set	#Attributes	#Examples	Majority class (%)
1	Anneal[{3}]	38	898	76.16
2	Australian	14	690	55.5
3	Balance[{L}]	4	625	46.08
4	Car[{unacc}]	6	1728	70.02
5	Credit-g	20	1000	70
6	Diabetes	8	768	65.1
7	Glass[{build wind non-float}]	9	214	35.51
8	Heart-stat	13	270	55.56
9	Ionosphere	34	351	64.1
10	Iris[{Iris-setosa}]	4	150	33.33
11	Lymph[{metastases}]	18	148	54.72
12	Segment[{brickface}]	19	2310	14.29
13	Sonar	60	208	53.36
14	Tic-tac-toe	9	958	65.34
15	Vehicle[{bus}]	18	846	25.77
16	Wine[{2}]	13	178	39.89
17	Zoo[{mammal}]	17	101	40.59

The performance of different variants of the CN2 rule induction algorithm was measured using 10-fold stratified cross-validation. In particular, we compared the CN2-SD subgroup discovery algorithm with the standard CN2 algorithm (*CN2-standard*, described in [4, 5, 3]) and the CN2 algorithm using *WRAcc* (*CN2-WRAcc*, described in [15]). All these variants of the CN2 algorithm were first re-implemented in the WEKA data mining environment [16], because the use of the same system makes the comparisons more impartial.

The results of these comparisons are presented in Tables 2 and 3, comparing *CN2-SD* with *CN2-standard* and *CN2-WRAcc* in terms of accuracy (Table 2), and size of the rule set (number of rules including the default rule), average example coverage and likelihood ratio[1] per rule (Table 3). Tables for the ordered algorithm are skipped due to space restrictions, and due to the fact that the unordered algorithm is better suited to the philosophy of subgroup discovery due to its aim at inducing independent individual rules. The results of the *CN2-SD* algorithm were computed using both the multiplicative weights (with $\gamma = 0.5, 0.7, 0.9$) and the additive weights. All other parameters of the CN2 algorithm were set to their default values (beam-size = 5, significance-threshold = 99%).

The experimental results show that *CN2-SD* achieves improvements across the board. Additive weights result in about half the number of rules on av-

[1] The likelihood ratio is used in CN2 for testing the significance of the induced rule [5]. For two-class problems this statistic is distributed approximately as χ^2 with one degree of freedom.

Table 2. Accuracy with standard deviation ($Acc \pm sd$) for different variants of the unordered algorithm.

#	**CN2 standard** $Acc \pm sd$	**CN2 WRAcc** $Acc \pm sd$	**CN2-SD** ($\gamma = 0.5$) $Acc \pm sd$	**CN2-SD** ($\gamma = 0.7$) $Acc \pm sd$	**CN2-SD** ($\gamma = 0.9$) $Acc \pm sd$	**CN2-SD (add. weight.)** $Acc \pm sd$
1	98.33 ± 0.11	94.54 ± 0.20	94.77 ± 0.19	95.21 ± 0.19	93.88 ± 0.21	94.65 ± 0.21
2	38.55 ± 0.53	85.51 ± 0.35	84.93 ± 0.35	84.93 ± 0.35	84.78 ± 0.35	84.93 ± 0.35
3	75.68 ± 0.39	81.76 ± 0.38	85.12 ± 0.38	86.40 ± 0.38	86.40 ± 0.38	83.68 ± 0.39
4	97.74 ± 0.11	95.08 ± 0.33	95.14 ± 0.33	90.28 ± 0.32	89.53 ± 0.33	85.53 ± 0.34
5	74.40 ± 0.43	69.90 ± 0.43	70.70 ± 0.43	70.80 ± 0.43	70.50 ± 0.43	69.90 ± 0.43
6	68.62 ± 0.45	72.79 ± 0.42	72.14 ± 0.42	73.18 ± 0.42	74.22 ± 0.42	72.92 ± 0.42
7	80.37 ± 0.38	79.91 ± 0.40	68.22 ± 0.46	69.16 ± 0.45	69.63 ± 0.45	68.69 ± 0.46
8	66.30 ± 0.47	71.85 ± 0.46	76.67 ± 0.41	78.52 ± 0.39	81.11 ± 0.39	78.15 ± 0.41
9	85.76 ± 0.33	85.76 ± 0.33	86.04 ± 0.33	86.89 ± 0.31	87.75 ± 0.31	83.48 ± 0.34
10	99.33 ± 0.05	99.33 ± 0.05	100.00 ± 0.07	99.33 ± 0.10	99.33 ± 0.10	98.00 ± 0.14
11	86.49 ± 0.33	75.68 ± 0.39	83.78 ± 0.37	83.11 ± 0.37	83.11 ± 0.37	81.08 ± 0.38
12	90.22 ± 0.26	87.88 ± 0.31	97.71 ± 0.13	97.71 ± 0.13	97.58 ± 0.15	97.53 ± 0.15
13	71.15 ± 0.49	61.06 ± 0.50	66.83 ± 0.49	67.79 ± 0.47	67.31 ± 0.48	65.38 ± 0.48
14	98.33 ± 0.08	70.56 ± 0.42	84.45 ± 0.38	85.07 ± 0.38	88.41 ± 0.37	83.92 ± 0.39
15	87.47 ± 0.29	80.73 ± 0.36	89.60 ± 0.33	89.95 ± 0.33	90.19 ± 0.33	88.89 ± 0.34
16	85.39 ± 0.33	91.57 ± 0.27	93.26 ± 0.25	93.82 ± 0.25	93.82 ± 0.25	92.13 ± 0.29
17	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00
Average	**82.60 ± 0.30**	**82.58 ± 0.33**	**85.26 ± 0.31**	**85.42 ± 0.31**	**85.74 ± 0.31**	**84.05 ± 0.33**

Table 3. Average size (S), coverage (CVG) and likelihood ratio (LHR) of rules for different versions of the unordered algorithm.

#	**CN2 standard**			**CN2 WRAcc**			**CN2-SD** ($\gamma = 0.5$)			**CN2-SD** ($\gamma = 0.7$)			**CN2-SD** ($\gamma = 0.9$)			**CN2-SD (add. weight.)**		
	S	CVG	LHR	S	CVG	LHR	S	CVG	LHR	S	CVG	LHR	S	CVG	LHR	S	CVG	LHR
1	26	49.3	68.8	26	58.1	61.2	14	115.6	100.9	14	126.3	130.7	13	190.7	136.1	8	150.5	193.0
2	58	36.0	21.5	6	156.8	89.9	10	181.0	136.6	9	239.7	170.5	8	296.0	189.8	6	269.7	211.6
3	113	9.5	11.6	42	24.7	20.2	17	75.0	28.8	18	72.0	31.3	11	125.0	38.0	9	105.0	43.8
4	84	30.9	45.7	22	128.1	112.9	11	253.2	136.1	11	282.0	167.0	11	422.4	167.0	6	282.0	212.3
5	91	15.1	13.2	14	98.7	25.2	13	151.0	37.9	12	185.1	47.9	15	263.0	48.9	7	191.5	55.4
6	58	26.5	13.2	12	90.6	27.7	11	113.7	39.7	14	102.3	37.0	12	132.0	40.0	9	116.1	42.8
7	23	11.9	12.2	15	16.5	11.9	11	39.8	14.6	15	35.5	15.0	17	62.0	16.1	7	35.1	18.1
8	42	14.6	14.2	11	57.3	18.4	16	51.8	29.4	16	69.6	36.4	20	79.7	36.4	11	66.1	42.4
9	42	19.7	19.5	26	23.5	21.5	27	40.6	39.7	25	47.7	44.9	26	63.0	43.6	13	49.6	52.4
10	11	16.3	30.0	11	16.3	30.0	14	21.8	27.4	14	21.8	27.4	14	24.4	27.4	10	21.8	33.8
11	17	14.6	18.2	10	21.3	19.9	16	27.1	24.1	16	29.2	24.1	23	39.3	25.1	10	28.2	30.7
12	184	21.6	94.6	38	103.2	139.4	11	337.1	345.1	8	398.5	390.0	7	440.0	437.1	6	407.0	509.6
13	36	7.8	12.5	22	15.8	13.5	28	19.4	13.7	32	20.8	14.7	41	34.8	14.6	12	24.0	17.9
14	30	38.9	76.4	27	55.5	44.0	20	83.7	62.6	18	94.2	63.4	15	117.6	74.9	11	101.8	68.2
15	82	19.6	32.7	38	34.1	28.3	14	154.6	101.3	14	166.3	107.3	15	218.0	107.3	9	189.7	131.5
16	28	10.0	16.0	18	13.8	20.5	21	20.0	19.8	20	20.9	20.0	21	27.6	20.5	11	21.9	25.5
17	3	50.5	68.2	3	50.5	68.2	3	50.5	68.2	3	50.5	68.2	3	50.5	68.2	3	50.5	68.2
Avg	**54.6**	**23.1**	**33.5**	**20.0**	**56.8**	**44.3**	**15.1**	**102.1**	**72.2**	**15.2**	**115.5**	**82.1**	**16.0**	**152.1**	**87.8**	**8.7**	**124.2**	**103.4**

erage obtained by multiplicative weights. Average rule coverage is optimal for multiplicative weights with high γ, improving on the average coverage of *CN2-standard* rules with a factor of 6 and on *CN2-WRAcc* with a factor of 3. We conclude that both rules obtained with additive weights and with multiplicative weights with high γ are highly overlapping, due to the relatively modest decrease of example weights.

In addition, there is also a big increase in the average likelihood ratio: while the ratios achieved by *CN2-standard* are already significant at the 99% level,

this is further pushed up by *CN2-SD* with maximum values achieved by additive weights. An interesting question, to be verified with further experiments, is whether the weighted versions of the CN2 algorithm improve the significance of the induced subgroups also in the case when CN2 rules are induced without applying the significance test.

In summary, *CN2-SD* produces substantially smaller rule sets, where individual rules have higher coverage and significance. These three factors are important for subgroup discovery: smaller size enables better understanding, higher coverage means larger support, and rules should describe discovered subgroups that are significantly different from the entire population.

The increased accuracy of *CN2-SD* compared to *CN2-standard* and *CN2-WRAcc* (see Table 2) improves on the findings in [15], where the rule size decreased at the expense of a small drop in accuracy. It should be noted that the results of *CN2-standard* and *CN2-WRAcc* cannot be directly compared to those reported in [15] due to the following reasons: first, different datasets were selected in the two experiments, second, attribute discretisation was performed, third, minor differences in the algorithm implementations exist, and finally, results in this paper were obtained for binarised learning problems. Our hypothesis, that needs to be verified in further work, is that the improved results reported in this paper may be due to the binarised problem domains for which *WRAcc* may be better suited than for multi-class domains.

5 Conclusions

We have presented a novel approach to adapting standard classification rule learning to subgroup discovery. To this end we have appropriately adapted the covering algorithm, the search heuristics and the probabilistic classification procedure. Experimental results on 17 UCI datasets are very promising, demonstrating big improvements in number of induced rules, rule coverage and rule significance, as well as smaller improvements in rule accuracy.

In further work we will investigate the behaviour of *CN2-SD* in multi-class problems. We are also planning to evaluate the approach using the area under the ROC convex hull metric which is more appropriate for subgroup discovery than the standard accuracy metric. See the appendix for some ROC results. Finally, we plan to use our adapted procedure for subgroup discovery for solving practical problems, in which expert evaluations of induced subgroup descriptions will be of ultimate interest.

Acknowledgements

Thanks to Dragan Gamberger for inspiring the work on a weighted covering algorithm. The work reported in this paper was supported by the Slovenian Ministry of Education, Science and Sport, the IST-1999-11495 project Data Mining and Decision Support for Business Competitiveness: A European Virtual Enterprise, and the British Council project Partnership in Science PSP-18.

References

1. Agrawal, R., Mannila, H., Srikant, R., Toivonen, H., & Verkamo, A.I. (1996) Fast discovery of association rules. In U.M. Fayyad, G. Piatetski-Shapiro, P. Smyth and R. Uthurusamy (Eds.) *Advances in Knowledge Discovery and Data Mining* (pp. 307–328). AAAI Press.
2. B. Cestnik. Estimating probabilities: A crucial task in machine learning. In L. Aiello, editor, *Proceedings of the 9th European Conference on Artificial Intelligence*, pp. 147–149, Pitman, Stockholm, Sweden, 1990.
3. Clark, P. and Boswell, R. (1989). Rule induction with CN2: Some recent improvements. In Y. Kodratoff, editor, *Proceedings of the 5th European Working Session on Learning*, Springer-Verlag, 151–163.
4. P. Clark and T. Niblett. Induction in noisy domains. In I. Bratko and N. Lavrač, editors, *Progress in Machine Learning (Proceedings of the 2nd European Working Session on Learning)*, pp. 11–30, Sigma, Wilmslow, UK, 1987.
5. Clark, P. and Niblett, T. (1989). The CN2 induction algorithm. *Machine Learning*, Kluwer, 3(4):261–283.
6. Džeroski, S., Cestnik, B. and Petrovski, I. (1993) Using the m-estimate in rule induction. Journal of Computing and Information Technology, 1(1):37 – 46.
7. Ferri-Ramírez, C., Flach, P. and Hernandez-Orallo, J. (2002) Learning Decision Trees Using the Area Under the ROC Curve. *Proceedings of the 19th International Conference on Machine Learning*, Morgan Kaufmann, in press.
8. Fayyad, U.M. and Irani, K.B. (1993). Multi-interval discretisation of continuous-valued attributes for classification learning. In Bajcsy, R. (Ed.) *Proceedings of the 13th International Joint Conference on Artificial Intelligence*, Morgan Kaufmann, 1022–1027.
9. Kukar, M., Kononenko, I., Grošelj, C., Kralj, K., & Fettich, J.J. (1998) Analysing and improving the diagnosis of ischaemic heart disease with machine learning. *Artificial Intelligence in Medicine*, special issue on *Data Mining Techniques and Applications in Medicine*, 16, 25–50. Elsevier.
10. Lavrač, N., Flach, P. and Zupan, B. (1999) Rule Evaluation Measures: A Unifying View. In *Proceedings of the Ninth International Workshop on Inductive Logic Programming, volume 1634 of Lecture Notes in Artificial Intelligence*: 74–185. Springer-Verlag.
11. Michalski, R.S., Mozetič, I., Hong, J., & Lavrač, N. (1986) The multi-purpose incremental learning system AQ15 and its testing application on three medical domains. In *Proc. Fifth National Conference on Artificial Intelligence*, (pp. 1041–1045), Morgan Kaufmann.
12. Murphy, P.M. and Aha, D.W. (1994) *UCI repository of machine learning databases* [http://www.ics.uci.edu/ ˜mlearn/MLRepository.html]. Irvine, CA: University of California, Department of Information and Computer Science.
13. Provost, F. & Fawcett, T. (2001) Robust classification for imprecise environments. *Machine Learning*, 42(3), 203–231.
14. R. L. Rivest. Learning decision lists. *Machine Learning*, 2(3):229–246, Kluwer, 1987.
15. Todorovski, L., Flach, P. and Lavrač, N. (2000). Predictive Performance of Weighted Relative Accuracy. In Zighed, D.A., Komorowski, J. and Zytkow, J., editors, *Proceedings of the 4th European Conference on Principles of Data Mining and Knowledge Discovery (PKDD2000)*, Springer-Verlag, 255–264.

16. Witten, I.H. and Frank, E. (1999). *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations.* Morgan Kaufmann, San Francisco.
17. Wrobel, S. (1997) An algorithm for multi-relational discovery of subgroups. *Proc. First European Symposium on Principles of Data Mining and Knowledge Discovery*, (pp. 78–87), Springer.

Appendix: Area under ROC convex hull evaluation

A point on the *ROC curve* (ROC: Receiver Operating Characteristic) [9, 13] shows classifier performance in terms of false alarm or *false positive rate* $FPr = \frac{FP}{TN+FP}$ (plotted on the X-axis) that needs to be minimized, and sensitivity [2] or *true positive rate* $TPr = \frac{TP}{TP+FN}$ (plotted on the Y-axis) that needs to be maximized. In the ROC space, an appropriate tradeoff, determined by the expert, can be achieved by applying different algorithms, as well as by different parameter settings of a selected data mining algorithm or by taking into the account different misclassification costs. The ROC space is appropriate for measuring the success of subgroup discovery, since subgroups whose TPr/FPr tradeoff is close to the diagonal can be discarded as insignificant. The area under the ROC curve (AUC) can be used as a quality measure for comparing the success of different learners.

In subgroup discovery there are two ways in which a rule learner can give rise to a ROC curve.

AUC-Method-1. The first method treats each rule as a separate subgroup which is plotted in the ROC space with its true and false positive rates. We then calculate the convex hull of this set of points, selecting the subgroups which perform optimally under a particular range of operating characteristics. The area under this ROC convex hull (AUC) indicates the combined quality of the optimal subgroups.[3]

AUC-Method-2. The second method employs the combined probabilistic classifications of all subgroups, as indicated below. If we always choose the most likely predicted class, this corresponds to setting a fixed threshold 0.5 on the positive probability: if the positive probability is larger than this threshold we predict positive, else negative. A ROC curve can be constructed by varying this threshold from 1 (all predictions negative, corresponding to (0,0) in the ROC space) to 0 (all predictions positive, corresponding to (1,1) in

[2] *Sensitivity* measures the fraction of positive cases that are classified as positive, whereas *specificity* measures the fraction of negative cases classified as negative. If TP denotes true positives, TN true negatives, FP false positives, FN false negatives, Pos all positives, and Neg all negatives, then $Sensitivity = TPr = \frac{TP}{TP+FN} = \frac{TP}{Pos}$, and $Specificity = \frac{TN}{TN+FP} = \frac{TN}{Neg}$, and $FalseAlarm = FPr = 1 - Specificity = \frac{FP}{TN+FP} = \frac{FP}{Neg}$.

[3] In fact, we would have two convex hulls as some subgroups shift the distribution to the positive class and others shift it to the negative class. This method does not take account of overlapping subgroups.

the ROC space). This results in $n + 1$ points in the ROC space, where n is the total number of classified examples. Equivalently, we can order all the examples by decreasing predicted probability of being positive, and tracing the ROC curve by starting in (0,0), stepping up when the example is actually positive and stepping to the right when it is negative, until we reach (1,1).[4] The area under this ROC curve indicates the combined quality of all subgroups (i.e., the quality of the entire rules set). This method can be used with a test set or in cross-validation, but the resulting curve is not necessarily convex. A detailed description of this method can be found in [7].

Table 4. Area under the ROC curve with standard deviation ($AUC \pm sd$) for different variants of the unordered algorithm using 10-fold stratified cross-validation.

#	**CN2 standard** $AUC \pm sd$	**CN2 WRAcc** $AUC \pm sd$	**CN2-SD** ($\gamma = 0.5$) $AUC \pm sd$	**CN2-SD** ($\gamma = 0.7$) $AUC \pm sd$	**CN2-SD** ($\gamma = 0.9$) $AUC \pm sd$	**CN2-SD (add. weight.)** $AUC \pm sd$
1	99.41 ± 0.01	99.72 ± 0.00	99.24 ± 0.01	98.84 ± 0.01	98.51 ± 0.01	98.17 ± 0.01
2	35.10 ± 0.11	87.83 ± 0.05	83.15 ± 0.05	84.12 ± 0.04	84.32 ± 0.05	84.97 ± 0.04
3	86.22 ± 0.03	89.00 ± 0.03	93.89 ± 0.02	93.69 ± 0.02	93.56 ± 0.02	91.82 ± 0.03
4	99.93 ± 0.00	96.55 ± 0.02	94.67 ± 0.02	93.86 ± 0.02	93.00 ± 0.02	86.78 ± 0.02
5	70.10 ± 0.09	72.11 ± 0.06	71.38 ± 0.07	71.31 ± 0.07	72.68 ± 0.07	70.12 ± 0.06
6	69.52 ± 0.08	78.93 ± 0.05	79.89 ± 0.04	79.93 ± 0.05	80.14 ± 0.05	79.43 ± 0.05
7	68.23 ± 0.08	73.85 ± 0.12	70.71 ± 0.16	72.59 ± 0.15	72.91 ± 0.15	72.67 ± 0.14
8	74.75 ± 0.09	74.56 ± 0.07	82.96 ± 0.08	83.83 ± 0.11	86.16 ± 0.11	84.76 ± 0.09
9	93.81 ± 0.03	90.21 ± 0.06	90.66 ± 0.06	91.48 ± 0.06	91.80 ± 0.06	91.36 ± 0.05
10	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00
11	94.34 ± 0.04	89.16 ± 0.08	88.15 ± 0.07	91.14 ± 0.06	90.76 ± 0.06	88.53 ± 0.08
12	99.73 ± 0.01	99.79 ± 0.00	98.99 ± 0.01	98.69 ± 0.02	98.19 ± 0.02	98.05 ± 0.02
13	65.32 ± 0.12	60.61 ± 0.10	69.35 ± 0.13	72.04 ± 0.15	71.19 ± 0.16	65.10 ± 0.16
14	100.00 ± 0.00	81.00 ± 0.08	92.97 ± 0.03	92.37 ± 0.04	91.96 ± 0.04	90.24 ± 0.04
15	97.27 ± 0.02	92.41 ± 0.03	94.38 ± 0.03	94.60 ± 0.02	94.18 ± 0.02	93.43 ± 0.02
16	94.14 ± 0.05	96.30 ± 0.06	95.39 ± 0.05	95.53 ± 0.05	95.53 ± 0.05	92.16 ± 0.09
17	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00
Average	**85.17 ± 0.04**	**87.18 ± 0.05**	**88.58 ± 0.05**	**89.06 ± 0.05**	**89.11 ± 0.05**	**87.51 ± 0.05**

[4] In the case of ties, we make the appropriate number of steps up and to the right at once, drawing a diagonal line segment.

Large and Tall Buildings: A case study in the application of Decision Support and Data Mining

Steve Moyle[1], Marko Bohanec[2,3], and Eric Ostrowski[4]

[1] Oxford University Computing Laboratory, Oxford, UK
steve.moyle@comlab.ox.ac.uk
[2] Jožef Stefan Institute, Ljubljana, Slovenia
[3] University of Ljubljana, School of Public Administration, Slovenia-
marko.bohanec@ijs.si
[4] EC Harris, London, UK
eric.ostrowski@echarris.com

Abstract. Large and Tall buildings can be broadly classified into three groups: sprawling, squat, or tall. The decision to build a particular type of large building can be based on a large number of attributes. One set of possible values of the attributes represents a building design, which must be feasible. In addition, it is also important to understand how such a set of attributes impacts on the value of the proposed building to the customer – in particular the value of the proposed building design with respect to the client's value drivers. A building construction expert's analysis of seventy international building projects was used as input to decision support and data mining analyses. Decision models were developed that mapped customer values of proposed construction project's attributes. Data mining – on albeit limited examples – was used to model the feasibility of construction projects from their input attributes. On this basis, we propose a novel way of combining Data Mining and Decision Support methods: both techniques are employed to utilise the same input vectors, but one – Decision Support Models – is designed to assess and possibly maximise utility, while the other – Data Mining Models – provides a test for feasibility.

1. Introduction

The expertise of a large number of individuals is required to successfully complete any significant engineering project, including that of constructing a large building. A vast range of skills – including surveying, architectural design, structural engineering, and service facility design – are deployed in the building design phase alone. The very early phase of defining the project scope and broad specification can lead to difficult negotiations with the prospective owner of the building (referred to as the client). The client has a set of preferences about the type of building they desire – which may significantly impact on both the feasibility of the design and the cost of the project. Moreover different clients place different *values* on attributes of the building project.

In the early building specification phase, clients work with construction project experts to settle on a broad building design. This can require the selection of particular building attributes that are feasible, but also maximize the utility of the building for

the client. The construction experts need to be able to articulate the complex interaction between the attributes, to help guide the clients to a feasible and valuable design.

1.1 Large and tall buildings, and Value Drivers

The size and shape of a building has a dramatic impact on many aspects including capital cost, ease of design, the use of standard components, programme, logistics and whole life costs which all affect the rate of return to a client. Unlike many other industries, construction is a complex blend of disperate needs, skills and techniques that are difficult to co-ordinate.

The Capital Projects and Facilities Consultants *EC Harris* have recently completed a study examining the value drivers between different shapes of building. In order to understand the interaction between the different shapes and sizes, it was necessary to define 18 "virtual buildings" which are grouped by size (medium, large and very large), shape (squat, sprawling and tall) and finally quality (medium and high). The study of the 18 benchmarks uncovered the knowledge that buildings with 20 storeys are most likely to give the best return on investment.

Although value drivers have been talked about over the years, they have not really been properly accounted for within the construction industry as a whole. Instead, most of the focus in this general area has been on establishing key performance indicators that can be easily measured, a fact that to some extent explains the approach to date.

In order to understand the purpose of undertaking this particular research into value drivers, it is necessary to first look at the fundamental difference between value and performance. Put simply, the value of any aspect of a construction project to a client could be said to be that which would make him very happy and view the project as a success, or perhaps that which would make him dissatisfied with the project outcome. In this sense, if the client was a food retailer considering building a new store, some of his values might include *Cleanliness*; *Time*; *Appearance*; and *Shareholder Value*.

The traditional method within the construction industry for measuring the clients' satisfaction tends to focus on tangible, or measurable attributes that can be directly compared from client to client or project to project and would perhaps include *Time*; *Cost*; *Safety Record*; and *Number of Defects*.

Although value and performance are sometimes the same (e.g. time), EC Harris have been looking at ways in which client value drivers can be met through traditional means. This is what drove the requirement for this research, which above all was intended to identify a method of linking client drivers with building characteristics.

1.2 Data Mining and Decision Support

Data Mining (DM) is a process which utilizes a range of techniques and tools to extract patterns from data. Witten and Fank [9] describe data mining as "solving problems by analyzing data already present in databases". In this work, the *problem* was to analyse the value drivers in the construction of large and tall buildings, and the *database* was a set of buildings. The main objective of the DM task was to investigate the relationship between the attributers of large and tall buildings and their classification into three dimensions: *Size*, *Shape*, and *Quality* (known here as the *SSQ* dimensions).

Decision Support (DS) is a broad field concerned with supporting people in making decisions [1]. In our case, the problem was to evaluate and analyse buildings de-

scribed by various attributes, and on this basis to select buildings and their characteristics so as to best match the client's value drivers. Such DS problems are addressed within Decision Analysis [3], which provides a suitable methodology: hierarchical multi-attribute modeling. The idea is to develop a model that evaluates available choices (options) giving an estimate of their worthiness (utility) for the decision-maker. In the model, the whole decision problem is decomposed into smaller and less complex subproblems. These are represented by variables, which are organized into a hierarchy. In addition, some rules or procedures are defined that aggregate the evaluation of sub problems into the overall evaluation of options. The development of models is performed in an "expert modeling" way, which means that they are hand-made by an expert, possibly supported by a decision analyst and suitable software tools.

1.3 Outline of the remainder of the paper

Section 2 details the problem solving methodology – based on CRISP-DM – and its execution for this project. Much of the process is common between both data mining and decision support, but it does deviate in the modeling phase. Section 3 provides a discussion of the solutions provided, while suggestions for future work are considered in Section 4. Finally, Section 5 presents the conclusions.

2. Problem solving methodology and execution

Data Mining and Decision Support analysis processes broadly consist of a number of phases. Within the Sol-EU-Net consortium [8], the CRISP-DM methodology is used — CRoss Industry Standard Process for Data Mining [2]. In CRISP-DM, six interrelated phases are used to describe the data mining process: *business understanding*, *data understanding*, *data preparation*, *modeling*, *evaluation*, and *deployment*. Many of the phases are useful in both DM and DS problem solving methods.

2.1 Business Understanding

The data analytical problem to be solved must be set in the context of the business from which it is drawn. This is the aim of the *business understanding phase*. The owner of the problem in this work was the Capital Projects and Facilities Consultancy, *EC Harris*, who have offices in over twenty countries. The following quotations are taken from their web site (www.echarris.com):

> *"EC Harris is a leading International Capital Project and Facilities Consultancy with nearly 1,800 people directly managed and employed on construction and facilities consultancy worldwide. ... We serve clients whose needs span the whole life of a property asset, from setting winning strategies through delivery of the asset and its operation. Performance objectives which reflect clients' needs typically include ... capabilities as a Capital Project and Facilities Consultancy."*

The main objective, from EC Harris' point of view, of this project was to further explore work already undertaken in order to establish or confirm the following with respect to their understanding of the construction of large and tall buildings:

- That the work done to date was essentially sound and valid.
- To explore different ways of looking at the existing data and potentially discover new links between the various attributes of large and tall buildings.
- That it was reasonable and realistic to assume that client's value drivers could be linked to building attributes.
- To establish a method of linking client values with tangible attributes.
- That client satisfaction could probably be increased by establishing and respecting which building attributes most closely related to their values.
- To create a concept model that could be further developed as a discussion and decision support tool with clients. This would be used to establish clients value priorities, translated into building attributes, at a very early stage in the project.
- To explore the use of DM and DS as techniques for wider use in future projects.
- To explore whether subjective client views (values, which are largely tacit) could be extracted and linked to tangible attributes such as the height of a building.

An existing decision support tool focusing on large and tall buildings had already been developed at EC Harris. This tool is based on expert analysis of seventy buildings, as well as the combined expertise from within the EC Harris organization. The approach was to produce a set of *typical* or *benchmark* buildings for the space of the SSQ dimensions, which can be seen as an 18 cell cube (pictured in Figure 1).

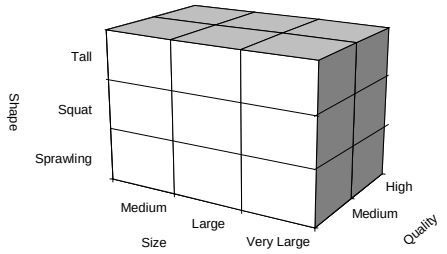


Figure 1 The Size-Shape-Quality Cube (SSQ)

2.2 Data Understanding

The purpose of the data understanding phase is to become familiar with all sources of the data. For this problem, the data was delivered in the form of a spreadsheet. This contained a set of building construction project attributes, 18 benchmark buildings, and 70 further real building projects. The following is a list of the categories of building construction project attributes, with the number of items in parentheses: *Location* (9), *Function* (6), *Description* (16), *Capacity* (*Building*) (36), *Capacity (Services)* (15), Cost (100), *Procurement* (2), *Programme* (5), *Access & Safety* (6), *Whole Life* (2), *Appearance* (4), *Project Team* (10). For each of the buildings (18 benchmark, and 70 real) values to the above attributes were provided.

Data relating to the value drivers were also provided in the spreadsheet (hereafter referred to as *attribute/value-driver matrix*, AVDM). For each of the building attributes, a series of value dimensions were available, with respect to three different client types: *Developer*, *Government*, and *Owner-occupier*. These value attributes were classed in to three categories: **Financial** (Profit, Shareholder Value, Market Value,

Growth), **Process** (Risk Control, Quality, Standards, Safety, Best Practice, Innovation, Supply Chain, Reuse of Resources, Whole Life), and **Market** (Image, Flexibility, Occupants Needs, Public Perception, Environmental Impact). AVDM is further explained in section 2.4.2, and a part of AVDM is shown in Table 2.

2.3 Data Preparation

The data preparation phase covers all the activities for constructing the final data sets for the modeling tools.

For DM, the following data preparation was performed. The main objectives of the DM were to (1) perform exploratory data analyses, and (2) to establish models between the building construction attributes and the SSQ dimensions – this was focused on the 18 benchmark building cases. Data preprocessing was aimed at reducing the number of attributes from the 211 attributes. With respect to the benchmark records, there were many attributes that did not discriminate on any of the SSQ dimensions. The attributes were sorted into three categories as follows: *Non-discriminating* means that the attribute values did not change, *Functionally dependent* were those attributes which were functionally dependent on other attributes (as they were formulae in the spreadsheet), and *Discriminating* attributes had values that varied. The discriminating attributes from the above (numbering 47) were all selected for the DM data set.

2.4 Modeling

The modeling techniques used by Data Mining and Decision Support differ sufficiently that they can be considered separately.

2.4.1 Data Mining Modeling

Three DM approaches were used in the modeling process: Clustering; Automatic attribute subset selection; and Decision tree induction.

Basic clustering. There was an implicit assumption from the domain expert that the benchmark buildings could be grouped into clusters. To test such a hypothesis, clustering was performed on the records of the 18 benchmark buildings. Weka's [9] KMeans implementation was used with three clusters selected. Studying the cluster numbers versus various attributes, lead to the tentative conclusion that the clusters seemed to separate based on Building Size.

Automatic attribute subset selection. Weka was used to determine further whether the attributes could be reduced. This was performed with respect to each of the SSQ Cube dimensions: Size, Shape, and Quality. (No results are reported here.)

Decision tree induction. The aim of this modeling was to produce – if possible – simple decision trees to map attributes into each of the SSQ dimensions. For this, the Weka implementation of C4.5 [7] – J48 – was deployed. Given the limited data, J48 was used with no test set (only using the 18 benchmark buildings as both the training and test sets), with validation only on the training set. The basic scheme was weka.classifiers.j48.J48 -C 0.25 -M 2, using 18 records and 63 attributes. The approach was to determine which attributes were selected by J48 as significant. For

further runs, those attributes that were significant in previous runs were excluded to highlight the next level of significant attributes. Sample results for each of the three target dimensions of *Size*, *Shape*, and *Quality* are summarized in Table 1. The trees induced were typically very concise. The results were discussed with the expert who was able to interpret each of the trees in light of his knowledge. A number of trees provided new insights into the relationships between the building attributes.

Table 1 Sample results of decision tree induction for the SSQ dimensions

Target	*Result*
Size	B6 "Flexibility Rating" <= 3 | B3 "Hours of Building use" <= 12: medium (6.0) | B3 "Hours of Building use" > 12: large (6.0) B6 "Flexibility Rating" > 3: very_large (6.0)
Shape	D10 "Clear Suspended Ceiling Depth" <= 350 | H3 "Proportion of Prefabrication (Off Site)" <= 2: Squat (6.0) | H3 "Proportion of Prefabrication (Off Site)" > 2: Tall (6.0) D10 "Clear Suspended Ceiling Depth" > 350: Sprawling (6.0)
Quality	L2 "Finishes" <= 3: medium (9.0) L2 "Finishes" > 3: high (9.0)

2.4.2 Decision Support Modeling

Three multi-attribute models were developed for the evaluation of buildings. All of them are hierarchical and represent a relationship between a building, described by a set of input variables, and its utility to the client. However, in order to explore various relations between attributes and value drivers, and to experiment with different knowledge representations, the models greatly differ in the:

- selection of input variables, which can be subsets of the attributes, value drivers, or both;
- level of detail in terms of the number of input variables and the depth of hierarchy;
- type of variables used in the model: continuous or discrete;
- relationship between attributes and value drivers: dependent or independent.

In the following, we refer to the models by the names of software packages that have been used for their development: Microsoft Excel [6], HIVIEW [4], and DEXi [5].

Excel Model

This is the most detailed DS model that uses 181 input attributes, which practically represent the whole input set. From the 211 available attributes, we only excluded 30 functionally dependent attributes that are obtainable from other attributes by a multiplication by a constant; these were redundant for this task.

A building project, described by the input attributes, is evaluated by a two-stage linear aggregation procedure, sketched in Figure 2. First, the values of attributes are mapped into 18 criteria, which represent value drivers. In the second stage, these are, according to their own hierarchical structure, aggregated into the overall evaluation. All the variables are continuous and represented on a [0,5] preference scale, where 0 and 5 correspond to the most and the least preferred evaluation, respectively.

The evaluation in both stages is carried out according to the expert-defined attribute/value-driver matrix (AVDM). Basically, this is a 181×18 matrix containing elements $e_{a,v} \in [0,5]$, specifying the expected influence of attribute a to the value driver v; the value 0 indicates no influence, and 5 very important influence. A small fragment of AVDM, which defines the attribute/value-driver relationship for governmental

clients, is presented in Table 2. For this type of model, the expert actually developed three such matrices, estimating the typical value drivers for three types of clients: Government, Development, and Owner Occupier.

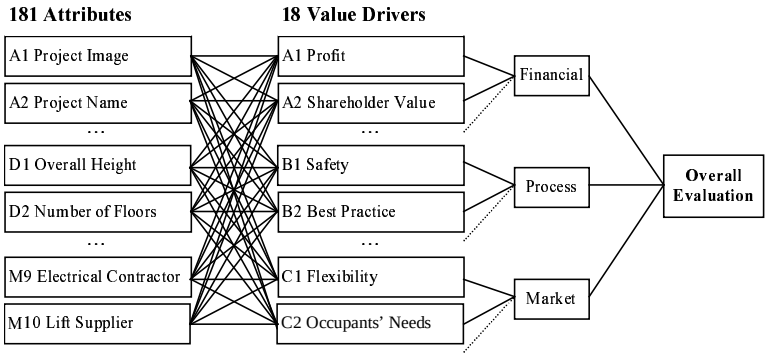


Figure 2 Two-stage evaluation schema of the Excel Model

Table 2 A part of attribute/value-driver matrix for governmental clients

Attributes	Value Drivers					
	Profit	Sharehld. Value	Safety	Best Practice	Flexibility	Occupants' Needs
A1 Project Image	0	1	0	0	1	1
A2 Project Name	0	1	0	0	1	1
D1 Overall Height	0	1	4	4	2	5
D2 Number of Floors	0	1	4	4	2	5

In the absence of dedicated DS software suitable for this kind of two-stage modeling, this model was implemented using Microsoft Excel. Figure 3 shows a few most essential parts of this fairly large spreadsheet – hiding several columns and most of the N=181 attributes. Input to the evaluation is specified in column J by of preferences $p_a \in [0,5]$, defined for each attribute a. There are four essential outputs of the model:

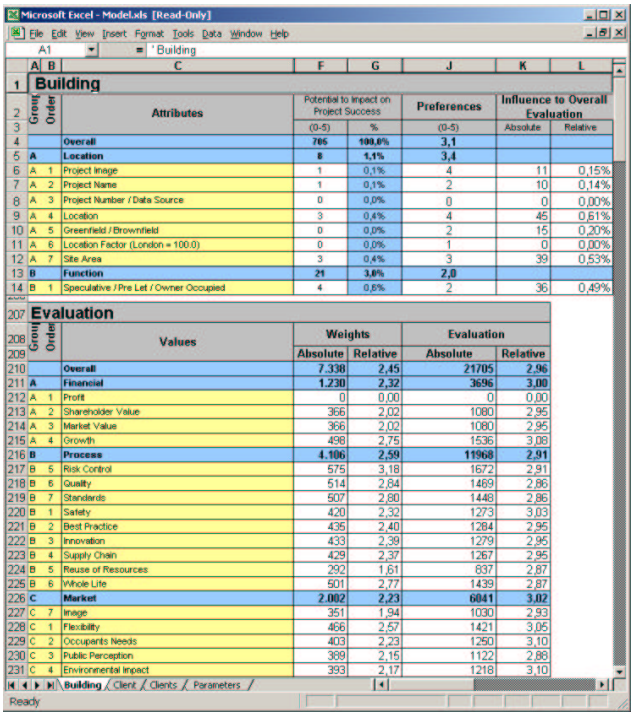
1. *Absolute and relative weights of value drivers* (see Figure 3, columns F and G, rows 210 and below). These are obtained from AVDM. For each value driver v, its absolute weight W_v is defined as

$$W_v = \sum_{i=1}^{N} e_{i,v}$$

   Relative weights are then proportionally normalised to the [0,5] scale.
2. *Absolute and relative evaluation of value drivers* (columns K and L, rows 210 and below). These are obtained from building's preferences p and taking into account elements e of AVDM. Again, relative evaluation is obtained by normalisation from the following absolute evaluation:

$$E_v = \sum_{i=1}^{N} e_{i,v} p_i$$

3. *Overall evaluation* (J210, K210): Final estimation of the building's utility to the client. It is obtained by a bottom-up aggregation of individual E_v's weighted by W_v's according to the hierarchical structure of value drivers.
4. *What-if analysis* (columns K and L, rows 5 to 14): For each attribute, it shows how would the change of that attribute by one unit influence the overall evaluation of the building. For example, the improvement of the attribute *A4 Location* from 4 to 5 would improve the overall evaluation by 45 absolute "points" or, in relative terms, by 0.61%.



Microsoft Excel - Model.xls [Read-Only]

File Edit View Insert Format Tools Data Window Help

A1 = 'Building

Building

Group	Order	Attributes	Potential to Impact on Project Success (0-5)	Potential to Impact on Project Success (%)	Preferences (0-5)	Influence to Overall Evaluation: Absolute	Influence to Overall Evaluation: Relative
		Overall	**765**	**100,0%**	**3,1**		
A		**Location**	**8**	**1,1%**	**3,4**		
A	1	Project Image	1	0,1%	4	11	0,15%
A	2	Project Name	1	0,1%	2	10	0,14%
A	3	Project Number / Data Source	0	0,0%	0	0	0,00%
A	4	Location	3	0,4%	4	45	0,61%
A	5	Greenfield / Brownfield	0	0,0%	2	15	0,20%
A	6	Location Factor (London = 100.0)	0	0,0%	1	0	0,00%
A	7	Site Area	3	0,4%	3	39	0,53%
B		**Function**	**21**	**3,0%**	**2,0**		
B	1	Speculative / Pre Let / Owner Occupied	4	0,5%	2	36	0,49%

Evaluation

Group	Order	Values	Weights: Absolute	Weights: Relative	Evaluation: Absolute	Evaluation: Relative
		Overall	**7.338**	**2,45**	**21705**	**2,96**
A		**Financial**	**1.230**	**2,32**	**3696**	**3,00**
A	1	Profit	0	0,00	0	0,00
A	2	Shareholder Value	366	2,02	1080	2,95
A	3	Market Value	366	2,02	1080	2,95
A	4	Growth	498	2,75	1536	3,08
B		**Process**	**4.106**	**2,59**	**11968**	**2,91**
B	5	Risk Control	575	3,18	1672	2,91
B	6	Quality	514	2,84	1469	2,86
B	7	Standards	507	2,80	1448	2,86
B	1	Safety	420	2,32	1273	3,03
B	2	Best Practice	436	2,40	1284	2,95
B	3	Innovation	433	2,39	1279	2,95
B	4	Supply Chain	429	2,37	1267	2,95
B	5	Reuse of Resources	292	1,61	837	2,87
B	6	Whole Life	501	2,77	1439	2,87
C		**Market**	**2.002**	**2,23**	**6041**	**3,02**
C	7	Image	351	1,94	1030	2,93
C	1	Flexibility	466	2,57	1421	3,05
C	2	Occupants Needs	403	2,23	1260	3,10
C	3	Public Perception	389	2,15	1122	2,88
C	4	Environmental Impact	393	2,17	1218	3,10

Building / Client / Clients / Parameters /

Ready

Figure 3: **An excerpt from the Excel Model**

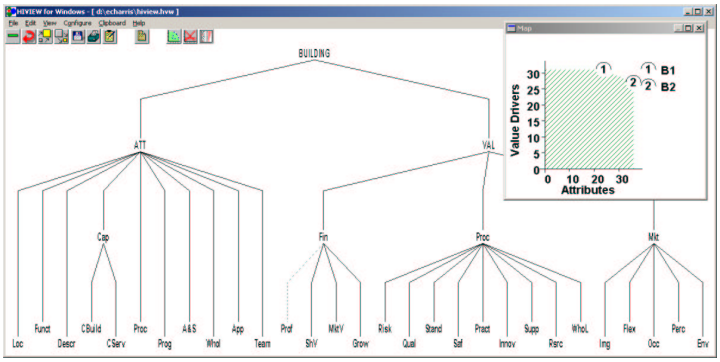


Figure 4: HIVIEW Model: Hierarchical structure of variables and a map

HIVIEW Model

In the second DS model, the aim was to explore a different way of combining attributes and value drivers. Instead of the Excel Model's two-way sequential procedure, they were evaluated in parallel, by combining two hierarchies of preference variables: one consisting of attributes and the other consisting of value drivers. The hierarchy of attributes was defined in less detail than previously, using only 11 top-level attributes (Figure 4). The essential characteristic of this model is that it facilitates a direct analysis of the relationship between attributes and value drivers using maps. Also, HIVIEW provides tools for sensitivity analysis, which are particularly suitable for this problem.

DEXi Model

The third DS model produced is the least detailed one. It involves only value drivers, which are represented by a four-level hierarchy (Figure 5). All the variables in the hierarchy are qualitative, i.e., they assess and evaluate value drivers using qualitative descriptive values, such as: *unacceptable*, *acceptable*, *good*, *excellent*. The aggregation is carried out by expert-defined *if-then* decision rules.

This model is best suited for a quick, qualitative analysis of value drivers, especially when comparing different buildings and their variants. The model supports what-if analysis and can represent evaluation results using radar charts (Figure 5). Furthermore, since DEXi's evaluation mechanism can deal with missing data, this is the only model of the three that can produce results – albeit less precise – even when some client's value drivers are unknown.

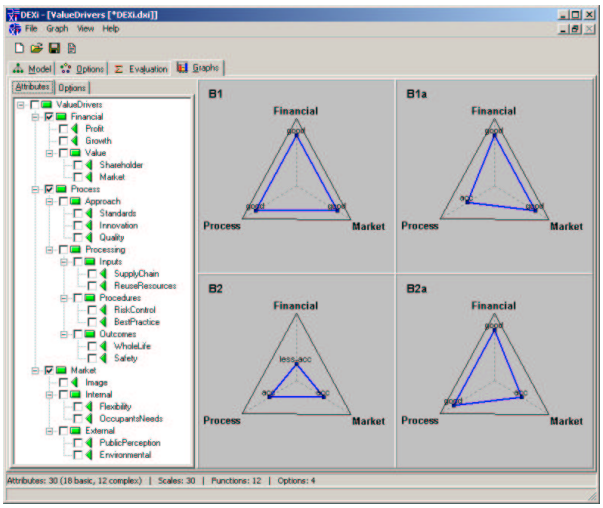


Figure 5: DEXi Model: Hierarchy of value drivers and radar charts

2.5 Evaluation

To date, the evaluation of the models was based solely on feedback from the building construction expert. After the models were presented and discussed, the expert gave the following comments:

"The data mining and decision support techniques used on this project have been extremely valuable to me personally as well as the EC Harris organization and despite the relatively limited pool of data and time available have still added value to the work that we were already doing. The initial expectations have been fulfilled, with the key benefits being;

- *Existing work by EC Harris was validated.*
- *Some new links between attributes were discovered.*
- *Client values linked to tangible attributes.*
- *Concept value vs. attribute model created.*
- *Established potential for future data mining and decision support."*

2.6 Deployment

To date none of the models have been deployed in the EC Harris organization. There are plans to perform further modeling, and potentially deploy some of the models.

3. Discussion

The inputs to this project were data about the construction of large and tall buildings, as well as expert opinion on the values of aspects of these buildings for their owners. The data contained 211 attributes, and of two types of records: (1) 18 expert-assembled benchmark buildings, which were created from experience within the EC Harris organization, and (2) 70 partially complete records of real building construction projects or their designs. It was remarkable that the expert was able to complete the assignment of values to over ten thousand items in less than 24 hours. Furthermore, the analyses of the value data showed that the assignments were reasonably consistent.

The broad goals of the project were to: (1) explore in detail relevant data for further knowledge development; (2) identify and articulate the key issues around value drivers in construction; and (3) build models for use in a decision support system. These objectives were considered to have been achieved by the building expert.

In what follows we highlight some of the achievements of the project.

Attribute mapping strategies. Two main mapping strategies were developed. The first was the mapping building attributes to the evaluation of the building with respect to the client's values. The second was the mapping of attributes to the type of building.

Decision Models with two stages of evaluation. Decision Models were produced that utilized a two state evaluation procedure in which building construction project attributes are mapped to an overall evaluation. This is performed by a two-stage linear aggregation procedure (refer back to Figure 2). First, the values of attributes are mapped into 18 criteria, which represent value drivers. In the second stage, these, according to their own hierarchical structure, are aggregated into the overall evaluation. All the variables in the model are continuous.

Combining Data Mining and Decision Support models. The two mapping strategies, outlined above, were performed using two different analysis approaches. The development of the value drivers models was performed using DS, while the models for mapping attributes to building types were produced by DM. Both of the approaches use (in part) similar input attributes. This gives rise to a novel opportunity to combine the two approaches (Figure 6).

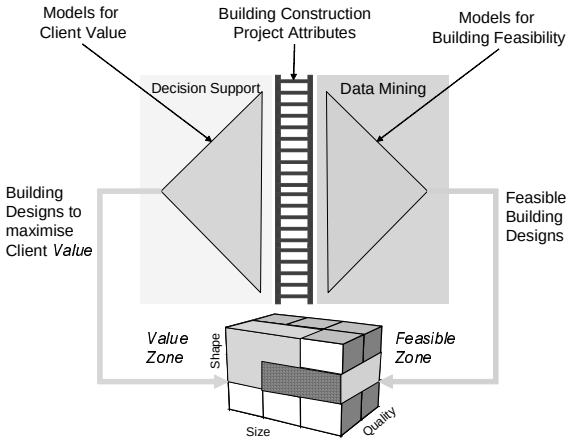


Figure 6 Combining Decision Support and Data Mining Models

The combination of the DM and DS models suggests a development of decision support system to assist clients in choosing their most suitable building type. A client – with the help of a building expert – can articulate his values on the attributes that have been selected. These values can be maximized using the decision support models. Having chosen the attributes, the data mining models can then be used to suggest whether such a building project is broadly feasible. This process can be repeated further until the client is satisfied with a building type.

A generic framework for analysing value drivers for business. The procedure utilized in producing decision models for value drivers could be deployed in other business environments.

4. Future Work

The execution of this project has – as is common – provided many insights, but has also lead to new questions and directions for future work. Some of the directions are specific to the problems tackled in the project, while others have broader applicability. Some directions are listed below.

- Simplification of Models
- Analysis of design constraints and building project feasibility
- Integrating DEXi models with the Excel models
- Establish mappings from Attributes to the preferences of attributes

- Further develop and extend a framework for the generic analysis of value drivers in business.

Due to space constraints, none of these directions are discussed in any detail.

5. Conclusions

The objective of this work was to employ the tools and techniques of two related disciplines – Data Mining and Decision Support – to the solution of the practical problem of knowledge development and articulation in the domain of the building construction of large and tall buildings.

Data Mining techniques produced models that validated existing expert analyses, as well as providing some new insights. Significant models, however, were produced by Decision Support techniques, to be used in optimizing the perceived value of possible building projects by clients. The original modeling objectives were achieved with the possibility of deploying the models within the building expert's organization.

In working on the specific problem, a concept of how data mining and decision support models may be integrated was proposed. In such a setting, the different models utilize similar input vectors, but produce differing outputs. This may give rise to the development of decision support systems that allow the cycling between the states provided by the models until a suitable candidate solution is reached.

Acknowledgements

The work reported in this paper was supported by the IST-1999-11495 project Sol-EU-Net, Data Mining and Decision Support for Business Competitiveness: A European Virtual Enterprise. The authors would like to thank the Knowledge Management Forum at Henley Management College UK for bringing them together.

References

1. Bohanec, M.: What is decision support? In: Škrjanc, M., Mladenić, D. (eds.): Information Society IS-2001: Data Mining and Decision Support in Action! Ljubljana, 86–89 (2001).
2. Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., and Wirth, R.: CRISP-DM 1.0: Step-by-step data mining guide. CRISP-DM consortium (2000).
3. Clemen, R.T.: Making Hard Decisions: An Introduction to Decision Analysis. Duxbury Press (1996).
4. Enterprise LSE: HIVIEW for Windows. http://www.enterprise-lse.co.uk/hiview.htm (1999).
5. Gams, M., Bohanec, M.: Intelligent systems applications. Informatica 25(3), 387–392 (2001).
6. MS Excel: Microsoft Office – Excel Home Page. http://www.microsoft.com/office/ excel/default.asp (2002).
7. Quinlan, J. R.: C4.5: Programs for Machine Learning. Morgan Kaufmann, (1993).
8. SolEuNet: Project Sol-EU-Net: Data Mining and Decision Support for Business Competitiveness: A European Virtual Enterprise, http://soleunet.ijs.si/website/html/euproject.html (2002).
9. Witten, I.H., and Frank, E.: Data mining: practical machine learning tools and techniques with Java implementations. Morgan Kaufman Publishers, San Francisco (1999).

Committee-Based Selective Sampling with Parameters Set by Meta-Learning

Miloslav Nepil and Luboš Popelínský

KD Group at Faculty of Informatics
Masaryk University in Brno, Czech Republic
{nepil,popel}@fi.muni.cz

Abstract. This contribution presents a parametric variant of committee--based selective sampling. The committee members learned on small subsets obtained by random sampling from the original dataset are used to classify the rest of the dataset. Those examples on which the committee came to consensus are considered to be easy, the others to be hard. The main idea is to select the resulting training subset with a different ratio of easy to hard examples. In the second part of the paper meta-learning technique for parameter setting is introduced and experimental results obtained with it are discussed. This selective sampling method has been proven useful in reducing the learning time while keeping the accuracy at a better level than random selection does. The meta-learning method for parameter settings displays fairly low ranking error and is sufficient for a reliable and immediate prediction of parameters.

1 Introduction

Instance selection methods are aimed at finding a representative subset of training data which would be smaller than the original dataset, but still would provide enough information to achieve an accurate model. Three main motivations can be found for reducing the number of training examples. The first reason could be that a sufficient amount of labeled examples is difficult to obtain; a problem frequently faced in natural language processing. If we cannot rely on unsupervised learning and examples should be annotated manually by a human, then we need to save the annotation costs [1, 2].

The second, quite an opposite situation arises when we have a cheap access to a large, or even potentially unlimited amount of training data. It happens when our data mining task can be solved with an unsupervised or implicitly supervised approach [3]. But still, we need to select a finite training subset of reasonable size since we are always limited in learning time.

And the third convenience for a selection of training examples comes up when the learning on a whole dataset leads to a huge model. This can be caused either by presence of noisy examples or outliers in the training data [4], or by an inherent property of the given learning algorithm. Instance-based learning algorithms are clearly the case, but also for several other algorithms (including tree construction ones such as C4.5, and rule construction ones such as C4.5rules)

it has been observed [5] that increasing the amount of data used to build a model often results in a linear increase in model size, although that additional complexity results in no significant increase in model accuracy. Therefore, a selection of training data could help us to make the model more compact and concise.

This work was motivated by the fact that for large datasets which are being treated in data mining the experiments took too much time. Therefore, our primary goal was to decrease the learning time (and perhaps the model size) while keeping the error rate as low as possible; a goal perfectly addressed by selective sampling. Selective sampling proceeds, in general, by measuring the information content of each training example. The objective is to select those examples which could provide the most informative description of a target concept being learned. The measure of information content can be either uncertainty-based [6] or committee-based [7]. Approaches based on uncertainty often derive an explicit measure of the expected information gained by using the example. However, the main drawback of these approaches is that they are usually dependent on a particular learning algorithm. Since we have been concerned with a simple selective sampling technique which could be easily applied to many different learning algorithms, we gave precedence to the committee-based approach.

The structure of this paper is as follows. In Section 2 we first explain the main idea of our variant of selective sampling and then describe the algorithm. Discussion on speed up of this way of sampling follows. Section 3 brings experimental verification of usefulness of this selective sampling method. The second part of this paper concerns settings of parameters of the method. In Section 4 we describe the meta–learning method used. Section 5 displays the results obtained by meta-learning.

2 Committee-Based Selective Sampling

2.1 General scheme

The general scheme of our variant of selective sampling driven by committee of classifiers is as follows. In the beginning, we learn a set of several fast classifiers – members of the committee. Then, we let the committee make a decision about each given training example, which means that each member has to classify the example according to its own knowledge about the target concept. Thus, we get several (possibly different) class predictions for each example. Hence, the information content of the example is evaluated as a measure of disagreement among the committee members. For final training we select a subset of examples with highest information content.

If we want to devise a particular variant of committee-based selective sampling, several questions should be answered:

1. How many committee members do we need?
2. How to choose the committee members?
3. How to measure the disagreement among committee members?

4. How to select the resulting subset of training examples?

Our parametric variant of committee-based selective sampling adopts the following solution. It presumes that we have a fast (low complexity) learning algorithm $\mathcal{A}_{init}$ which we use for training initial classifiers (committee members) and a slow (but robust) learning algorithm $\mathcal{A}_{final}$ which we use for training the final classifier. Both training and prediction times of the initial classifiers are important, due to the fact that predictions on the whole dataset have to be obtained.[1] Our method treats the number of committee members as a fixed parameter N. The committee members are established by learning on small subsets obtained by random sampling from the original dataset. The size of these small subsets is given by another parameter I. Our measure of disagreement is rather rough, since we distinguish only two categories: a complete consensus and a dissension. Those examples on which the committee came to a consensus are considered to be *easy*, while the others are considered to be *hard*. The main idea of our method is to select the resulting training subset in such a way that the ratio of *easy* to *hard* examples in the resulting subset is computed as a function of the corresponding ratio which was observed in the initial dataset. As this function we simply took a multiplication by a coefficient X. The values $0 \leq X < 1$ mean that we want to decrease the ratio of easy examples in the final subset (actually, $X = 0$ implies no easy examples there). On the other hand, the values $X > 1$ mean that we intend to add even more easy examples to the final subset. Note that the value $X = 1$ results in no change of the easy/hard ratio, therefore this setting corresponds to random sampling. Another parameter F determines the size of the final training subset.

2.2 Algorithm

We can already see that our selective sampling technique is parameterised by four numerical values:

N – a number of initial classifiers (members of the committee)
I – a size of the initial training subset used for learning initial classifiers
F – a size of the final training subset used for learning a final classifier
X – a coefficient for modifying the original ratio of *easy* to *hard* examples

More formally, our example selection works as follows:

1. The number of committee members is given by a parameter N, $N \geq 2$.
2. From the given training set we draw randomly an initial subset of the relative size I, $0 < I < 1$, as a fraction of the original dataset. The initial subset is randomly split into N blocks and each block is used for training one initial classifier with a learning algorithm $\mathcal{A}_{init}$.

[1] However, this need not be the case. In Section 2.3 we describe an improvement of our basic method which estimates the size of a subset sufficient for submitting to the committee.

3. Each initial classifier is applied to the whole training set. Therefore, we obtain N class predictions for each example. Those examples which were classified consistently (it means that all N predictions were identical) are considered as *easy* ones while the others are considered as *hard* ones. Let's denote the ratio of *easy* to *hard* examples as e/h.
4. We select randomly a final training subset so that its ratio of *easy* to *hard* examples is given by the expression $X \cdot e/h$ where the coefficient X, $X \geq 0$, is another fixed parameter. The final subset's size is determined by a parameter F, $0 < F < 1$, as a fraction of the original dataset. The final subset is used for training a final classifier with a learning algorithm $\mathcal{A}_{final}$.

It is not difficult to guess that a particular setting of the parameters presented above has an important impact on performance of the method. The appropriate parameter setting is not a trivial task since it depends not only on properties of the dataset at hand, but also on our preferences with regard to the learning time, the precision of learned model, and the model size.

2.3 Speeding up the Sampling

In the previous description of our basic sampling algorithm we have stated – for the sake of simplicity – that the committee should give class predictions on the whole dataset. On the contrary to this, for the final subset we want to select only a certain (possibly small) amount of original training examples. Of course, it makes many computed predictions redundant and, as a consequence, it means that we would waste computational time for the sampling. Although the initial classifiers are assumed to be fast, they need some time to predict the target class. Considering that we want to sample from large datasets and the number of committee members can be higher as well, we have concerned us with the question if there is a possibility to estimate the size of a subset of original dataset which would be sufficient for subsequent processing.

Let s denotes the size of an original dataset. Then we know that the final subset must contain $F \cdot s$ examples.[2] It implies that the committee should give class predictions on $F \cdot s$ examples, at least. So, we run the committee on these $F \cdot s$ examples and find out that e_1 examples out of them are easy and h_1 are hard, $e_1 + h_1 = F \cdot s$. From the fourth step of the basic algorithm in Section 2.1 we already know that $e_2/h_2 = X \cdot e_1/h_1$ where e_2 and h_2 denote the required numbers of easy and hard examples in the final subset, respectively. But, at the same time, $e_2 + h_2 = F \cdot s$. It follows that

$$e_2 = \frac{X \cdot e_1 \cdot (e_1 + h_1)}{X \cdot e_1 + h_1}, \qquad h_2 = \frac{h_1 \cdot (e_1 + h_1)}{X \cdot e_1 + h_1}.$$

Now if $h_2 > h_1$ (which means $X < 1$) then we need to find some additional hard examples, otherwise, if $e_2 > e_1$ (which means $X > 1$) then we need to find some additional easy examples. Therefore, in the former case, it suffices to let the

[2] Rounding to whole numbers is omitted in this section.

committee judge $(h_2 - h_1) \cdot (e_1/h_1 + 1)$ additional examples to obtain $h_2 - h_1$ new hard examples, and, in the latter case, it suffices to judge $(e_2 - e_1) \cdot (h_1/e_1 + 1)$ additional examples to obtain $e_2 - e_1$ new easy examples. Of course, an important point here is that we assume the distribution of easy and hard examples to be the same on the whole original dataset.

This improvement of our basic method makes the sampling algorithm two--fold: at first, the committee is applied to $F \cdot s$ examples, and then it is run on an additional block of data, whose size is determined by the result of the first run. As an asset, the committee does not need to explore the whole given dataset, which significantly saves sampling time in many cases.

3 Experimental Results of Selective Sampling

Table 1. The comparison of results achieved on whole dataset (WD), by selective sampling (SS), and by random sampling (RS). The initial algorithm $\mathcal{A}_{init}$ was `c50tree` and the final algorithm $\mathcal{A}_{final}$ was `c50boost`. The parameters of selective sampling were set as follows: $N = 2$, $I = 0.2$, $F = 0.3$, and $X = 0.1$. The random sampling was set to select the same resulting fraction of data (30 %).

Dataset	Total Time (sec)			Model Size			Error Rate (%)		
	WD	SS	RS	WD	SS	RS	WD	SS	RS
adult	139.3	22.3	22.1	23768	5977	8751	14.49	14.40	15.33
letter	81.1	21.3	15.3	11691	6966	5484	4.69	7.85	9.73
optical	53.0	11.9	7.8	1771	1009	788	2.47	3.83	4.48
pendigits	32.0	8.8	6.2	1703	1122	904	1.15	1.49	2.27
quisclas	19.2	8.3	6.9	5447	1829	1713	35.24	36.77	36.04
satimage	51.9	12.5	8.5	2652	1346	959	9.54	9.88	11.31

Table 2. The similar experiment as above, but for the final algorithm `c50rules`. The parameters of selective sampling were set here as follows: $N = 2$, $I = 0.1$, $F = 0.3$, and $X = 0.2$. The random sampling was set again to select the same resulting fraction of data (30 %).

Dataset	Total Time (sec)			Model Size			Error Rate (%)		
	WD	SS	RS	WD	SS	RS	WD	SS	RS
adult	89.3	16.2	12.8	327	218	215	13.67	14.34	14.80
letter	231.5	29.1	20.3	1177	721	548	11.03	18.37	19.56
optical	16.0	3.3	1.6	209	108	95	8.64	11.17	13.49
pendigits	18.1	3.6	2.1	188	121	106	3.22	4.80	5.99
quisclas	16.9	3.5	2.8	475	179	180	35.34	37.48	37.16
satimage	24.0	3.9	1.9	281	138	108	13.36	14.65	15.77

At first, we shall show that the selective sampling method really selects representative subsets of the training data. A better quality of the dataset obtained by selective sampling displays a better accuracy of the learned model when compared to the random sampling. In our experiments we tried a family of C5.0 [8]

algorithms.[3] While c50tree has been used as an initial learner, we have used c50boost and c50rules as final learners: corresponding results on several datasets explored inside the MetaL project[4] are shown in Tables 1 and 2, respectively. These tables show results concerning the total time, the size of learned model, and its error rate on a test set. All numbers were computed through 10-fold cross-validation. We can see that the error rate achieved by selective sampling remains in many cases close to the original error rate. For adult dataset it even decreased, when c50boost has been used as a final learner. Furthermore, selective sampling is always better than random sampling in terms of accuracy, except for quisclas dataset (which has an excessive error rate on the whole dataset, either). However, it should be noted that this singularity of quisclas dataset does not mean that the selective sampling is not useful for it at all. If we use c50boost as the final learner and choose a different setting, namely $N = 4$, $I = 0.3$, $F = 0.3$, and $X = 0.3$, then we get the following results: Total Time 9.1, Model Size 1831, and Error Rate 35.58. Thus, the accuracy of selective sampling is better than of random sampling for quisclas as well, but we must hit the appropriate parameter setting.

As for the reduction of model size, often the selective sampling is almost as successful as the random sampling. For adult dataset with c50boost as a final learner and quisclas dataset with c50rules as a final learner, the selective sampling produced even smaller model than random sampling did.

Nevertheless, the most significant is the reduction of total time. The total time comprises the time taken by sampling, training and testing together. It means that in the case of selective sampling the total time subsumes also the time spent on learning and application of initial classifiers. Consequently, the random sampling is a bit faster, but the extra time spent on selective sampling seems to be really useful, considering the better preserved accuracy. The main asset of time reduction does not rest in the fact that we are able to shrink the total time from 51.9 to 12.5 seconds, but the important thing is that 5:1 --time reduction with no considerable decrease in accuracy can help the learning algorithm to scale up to significantly larger datasets.

The results and discussion presented above concern the settings $X < 1$, when hard examples are being added to the final subset. We have also tried the settings $X > 1$, which means to add easy examples. These settings resulted in a greater reduction of the total time as well as the model size, however, the accuracy was worse than that of random sampling.

4 Meta-Learning for Parameter Setting

4.1 Ranking Function

As we could see earlier, particularly in the discussion about "abnormal" quisclas dataset, the performance of our selective sampling method strongly depends on

[3] http://www.rulequest.com
[4] http://www.metal-kdd.org

the setting of its parameters. Table 3 demonstrates the impact of parameter X (the coefficient for modifying the original ratio of *easy* to *hard* examples) on the performance criteria. It is not surprising that the demand on a fast processing and small model goes against the demand on a high accuracy.

Table 3. The impact of parameter X on the resulting time, model size and error rate, shown on satimage dataset with c50tree as an initial learner and c50boost as a final learner. The resting parameters are fixed to these values: $N = 2$, $I = 0.2$, and $F = 0.3$. The expression e_1/h_1 denotes the original (observed) ratio of *easy* to *hard* examples whereas the expression e_2/h_2 refers to the resulting (computed) ratio. The following time values are listed: T_1 – sampling time, T_2 – training time, T_3 – testing time, and T – total time. Selective sampling with the setting $X = 1.0$ corresponds to random sampling, therefore the sampling time is considered to be zero.

X	e_1/h_1	e_2/h_2	T_1	T_2	T_3	T	Size	Error
0.1	1389/348	495/1186	1.9	10.5	0.1	12.5	1346	9.88
0.2	1389/348	771/965	1.7	10.0	0.1	11.8	1259	10.46
0.3	1389/348	946/790	1.5	9.4	0.1	11.1	1176	10.57
0.4	1389/348	1067/668	1.4	9.0	0.1	10.5	1124	10.41
0.5	1389/348	1157/577	1.4	8.9	0.1	10.4	1094	10.69
0.6	1389/348	1225/505	1.3	8.3	0.1	9.7	1056	10.86
0.7	1389/348	1279/456	1.2	8.4	0.1	9.8	1025	11.31
0.8	1389/348	1322/410	1.2	8.1	0.1	9.4	1010	11.20
0.9	1389/348	1358/370	1.2	8.1	0.1	9.4	971	11.73
1.0	1389/348	1389/348	0.0	8.4	0.1	8.5	959	11.31

Therefore, it is clear that search for the best parameters needs to take into consideration not only the properties (the meta-characterisations) of the particular dataset, but also our preferences with regard to some performance criteria (time, accuracy, model size). This observation naturally leads to ranking techniques. Considering our experimental purposes we have resorted to very simple ranking function which takes into account just time and error rate and not model size:

$$R(K, T_s/T_w, E_s/E_w) = K \cdot (T_s/T_w) + (1 - K) \cdot (E_s/E_w)$$

Here T_s and T_w are the total times achieved by learning from a sample and by learning from a whole dataset, respectively. Similarly, E_s and E_w are the error rates achieved by learning from a sample and by learning from a whole dataset, respectively. And finally, K, $0 \leq K \leq 1$ is a balance parameter: $K = 0$ means that we are interested only in accuracy and, on the contrary, $K = 1$ means that we regard the total time only. We always want to minimise this ranking function's value.

4.2 Generation of Meta-Learning Data

As the meta-data characteristics of a dataset we have exploited a learning time T_p, a model size S_p, and an error rate E_p of a pilot classifier. The pilot classifier

was a classifier attained by learning with the initial algorithm $\mathcal{A}_{init}$ on a random sample of a small, fixed size (10 %). The choice of these meta–attributes was motivated by the fact that these characteristics are cheaper than most of statistical and information-theory measures to obtain, and also by our belief that for our purpose they will serve comparably well.

Then, we have run the selective sampling on six datasets with various settings to find out corresponding values of total time T_s and error rate E_s. As for the tested settings, the values of their particular parameters were drawn from these enumerations: $N \in \{2, 3, 4\}$, $I \in \{0.1, 0.2, 0.3\}$, $F = 0.3$, and $X \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$. All combinations of these values were tested. From all the obtained results we have compiled examples for learning a meta-model. Figure 4 shows an example from a training meta-dataset. Each training example for meta-learning consisted of attributes which could be divided into four groups: 1. data characteristics (pilot classifier results): T_p, S_p, E_p, 2. selective sampling parameters: N, I, F, X, 3. balance parameter K, and 4. the corresponding value of ranking function $R(K, T_s/T_w, E_s/E_w)$. The groups 1,2,3 represent independent (predictive) attributes, while the last attribute (group 4) is dependent (predicted).

Table 4. An example from a meta-dataset compiled from adult dataset. Using c50tree and c50boost as an initial and a final learner, respectively, for the setting $N = 2$, $I = 0.2$, $F = 0.3$, and $X = 0.1$ we got the total time $T_s = 22.3$ and error rate $E_s = 14.40\,\%$. Corresponding results on the whole dataset (without sampling) are $T_w = 139.3$ and $E_w = 14.49\,\%$. Therefore, the resulting value of ranking function is $R(0.1, 22.3/139.3, 14.40/14.49) = 0.911$.

group 1			group 2				group 3	group 4
T_p	S_p	E_p	N	I	F	X	K	R
0.56	72	0.152	2	0.2	0.3	0.1	0.1	0.911

4.3 Learning the Meta-Model

We do not aim at predicting the best parameter setting directly. Instead, a meta--model is designed to predict the value of ranking function. When processing an unseen data, the data characteristics (attributes from the group 1) and the balance parameter (group 3) are known and we need to find such selective sampling parameters (attributes from the group 2) which would minimise the ranking function value (output of the meta-model, group 4).

We decided to use regression trees as a meta-model for our purpose since in a regression tree it is easy to find values of unknown attributes which would minimise the output function. For learning the meta-model we utilised the system RT4.1 [9] which generates regression trees.[5] Figure 1 shows a part of the

[5] http://www.liacc.up.pt/~ltorgo/RT

regression tree which we have obtained. If the condition in a node holds than the left branch is chosen, otherwise the right one.

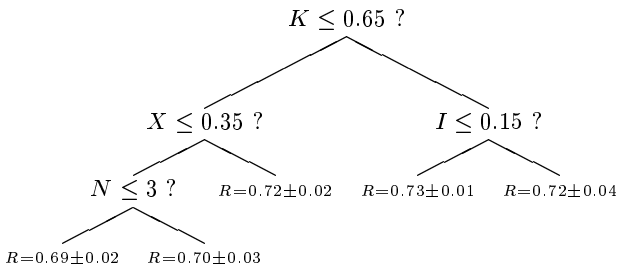


Fig. 1. Example of a regression tree used as a meta-model for parameter setting.

5 Experimental Results of Meta-Learning

Table 5 presents the results of our regression meta-model. We suppose that the most important performance measure is a Relative Increase in Ranking Value (RIRV). It is a comparison of ranking values of the predicted parameter setting and the best known parameter setting for a particular dataset and a learning algorithm. Hit Rate (HI) is a percentage of those cases when the best known parameter setting was also predicted. Relative Increase in Error Rate (RIER) is a comparison of error rates of a resulting (non-meta) classifier obtained from the predicted parameter setting and a resulting classifier obtained from the best known parameter setting. Finally, Relative Increase in Total Time (RITT) is a comparison of total time of a resulting (non-meta) classifier obtained for the predicted parameter setting and a resulting classifier obtained for the best parameter setting.

As we can see in Table 5 RITT is negative and RIER is positive. It means that our regression model tends to predict settings resulting in faster processing, but worse error rate. All numbers were computed from leave-one-out validation on six different datasets: adult, letter, optical, pendigits, quisclas, satimage. As we can see, the meta-attributes T_p, S_p, E_p made the parameter prediction surprisingly worse. We obtained more accurate meta-model (especially for c50boost) by not using those meta-attributes. This is probably due to the fact that the ranking function has a very similar curve for different datasets and thus the meta-characteristics do not bring additional information – they mislead the regression model instead. On the other hand, it should be noted that we have learned our regression model from relatively small amount of datasets. In fact, the training set in each fold of the leave-one-out validation consisted of five datasets. However, the presence of meta-attributes in the case of c50rules final algorithm led to a significant

increase in hit rate, whereas the ranking deviation stayed at almost the same level. Therefore, we suppose exploitation of the meta-attributes to be promising.

Table 5. The results of meta-learning for selective sampling with c50tree as an initial algorithm and c50boost and c50rules as final algorithms.

Algorithm	Meta Att.	RIRV	HI	RITT	RIER
c50boost	absent	7.1257 %	27.3 %	-0.5651 %	9.7485 %
c50boost	present	9.9497 %	19.7 %	-6.9764 %	13.5287 %
c50rules	absent	5.9469 %	21.2 %	-3.1301 %	7.3770 %
c50rules	present	5.9957 %	30.3 %	-10.6430 %	8.0315 %

6 Conclusion

We have presented a new parametric variant of committee-based selective sampling and a meta-learning technique for setting its parameters. The selective sampling has been proven useful in reducing the learning time while keeping the accuracy at a better level than random selection does. The main contribution of the meta-learning is that its ranking error is fairly low (7.13% for c50boost and 5.95% for c50rules) which is sufficient for a reliable and immediate prediction of the right parameters setting for selective sampling.

Acknowledgement

Our thanks go to Pavel Brazdil and other members of LIACC group for their help and also to the anonymous referee for helpful comments. This research has been partially supported by Esprit LTR Project MetaL and by the Czech Ministry of Education under the grant JD MSM 14330003.

References

1. Dagan, I., Engelson, S.P.: Selective sampling in natural language learning. In: Proceedings of the Workshop on New Approaches to Learning for Natural Language Processing at IJCAI 1995, Montreal, Canada, Morgan Kaufmann (1995)
2. Thompson, C.A., Califf, M.E., Mooney, R.J.: Active learning for natural language parsing and information extraction. In: Proceedings of 16^{th} International Conference on Machine Learning, ICML 1999, Bled, Slovenia, Morgan Kaufmann (1999) 406–414
3. Hirota, K., Pedrycz, W.: Implicitly-supervised learning and its application to fuzzy pattern classifiers. Information Sciences **106** (1998) 71–85
4. Gamberger, D., Lavrač, N.: Filtering noisy instances and outliers. In Liu, H., Motoda, H., eds.: Instance Selection and Construction for Data Mining. Kluwer Academic Publishers, Boston/Dordrecht/London (2001) 375–394

5. Oates, T., Jensen, D.: Large datasets lead to overly complex models: An explanation and a solution. In: Proceedings of The Fourth International Conference on Knowledge Discovery and Data Mining. (1998) 294–298
6. Lewis, D.D., Catlett, J.: Heterogeneous uncertainty sampling for supervised learning. In Cohen, W.W., Hirsh, H., eds.: Proceedings of 11^{th} International Conference on Machine Learning, Morgan Kaufmann (1994) 148–156
7. Freund, Y., Seung, H.S., Shamir, E., Tishby, N.: Selective sampling using the query by committee algorithm. Machine Learning **28** (1997) 133–168
8. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann (1993)
9. Torgo, L.: Inductive Learning of Tree-based Regression Models. PhD thesis, Department of Computer Science, Faculty of Sciences, University of Porto (1999)

Decision Tree-Based Data Characterization for Meta-Learning

Yonghong Peng[1] Peter A Flach[1] Pavel Brazdil[2] Carlos Soares[2]

[1] Department of Computer Science, University of Bristol, UK
{yh.peng,peter.flach}@bristol.ac.uk
[2] LIACC/Fac. of Economics, University of Porto, Portugal
{pbrazdil,csoares}@liacc.up.pt

Abstract. Appropriate selection of learning algorithms is essential for the success of data mining. Meta-learning is one approach to achieve this objective by identifying a mapping from data characteristics to algorithm performance. Appropriate data characterization is, thus, of vital importance for the meta-learning. To this effect, a variety of data characterization techniques, based on three strategies including simple measure, statistical measure and information theory based measure, have been developed, however, the quality of them is still needed to be improved. This paper presents new measures to characterise datasets for meta-learning based on the idea to capture the characteristics from the structural shape and size of the decision tree induced from the dataset. Their effectiveness is illustrated by comparing to the results obtained by the classical data characteristics techniques, including DCT that is the most wide used technique in meta-learning and Landmarking that is the most recently developed method and produced better performance comparing to DCT.

1 Introduction

Extensive research has been performed to develop appropriate machine learning techniques for different data mining problems, and has led to a proliferation of different learning algorithms. However, previous work has shown that no learner is generally better than another learner. If a learner performs better than another learner on some learning situations, then the first learner must perform worse than the second learner on other situations [18]. In other words, no single learning algorithm can perform well and uniformly outperform other algorithms over all data mining tasks. This has been confirmed by the 'no free lunch theorems' [31,32]. The major reasons are that a leaming algorithm has different performance in processing different dataset and different learning algorithms are implemented with different search heuristics, which results in variety of 'inductive bias' [15]. In real-world applications, the users need to select an appropriate learning algorithm according to the mining task that they are going to

perform [17,18,1,7,20,12]. An inappropriate selection of algorithm will result in slow convergence, or even produce a sub-optimal solution due to a local minimum.

Meta-learning has been proposed to deal with the issues of algorithm selection [5, 8]. One of the aims of meta-learning is assisting the user to determine the most suitable learning algorithm(s) for the problem at hand. The task of meta-learning is to find functions that map datasets to predicted data mining performance (e.g., predictive accuracies, execution time, etc.). To this end meta-learning uses a set of attributes, called meta-attributes, to represent the characteristics of data mining tasks, and search for the correlations between these attributes and the performance of learning algorithms in general or the optimal learning algorithm in particular [5,10,12]. Instead of executing all learning algorithms to obtain the optimal one, meta-learning is performed on the meta-data characterising the data mining tasks. Algorithm selection is performed by executing the meta-model induced on the characteristics of the dataset.

Three basic procedures are involved in meta-learning: 1) describing the characteristics of learning tasks using a set of meta-attributes; 2) developing the correlations between the meta-attributes and the performance of learning algorithms or the optimal learning algorithms, which is called meta-knowledge; 3) to search, given a new leaming task, the optimal learning algorithm(s) according the developed meta-knowledge. It is obvious that the effectiveness of meta-learning is largely dependent on the description of tasks (i.e., meta-attributes). Several techniques have been developed, such as data characterisation techniques (DCT) [13] to describe the problem to be analyzed, including simple measures (e.g. number of attributes, classes et al.), statistical measures (e.g. mean and variance of numerical attributes), and information theory-based measures (e.g. entropy of classes and attributes). There is, however, still a need for improving the effectiveness of meta-learning by developing more predictive meta-attributes and selecting the most informative ones [9].

In [3], the authors suggested to characterize dataset by measuring the characteristic of models induced on the dataset. Inspired by this idea, this paper presents new methods to measure the complexity of classification data mining tasks. The complexity of data mining tasks is related to the characteristics of datasets and the inductive bias of learning algorithms. The basic idea is to investigate the possibility of capturing dataset characteristics by measuring the properties of a decision tree induced from the dataset, i.e., to measure the structural shape and size of the tree generated by standard methods (c5.0 [22] is used in this paper). More specifically, these measures capture the structural properties of decision tree by some simple measures counting the number of nodes, leaves and attributes in the tree. The extracted meta-attributes have been applied in ranking-based meta-learning for classification algorithm selection. The experimental results clearly show the enhancement of ranking performance compared to the DCT techniques, which is the most commonly used technique, and landmarking, a recently introduced technique [19,2].

This paper is organized as following. In section 2, some related work is introduced, including meta-learning methods for algorithm selection and data characterisation techniques. The proposed method for characterising the datasets is stated in detail in section 3. Experiments illustrating the effectiveness of the proposed method are de-

scribed in section 4. Section 5 concludes the paper, and points out interesting possibilities for future work.

2 Related Work

Two basic factors are involved in meta-learning: the description of the learning tasks (datasets), and the correlation between the task description and the optimal learning algorithm. The first aspect is associated to techniques to characterise datasets with meta-attributes, whilst the second is the learning at meta-level, which develops the meta-knowledge for selecting appropriate algorithm in classification.

2.1 Work Related to Meta-Learning for Algorithm Selection

For algorithm selection, several meta-learning strategies have been proposed [6,25,26]. In general, there are three options in generating the output of the meta-learner. One is to select a single learning algorithm, i.e. to select the algorithm that is expected to produce the best model for the dataset. The second is to select a sub-group of learning algorithms, including not only the best algorithm but also the algorithms that are not significant worse that the best one. The third possibility is to rank the learning algorithms according to their performance. The ranking will assist the user to finally select the learning algorithm. This ranking-based meta-learning is the main approach in the Esprit Project MetaL (www.metal-kdd.org).

Ranking the preference order of algorithms is performed based on estimating the performance of algorithms. In data mining, performance can be measured not only in term of accuracy but also time or understandability of model generated. In this paper, we assess performance with the Adjusted Ratio of Ratios (ARR) measure, which combines the accuracy and time. ARR gives a measure of the advantage of a learning algorithm over another algorithm in terms of their accuracy and the execution time for a specific dataset. The user can adjust the importance of accuracy relative to time by a tunable parameter. The 'zoomed ranking' method proposed by Soares [26] based on ARR, which will be described briefly in section 4.1, is used in this paper for algorithm selection, taking into account of accuracy and execution time simultaneously.

2.2 Work Related to Dataset Characterization

As different learner exhibit sensitivity to specific characteristics of the dataset, the task of meta-learning is to model how these characteristics affect the relative performance of different learning algorithms, and then predict the preference for each learning algorithms before performing data mining process. The methods used to describe the characteristics of the dataset were called Data Characterization Tool (DCT) [13].

The first attempt to characterise datasets in order to predict the performance of classification algorithm was done by Rendell et al. [23]. So far, two main strategies

have been developed in order to characterise a dataset for algorithm selection. First one describes the properties of datasets using statistical and informational measures. In the second one a dataset is characterised using the performance (e.g. accuracy) of a set of simple learners, called landmarker [19,2].

The description of a dataset in terms of its information/statistical properties, appeared for the first time within the framework of the STATLOG project [14]. The authors used a set of 15 characteristics, spanning from simple ones, like the number of attributes or the number of examples, to more complex ones, such as the first canonical correlation between the attributes and the class. This set of characteristics was later applied in various studies, aimed at solving the problem of algorithm selection [5,29,27]. They distinguish three categories of dataset characteristics, namely simple, statistical and information theory based measures. Statistical characteristics are mainly appropriate for continuous attributes, while information theory based measures are more appropriate for discrete attributes. Linder and Studer [13] provide an extensive list of information and statistical measures of a dataset computed for each attribute or pairs of attributes. They provide a tool for the automatic computation of these characteristics, which was called Data characterisation Tools (DCT). However, they pointed out that only a limited set of these measures is relevant in providing recommendation, which in fact was very similar to the one defined in STATLOG. Sohn [27], also uses the STATLOG set as a starting point, and she proceeds with careful evaluation of their properties in a statistical framework. She discovers that some of the characteristics are highly correlated, and she omits the redundant ones from her study. Furthermore, she introduces new features that are transformation or combinations of the existing ones, like ratios or seconds powers, with the goal of providing successful predictions.

An alternative approach to characterise datasets called landmarking was proposed in [19,2]. The intuitive idea behind landmarking is that the performance of simple learner, landmarker, can be used to predict the performance of given candidate algorithms. That is, given landmarker A and B, if we know landmarker A outperforms landmarker B on the present task, then we could select the learning algorithms that has the same inductive bias of landmarker A to perform this data mining task. It has to be ensured that the chosen landmarkers have quite distinct learning biases. As a closely related approach, Bensusan [3, 33] had also used the information computed from the induced decision trees to characterise tasks in meta-learning, such as the ratio of the number of nodes to the number of the attributes, the ratio of number of nodes to the number of training instances. He listed 10 measures based on the unpruned tree, but the performance of these measures in algorithm selection was not evaluated.

3 The proposed measures for describing data characteristics

The task of characterizing dataset for meta-learning is to capture the information about learning complexity for the dataset. This information should enable the estimation of performance of the given learning algorithms. It should also be computable within a relative short time comparing to the whole learning process, which is desired to be

predictive in estimating the performance of the given learning algorithms. In this section we introduce new measures to measure the characteristics of the dataset based on measuring a variety of properties of a decision tree induced from that dataset.

The major idea here is to measure the complexity of learning by measuring the structure and size of decision tree, and use these measures to predict the model complexity generated by other learning algorithms. We employed the standard decision tree learner, c5.0tree. There are several reasons for selecting decision trees. The major reason is that decision tree has been one of the most popularly used machine learning algorithms in classification, and the induction of decision tree is deterministic, i.e. the same training set always produces the similar structure of decision tree.

Definition. A standard *tree* induced with c5.0 (or possibly ID3 or c4.5) consists of a number of *branches*, one *root*, a number of *nodes* and a number of *leaves*. One branch is a chain of *nodes* from *root* to a *leaf*; and each node involves one attribute. The *occurrence* of an attribute is the number of times the attribute occurs in the tree, which provides the information about the importance of the associated attribute. The *tree width* is defined as the number of lengthways partitions divided by parallel nodes or leave from the leftmost to the rightmost nodes or leave. The *tree level* is defined as the breadth-wise partition of tree at each success branches, and the *tree height* is defined by the number of tree levels, as shown in Fig.1. The *length of a branch* is defined as the number of nodes in the branch minus one.

We propose, based on above notations, to describe decision tree in term of the following three aspects: a) outer-profile of tree; b) statistic for intra-structure: including tree levels and branches; c) statistic for tree elements: including nodes and attributes.

To describe the outer-profile of the tree, the width of tree (*treewidth)* and the height of the tree (*treeheight*) are measured according to the number of nodes in each level and the number of levels, as illustrated in Fig.1. Also, the number of nodes (*NoNode)* and the number of leaves (*NoLeave*) are used to describe the overall property of a tree. In order to describe the intra-structure of the tree, the number of nodes at each level and the length of each branch are counted. Let us represent them with two vectors denoted as NoinL=[v1,v2,…vl] and LofB=[L1,L2,….Lb] respectively, where vi is the number of nodes at the ith level, Lj is the length of jth branch, l and b is the number of levels (treeheight) and number of branches. Based on NoinL and LofB, four measures are generated. The maximum and minimum number of nodes at one level:

$$maxLevel = \max(v_1, v_2, ..v_l) \qquad (1)$$
$$minLevel = \min(v_1, v_2, ..v_l)$$

(As the minLevel is always equal to 1, it is not used.) The mean and standard deviation of the number of nodes and leaves on levels:

$$meanLevel = \left(\sum_{i=1}^{l} v_i\right) \Big/ l, \qquad (2)$$

$$devLevel = \sqrt{\sum_{i=1}^{l} (v_i - meanLevel)^2} \Big/ (l-1)$$

The length of longest and shortest branches:

$$LongestBranch = \max(L_1, L_2, .. L_b) \quad (3)$$
$$ShortestBranch = \min(L_1, L_2, .. L_b)$$

The mean and standard deviation of the length of each branch:

$$meanBranch = \left(\sum_{j=1}^{b} L_j\right) \Big/ b, \quad (4)$$

$$devBranch = \sqrt{\sum_{j=1}^{b} (L_j - meanBranch)^2} \Big/ (b-1)$$

Besides the distribution of nodes, the frequency of attributes used in a tree provides further information regarding the dataset. For that, we calculate the times each attribute is used in a tree, which is represented by a vector NoAtt=[nAtt1, nAtt2, …. nAttm], where nAttk is the number of times the kth attribute is used and *m* is the total number of attributes in the tree. Again, the following measures are used:

The maximum and minimum occurrence of attributes:

$$maxAtt = \max(nAtt_1, nAtt_2, ...nAttm_m) \quad (5)$$
$$minAtt = \min(nAtt_1, nAtt_2, ...nAttm_m)$$

Mean and standard deviation of the number of occurrences of attributes:

$$meanAtt = \left(\sum_{i=1}^{m} nAtt_i\right) \Big/ m, \quad (6)$$

$$devAtt = \sqrt{\sum_{i=1}^{m} (nAtt_i - meanAtt)^2} \Big/ (m-1)$$

As a result, a total of 15 meta-attributes is used in our experiments.

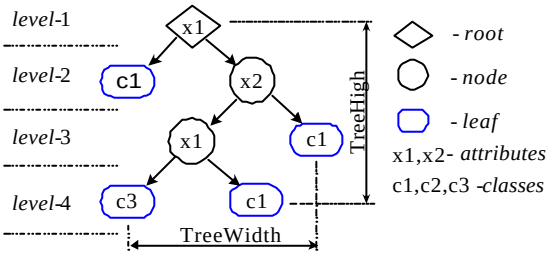


Fig. 1. Structure of Decision Tree.

4 Experimental Evaluation

In this section we experimentally evaluate the proposed data characteristics. In section 4.1 we describe our experimental set-up, in section 4.2 we compare our proposed meta-features with DCT and landmarking, and in section 4.3 we study the effect of meta-feature selection, and compare the performance of DCT and our methods for selected number of meta-features.

4.1 Experimental set-up

The technique of meta-learning employed in this paper is called ranking with *zooming* [26], which includes two phases: 1) training phase to collect the meta-data; 2) reasoning phase to rank the candidate learning algorithms for a given data mining task.

In the training phase, all the benchmark datasets are characterised using the data characterisation methods (e.g., DCT, landmarking or the method proposed in this paper). As a result, one dataset is described with a vector of a set of meta-attributes. These meta-attributes together with the analyzed performance (including accuracy and time) constitute the meta-data. In the reasoning phase, two steps are involved: 1) given a data mining problem (a dataset to analyze), the k-Nearest Neighbor (kNN) algorithm is used to select a subset with k dataset from the benchmark datasets, whose characteristics are similar to the characteristics of the present dataset according to some distance function; this step is called zooming [26]; 2) ranking the order of preference of candidate learning algorithms according to their performance on these datasets selected in zooming phase; this step is named ranking. The ranking is performed based on the *adjusted ratio of ratios* (ARR), a multi-criteria evaluation measure that combine the predicated accuracy and time. ARR has a parameter to enable the user to adjust the relative importance of accuracy and time according to fulfill his particular data mining objective. More details can be found in [26].

To evaluate a recommended ranking, we calculate its correlation to an ideal ranking obtained for the same dataset. The ideal ranking is obtained by estimating the performance of the candidate learning algorithms using 10-fold cross-validation. The similarity between the generated ranking and the ideal ranks is measured using the Spearman's rank correlation coefficient [30].

$$r_s = 1 - \frac{6D^2}{n(n^2-1)}, D^2 = \sum_{i=1}^{n} D_i^2 = \sum_{i=1}^{n} (r_i - \bar{r}_i)^2 \quad (7)$$

where the r_i and $\bar{r}_i$ are the predicted ranking and actual ranking for algorithm i respectively. The bigger r_s is, better of ranking result is, with $r_s = 1$ if the ranking is same as the ideal ranking.

4.2 Comparison with DCT and Landmarking

In our experiments, a total of 10 learning algorithms, including *c5.0tree*, *c5.0boost* and *c5.0rules* [21], Linear Tree (*ltree*), linear discriminant (*lindiscr*), MLC++ Naive Bayes classifier (*mlcnb*) and Instance-based leaner (*mlcib1*) [11], Clementine Multilayer Perceptron (*clemMLP*), Clementine Radial Basis Function (*clemRBFN*) and rule learner (*ripper*), have been evaluated on 47 datasets, which are mainly from the UCI repository [4]. The error rate and time were estimated using 10-fold cross-validation. Our aim in this paper is to evaluate the effect of new proposed meta-attributes (called *DecT* from now on) on ranking of these 10 learning algorithms. In other words, we are interested in comparing the rankings generated by DecT (15 meta-attributes) to the ranking generated by DCT (25 meta-attributes) and Landmarking (5 meta-attributes).

The first experiment is performed to rank the given 10 learning algorithms on the 47 datasets. The leave-one-out method is used to evaluate the performance of ranking, i.e., the performance for ranking the 10 given learning algorithms for each dataset on the basis of the other 46 datasets. In the first experiment, the parameters k=10, Kt=100, meaning that we are willing to tread 1% in accuracy for a 10 times speed-up or slow-down. The ranking performance is measured with r_s (Eq. (15)). The results of ranking performance of using DCT, landmarking and DecT are shown in Fig. 2. The overall average performance for DCT, Landmarking and DecT are 0.613875, 0.634945 and 0.676028 respectively, which demonstrates the improvement of using DecT in ranking algorithms, comparing to DCT and Landmarking.

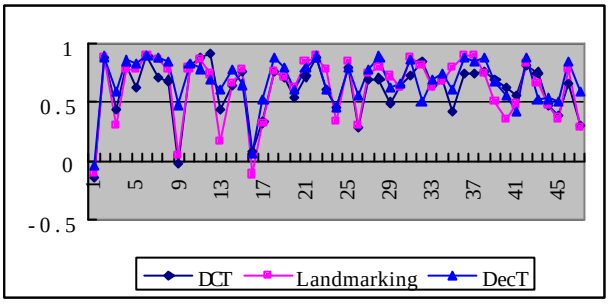


Fig. 2. Ranking performance for 47 datasets using DCT, landmarking and DecT.

In order to look in more detail at the improvement of DecT over DCT and Landmarking, we performed the experiment of ranking using different values of k and Kt. As stated in [26], the parameter Kt represents the relative importance of accuracy and execution time in selecting the learning algorithm (i.e., higher Kt means the accuracy is more important and time is less important). Fig.3 shows the ranking performances of DCT, landmarking and DecT along with different values of Kt={10, 100, 1000}, from which it is observed that, for all the used Kt, DecT improves the performance with different increased degree, comparing to DCT and landmarking.

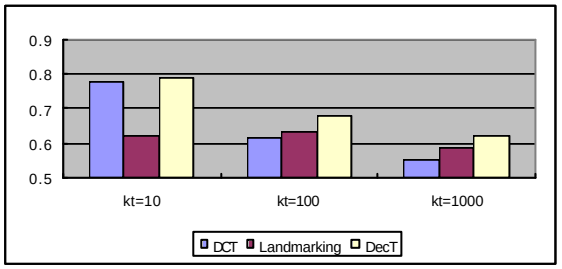


Fig. 3. The ranking performance for different values of Kt.

Fig. 4 shows the performance of ranking based on different zooming degree (different k), i.e., selecting different number of similar datasets, based on which the ranking is performed. From these results, we observe that 1) for all different values of k, DecT

produces better ranking performance than DCT and landmarking; 2) best performance is obtained by selecting 10-25 datasets among 46 datasets.

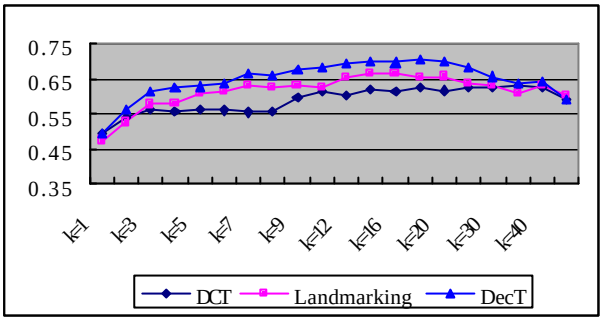


Fig. 4. The ranking performance for different values of k.

4.3 Performing meta-feature selection

The k-nearest neighbor learning method, employed to select k datasets for ranking the performance of learning algorithms for the given dataset, is known to be sensitive to the irrelevant and redundant features. Using smaller number of features could help to improve the performances of k-nearest neighbor learning, as well as to reducing the time used in meta-learning. In our experiments, we manually reduced the number of DCT meta-features from 25 to 15 and 8, and compare their results to those obtained based on the same number of DecT meta-features. The reduction for DCT meta-features is performed by removing the features thought to be redundant, and the features having a lot of *non-appl* values, and the reduction for DecT meta-features are performed by removing redundant features.

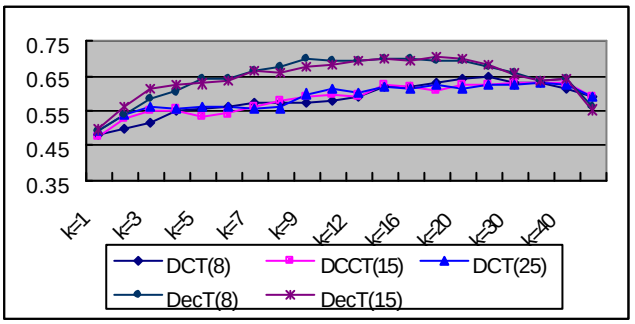


Fig.5. Results for reduced meta-features.

The ranking performances for these reduced meta-features are shown in Fig.5, in which DCT(8), DCT(15), DecT(8) represent the reduced 8, 15 DCT meta-features and 8

DecT meta-features, DCT(25) and DecT(15) represent the full DCT and DecT meta-features respectively. From Fig.5, we can observe that feature selection did not significantly influence the performance of either DCT or DecT, and that the latter outperforms the former across the board.

5 Conclusions and Future Work

Meta-learning strategy, under the framework of MetaL, aims at assisting the user in selecting appropriate learning algorithm for the particular data mining task. Describing the characteristics of dataset in order for estimating the performance of learning algorithm is the key to develop a successful meta-learning system.

In this paper, we proposed new measures to characterise the dataset. The basic idea of is to process the dataset using a standard tree induction algorithm, and then to capture the information regarding the dataset's characteristics from the induced decision tree. The decision tree is generated using standard c5.tree algorithm. A total of 15 measures, which constitute the meta-attributes for meta-learning, have been proposed for describing different kind of properties of a decision tree.

The proposed measures have been applied in ranking the learning algorithms based on accuracy and time. Extensive experimental results have illustrated the improvement of ranking performance by using the 15 meta-attributes generated by the proposed method, compared to the 25 DCT and 5 Landmarking meta-features. In order to reduce the effect of redundant or irrelevant features on the performance of zooming ranking, we also compared the performance based on selected 15 DCT meta-features and DecT, and selected 8 DCT and DecT meta-features. The results suggest that feature selection does not significantly change the performance of either DCT or DecT.

In other experiments, we observed that the combination of DCT with DecT or Landmarking with DCT and DecT did not produce better performance than DecT. This is an issue that we are interested in further investigation. The major reason may come from the use of k-nearest neighbor learning in zooming based ranking strategy. One possibility is to test the performance of the combination of DCT, landmarking and DecT in other meta-learning strategies, such as best algorithm selection. Another interesting subject is to look at the change of shape and size of the decision tree along with the change of examples used in tree induction, as it will be useful if it is possible to capture the data characteristics based on sampled dataset. This is especially important for large datasets.

Acknowledgements: this work is supported by the MetaL project (ESPRIT Reactive LTR 26.357).

References

1. Brodley, C. E.: Recursive automatic bias selection for classifier construction. *Machine Learning*, (1995) 20:63-94.
2. Bensusan, H., and Giraud-Carrier, C.: Discovering Task Neighbourhoods through Landmark Learning Performances. In Proceedings of the 4th European Conference on Principles of Data Mining and Knowledge Discovery. 325-330. Springer. (2000)
3. Bensusan, H., Giraud-Carrier, C., and Kennedy, C.: Higher-order Approach to Meta-learning. In Proceedings of the ECML'2000 workshop on Meta-Learning: Building Automatic Advice Strategies for Model Selection and Method Combination, 109-117. ECML'2000. (2000)
4. Blake, C., Keogh, E., and Merz, C.: www.ics.uci.edu/~mlearn/mlrepository.html. University of California, Irvine, Dept. of Information and Computer Sciences. (1998)
5. Brazdil, P., Gama, J. and Henery, R.: Characterizing the Applicability of Classification Algorithms using Meta Level Learning. Machine Learning-ECML94. (1994) 83-102, Spinger Verlag.
6. Dietterich, T G.: Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation*, , (1998) 10(7):1895-1924.
7. Gordon F. and desJardin, M.: Evaluation and selection of biases. *Machine Learning*, (1995) 20:5-22.
8. Kalousis A. and Hilario, M.: Model Selection via Meta-learning: a Comparative Study. In *Proceedings of the 12th International IEEE Conference on Tools with AI*, Vancouver. IEEE press. (2000)
9. Kalousis, A. and Hilario, M.: Feature Selection for Meta-Learning. In *Proceedings of the 5th Pacific Asia Conference on Knowledge Discovery and Data Mining*. Springer. (2001)
10. Koepf, C., Taylor, C. and Joerg Keller J.: Meta-analysis: Data characterisation for classification and regression on a meta-level. In Antony Unwin, Adalbert Wilhelm, and Ulrike Hofmann, editors, *Proceedings of the International Symposium on Data Mining and Statistics*, Lyon, France, (2000).
11. Kohavi, R.: Scaling up the Accuracy of Naïve-bayes Classifier: a Decision Tree hybrid. Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining, (1996) 202-207.
12. Lagoudakis, M. G. and Littman, M. L.: Algorithm selection using reinforcement learning. In *Proceedings of the Seventeenth International Conference on Machine Learning (ICML-2000)*, 511-518, Stanford, CA. (2000)
13. Linder, C. and Studer, R.: AST: Support for Algorithm Selection with a CBR Approach. Proceedings of the 16th International Conference on Machine Learning, Workshop on Recent Advances in Meta-Learning and Future Work. (1999).
14. Michie, D., Spiegelhalter, D., and Toylor, C.: Machine Learning, Neural Network and Statistical Classification. Ellis Horwood Series in Artificial Intelligence. (1994)
15. Mitchell, T.: Machine Learning. MacGraw Hill. (1997)
16. Salzberg. S.: On comparing classifiers: Pitfalls to avoid and a recommended approach. *Data Mining and Knowledge Discovery, Vol. 1*. (1997)

17. Schaffer, C.: Selecting a Claassification Methods by Cross Validation, Machine Learning, 13, 135-143. (1993)
18. Schaffer, C.: Cross-validation, stacking and bi-level stacking: Meta-methods for classification learning. In P. Cheeseman and R. W. Oldford, editors, *Selecting Models from Data: Artificial Intelligence and Statistics IV*, pages 51-59. Springer-Verlag. (1994)
19. Pfahringer, B., Bensusan, H., and Giraud-Carrier, C.: Landmarking various Learning Algorithms. Proceedings of the 17th International Conference on Machine Leaming. 743-750. Morgan Kaufman. (2000)
20. Provost F.J. and Buchanan B. G: Inductive policy: The pragmatics of bias selection. *Machine Learning*, 20:35-61. (1995)
21. Quinlan, J. R.: C4.5: Programs for Machine Learning, Morgan Kaufman. (1993)
22. Quinlan, J. R.: c5.0: An Informal Tuorial, RuleQuest, www.rulequest.com/see5-unix.html. (1998).
23. Rendell, L. Seshu, R., and Tcheng, D.: Layered Concept Learning and Dynamically Variable Bias Management. 10th Inter. Join Conference on AI. 308-314. (1987).
24. Schaffer, C.: A Conservation Law for Generalization Performance. Proceedings of the 11th International Conference on Machine Learning. (1994).
25. Soares, C.: Ranking Classification Algorithms on Past Performance. Master's Thesis, Faculty of Economics, University of Porto. (2000)
26. Soares, C.: Zoomed Ranking: Selection of Classification Algorithms based on Relevant Performance Information. Proceedings of the 4th European Conference on Principles of Data Mining and Knowledge Discovery, 126-135. Springer. . (2000)
27. Sohn, S.Y.: Meta Analysis of Classification Algorithms for Pattern Recognition. IEEE Trans. on Pattern Analysis and Machine Intelligence, (1999) 21, 1137-1144.
29. Todorvoski, L. and Dzeroski, S.: Experiments in Meta-Level Learning with ILP. Proceedings of the 3th European Conference on Principles on Data Mining and Knowledge Discovery, 98-106. Springer. (1999)
30. Webster, A.: Applied Statistics for Business and Economics, Richard D Irwin Inc, 779-784. (1992).
31. Wolpert, D.: The lack of a Priori Distinctions between Learning Algorithms. Neural Computation, 8, 1341-1390. (1996)
32. Wolpert, D.: The Existence of a Priori Distinctions between Learning Algorithms. Neural Computation, 8, 1391-1420. (1996).
33. Bensusan, H. God doesn't always shave with Occam's Razor - learning when and how to prune. In Proceedigs of the 10th European Conference on Machine Learning, pages 119--124, Berlin, Germany, April 1998. Springer.

Meta-Learning for Stacked Classification

Alexander K. Seewald

Austrian Research Institute for Artificial Intelligence, Schottengasse 3
A-1010 Wien, Austria; alexsee@oefai.at

Abstract. In this paper we describe new experiments with the ensemble learning method Stacking. The central question in these experiments was whether meta-learning methods can be used to accurately predict various aspects of Stacking's behaviour. The resulting contributions of this paper are two-fold: When learning to predict the accuracy of stacked classifiers, we found that the single most important feature is the accuracy of the best base classifier. A simple linear model involving just this feature turns out to be surprisingly accurate. When learning to predict significant differences between Stacking and three common meta-classification methods, we have found simple models, all but one of which are based on single features which can be efficiently computed directly from the dataset. For one of these models, we were able to offer an interpretation. These models may ultimately be used to decide in advance which meta-classification scheme to use on a given dataset, since neither of them is always the best choice. Furthermore, aiming to understand these models can lead to new insights into Stacking's behaviour.

1 Introduction

Meta-learning focusses on predicting the right algorithm for a particular problem based on characteristics of the dataset [3] or based on the performance of other, simpler learning algorithms [6]. Here we are concerned with meta-learning of *meta-classification schemes*. Stacking can be considered the best-known such scheme and was introduced in [11]. We take a more general view of meta-learning and use it to predict two aspects of Stacking's behaviour: accuracy as estimated via ten-fold crossvalidation; and also significant differences vs. other common meta-classification schemes. We use Stacking in the extension proposed in [10].

2 Experimental setup

In our experiments, we used twenty-six datasets from the UCI machine learning repository [2]. Details can be found in [9]. We used Stacking with all of the following seven base classifiers for our experiments, which were chosen in an attempt to maximize diversity. All algorithms were taken from the Waikato Environment for Knowledge Analysis (WEKA[1]), Version 3-1-8.

- DecisionTable: a decision table learner.
- IBk: the IBk instance-based learner using K=1 nearest neighbors.
- J48: a Java port of C4.5 Release 8 [7]
- KernelDensity: a simple kernel density classifier.

[1] The Java source code of WEKA has been made available at www.cs.waikato.ac.nz.

- KStar: the K* instance-based learner [4], using all nearest neighbors.
- MLR: a multi-class learner based on linear regression, which separates each class from all other classes by linear discrimination (*Multi-response Linear Regression*)
- NaiveBayes: the Naive Bayes classifier using kernel density estimation (-K)

We used the following four meta-classification schemes.

- Stacking is the stacking algorithm as implemented in WEKA, which follows [10]. It constructs the meta dataset by adding the entire predicted class probability distribution instead of only the most likely class. We used MLR as the level 1 learner.
- X-Val chooses the best base classifier on each fold by an internal ten-fold CV. This is just the selection by cross-validation we mentioned in the beginning.
- Voting is a straight-forward adaptation of voting for distribution classifiers, i.e. the mean class distribution of all classifiers is calculated. It is the only scheme which does not use an expensive internal cross-validation.
- Grading is an implementation of the grading algorithm evaluated in [8] which uses IBk ($K = 10$) as meta-classifier.

We used seventeen dataset-related features which characterize the dataset, inspired by [3]. A reference implementation is available from the author upon request.

- *Inst*, the number of examples.
- *log(Inst)* which is the natural logarithm of *Inst*.
- *Classes*, the number of classes.
- *Attrs*, the number of attributes (excluding the class)
- *PropNomAttrs*, number of nominal attributes as a proportion of *NumAttrs*.
- *PropContAttrs*, number of numeric attributes as a proportion of *NumAttrs*.
- *PropBinAttrs*, number of binary-valued attributes as a proportion of *NumAttrs*.
- *ClassEntropy*, the entropy of the class attribute.
- *AttrEntropy*, the entropy of all attributes.
- *MutualEntropy*, the mutual entropy of class and attributes.
- *EquivAttrs*, the equivalent number of attributes, $\frac{ClassEntropy}{MutualEntropy}$
- *RelEquivAttrs*, $\frac{EquivAttrs}{Attrs}$
- *S/N*, the signal-to-noise ratio.
- *MeanAbsCorr*, the mean absolute correlation over all pairs of numeric attributes.
- *MeanAbsSkew*, the mean absolute skew of all numeric attributes.
- *MeanAbsKurtosis*, the mean absolute kurtosis of all numeric attributes.
- *defAcc*, the default accuracy, i.e. the proportion of the most common class.

Additionally, we used the accuracies of our seven base-learners as features. We also calculated standard statistical features of this set of seven accuracies. Furthermore, we used the same statistical features over pairwise base classifier κ-statistics[2].

- 7 accuracies, one for each base classifier (*DT, IBk-K1, J48, KD, KStar, MLR, NB-K*)
- 8 statistical features describing the set of accuracy values (*MinAcc, MaxAcc, MeanAcc, StDevAcc, SkewAcc, SkewAcc*^{2}*, KurtosisAcc, relRangeAcc* $= \frac{MaxAcc-MinAcc}{StDevAcc}$)
- Eight statistical features describing the set of all pairwise κ-statistics between base classifiers (*MinK, MaxK, MeanK, StDevK, SkewK, SkewK*^{2}*, KurtosisK, relRangeK*)
- *relMeanAcc*=$\frac{AvgAcc}{defAcc}$, the ratio of average accuracy to default accuracy.

The above features were computed both on predictions estimated from the full data set (training set accuracy and diversity) and on predictions estimated via tenfold crossvalidation. For meta-learning of significant differences, we only used the latter set because it consistently offered better estimates during the first task. This also simplified the experimental evaluation. All statistical differences for meta-learning were computed via a t-Test with α=99%.

[2] 1.0 stands for identical predictions between two learners while 0.0 represents random correlations. A negative value signifies systematic disagreement, see [5].

3 Estimating Stacking's Accuracy

This section is concerned with predicting the accuracy of Stacking. In order to obtain a reasonable estimate, a ten-fold CV was used for accuracy estimation. We first investigated the simplest models: based on only a single feature. Thus, we assumed linear relationships between each feature and the accuracy of our stacked classifier and characterized this relation by statistical correlation coefficients and mean absolute errors (MAE). Afterwards, we considered more complex and non-linear models obtained by various regression algorithms from machine learning.

We computed statistical correlation coefficients and mean absolute errors (MAE) for all our features, always versus the accuracy of the stacked classifiers. Space restrictions prevent us from showing detailed results, which can be found in [9].

Correlations and MAEs were determined for all meta-data (**All**) and also via leave-one-out crossvalidation (**CV**). In the former, this estimate was based on the output of one linear regression model computed from all meta-examples. In the latter case, the estimate was based on twenty-six linear models which were trained using all but one meta-example and tested on the last one. This latter case is a more reliable indicator of model performance on unseen data than the former.

In the case of base-classifier related features, we have an additional dimension: we can estimate the base classifier accuracies on the full dataset (**AllT, CV T**, i.e. training set accuracies) or via tenfold crossvalidation (**All, CV**), yielding two different set of features. Since Stacking uses CV internally, we expect **All** and **CV** to be better predictors for stacked accuracy. This is indeed the case – a single feature, *MaxAcc*, already yields excellent results. However, computing a crossvalidation on the original dataset comes with a non-negligible computational cost. A computational cost reduction by an order of magnitude could be obtained by using training set output to compute our features – which motivates **AllT** and **CV T**. As expected, in this case we get less good but still acceptable results for best single feature, *MeanAcc*.

As should be expected from a high-bias linear model, all base-classifier related features show a graceful degradation from **All** to **CV**. We were surprised to note that this is not always true for the dataset-related features - about half of the features have a negative correlation for **CV** whose absolute value is higher than the positive correlation for **All**. This higher negative correlation can unfortunately not be used to predict stacked accuracy[3] and is always coupled to a large MAE. It seems that a lot of the dataset-related features are not relevant to this task or that a one-dimensional linear model is not appropriate to find a relevant relation.

In order to test how we may improve our results by using multiple features, we resorted to using standard machine-learning approaches for regression on our meta-dataset. We created one meta-dataset with accuracy estimation via training set (*MetaTrain*) and one estimated via tenfold CV (*MetaCV*). The dataset-related features were included in both cases. We evaluated linear regression, LWR (*locally weighted regression*), model trees, regression trees, KStar and IBk instance based learners at the meta-level. Linear regression and model trees proved superior[4]. However,

[3] The maximum negative correlation appears in feature *defAcc* (-0.94; **CV**) This correlation is based on twenty-six different models, one per leave-one-out training fold. All data would have to be used to determine the final regression line, but then this result can no longer be validated and seems certainly too optimistic.

[4] Both were always best by highest correlation and lowest MAE.

we were still unable to find any model which performed better than the best linear model based on a single feature.

Concluding, features derived from classifiers seem to be more relevant in the context of predicting accuracy than those derived directly from the datasets, which was also found in [1]. For example, the formula $StAcc = 1.074 * MaxAcc - 0.082$ predicts Stacking's accuracy with a correlation of 0.96 and a MAE of 0.022. Notice that although it seems at first glance that Stacking performs slightly worse than the best component classifier, this view is biased: $MaxAcc$, i.e. the best base classifier by hindsight, is a less fair comparison than accuracy of X-Val since its decision is based on all available data while X-Val and Stacking only see the training data from the leave-one-out CV, i.e. all but one meta-instance. Notice also that while computing $MaxAcc$ leaves us with a lot of data which could be used directly by Stacking, this would only enable us to compute the training set accuracy for Stacking and not the ten-fold cv estimate we used here.

Given our results, it is surprising that other meta-learning approaches have not considered that quite simple models may suffice, but instead rely on complex models whose interpretation may be quite difficult.

4 Meta-Learning of Significant Differences

This section is concerned with predicting significant differences between Stacking and three other meta-classification schemes. For each of Stacking vs. Voting, Stacking vs. Grading and Stacking vs. X-Val, we generated a separate meta-dataset consisting of all dataset-related and classifier-related features[5] followed by a binary class variable, being 1 if Stacking is significantly better than the other scheme and 0 otherwise. In case there is no significant difference, we removed the respective example from the meta-dataset, under the premise that in this case we can consider both variants to be equivalent and thus judge either answer to be correct.

On these meta-datasets, we evaluated a number of standard machine learning algorithms available in WEKA[6] via leave-one-out crossvalidation. We only discuss the best models which in most cases seem to be rather simple and based on single attributes only, hinting that they may be robust. In one case, insight into the workings of both meta-classification schemes suggests an interpretation.

For Stacking vs. Voting, there are twelve datasets without significant differences. After removing them from our meta-dataset, we have fourteen instances, seven with class=1, seven with class=0. The baseline accuracy is thus 50%. Here, IBk is the best meta-learner with an accuracy of 92.86% and a single error for *vote*. A cross-validation using only seven folds produces the exact same result.

When removing the base-classifier dependent features, IBk is still the best classifier with an additional error on *labor*, the smallest dataset. In this case MLR, another high-bias and global learner, is equally good. So we may tentatively conclude that for this meta-dataset, there seems to be no single feature which can predict the significant differences as good as a combination of all features.

For Stacking vs. Grading, there are again twelve datasets on which there are no significant differences. After removing them from our meta-dataset, we have fourteen

[5] Because of the much better results in predicting stacked classifier accuracy and also to simplify our experiments, we only considered those classifier features estimated via CV.

[6] All base learners plus 1R and DecisionStump.

instances whose classes are again equally distributed. Thus the baseline accuracy is also 50%. Here, J48 is the best choice with 92.86% accuracy and only a single error on the smallest dataset, *labor*. The training set model is based on a single attribute, *PropNomAttr*. In all fourteen folds but two there is the same model[7], which also appears as the training set model. In the two other folds, the same attribute appears in the same formula with 0.65 and 0.695652 resp. as value on the right side. It seems that the proportion of nominal attributes plays a role on the performance between Stacking and Grading: in case there about $\frac{2}{3}$ or less of the attributes are nominal, Stacking works significantly better than Grading.

A smaller proportion of nominal attributes makes learning harder for the base-learners, since most of them are better equipped to handle nominal data. Stacking seems to be able to compensate for this, since its meta-level data is independent of the base-level data[8] and is processed by MLR which is among all base learners best equipped to handle numeric data. However, Grading seems to be unable to compensate for this since its meta-level data contains just the base-level attributes. Thus its meta learner IBk can be expected to be susceptible in the same way as the base learners.

For Stacking vs. X-Val, seventeen examples offer no significant differences. Only nine examples remain for our experiments, the baseline accuracy is already 66.7%. Interestingly in this case the best model is from DecisionStump which learns a single J48 node, obtaining 88.9% accuracy, corresponding to a single error on dataset *balance-scale*. It seems J48 is prone to overfitting on this meta-dataset. The training set model[9] is based on *MeanAbsSkew* and appears in seven folds. Once the same model appears with value 0.53 instead of 0.31. Once a model based on $numClasses <= 13 : class = 1$ appears. The same overall accuracy is also obtained in a six-fold cross-validation.

5 Related Research

Up to now there is no research aiming to either predict the accuracy of meta-classification schemes or to predict which meta-classification scheme to use for a given dataset. In this paper we have investigated both tasks and found them to work quite well.

6 Conclusion

In this paper we have investigated the use of machine learning techniques in the context of meta-learning both to predict stacked classifier accuracy and significant differences between Stacking and three other meta-classification schemes. We used both dataset-related and base-classifier related features in our tasks.

In the context of predicting classifier accuracy, we found that classifier-related features, namely some of those derived from accuracy, are excellently suited to this task, as have others, [1, 6]. As feature, the accuracy of the best component classifier in the ensemble is able to predict the accuracy of the stacked classifier quite well. Other meta-learning approaches seem not to take into account that such simple models may be competitive to more complex models, but far much easier to understand.

[7] $IF\ PropNomAttr <= 0.684211\ THEN\ class = 1\ OTHERWISE\ class = 0$

[8] Meta-level data for Stacking = class probability distributions from all base learners.

[9] $IF\ (MeanAbsSkew <= 0.31\ OR\ missing)\ THEN\ class = 0\ OTHERWISE\ class = 1$

In the second part of the paper we investigated the prediction of significant differences between stacking and other meta-classification schemes. In this case we found that features derived directly from the dataset were usually better suited. For the model which predicts significant differences between Grading and Stacking, intimate knowledge of the inner workings of both schemes have enabled us to formulate a tentative explanation of the learned model.

At last we have found that there is no single best meta-classifier for predicting significant differences – a variety of machine learning algorithms had to be evaluated for best results. Although most of our best models were based on single features, it seems that no single learning algorithm is able to find all of them. This hints that pairwise learning problems have quite different properties, which may explain why meta-learning is usually so hard.

Acknowledgements

This research is supported by the Austrian *Fonds zur Förderung der Wissenschaftlichen Forschung (FWF)* under grant no. P12645-INF. This research was also partially supported by the ESPRIT LTR project METAL (26.357). The Austrian Research Institute for Artificial Intelligence is supported by the Austrian Federal Ministry of Education, Science and Culture. We would like to thank Johannes Fürnkranz for valuable comments.

References

1. Bensusan, H., Kalouis, A.: Estimating the Predictive Accuracy of a Classifier. In Proceedings of the twelveth European Conference on Machine Learning (2001), Freiburg, Germany, 25–36. Springer Verlag.
2. Blake, C. L., Merz, C. J: UCI repository of machine learning databases. `http://www.ics.uci.edu/~mlearn/MLRepository.html` (1998). Department of Information and Computer Science, University of California at Irvine, Irvine CA.
3. Brazdil, P. B., Gama, J., & Henery, B. Characterizing the applicability of classification algorithms using meta-level learning. *Proceedings of the 7th European Conference on Machine Learning (ECML-94)* (83–102). Catania, Italy: Springer-Verlag.
4. Cleary, J. G., Trigg, L. E: K*: An instance-based learner using an entropic distance measure. Proc. 12th International Conference on Machine Learning (1995) 108–114, Lake Tahoe, CA.
5. Dietterich, T. G: Ensemble methods in machine learning. In Kittler, J., Roli, F., First International Workshop on Multiple Classifier Systems (2000) 1–15. Springer-Verlag.
6. Pfahringer, B., Bensusan, H., & Giraud-Carrier, C. (2000). Meta-learning by landmarking various learning algorithms. *Proceedings of the 17th International Conference on Machine Learning (ICML-2000)*. Stanford, CA.
7. Quinlan, J. R: C4.5: Programs for Machine Learning (1993). Morgan Kaufmann, San Mateo, CA.
8. Seewald A.K., Fürnkranz J.: An Evaluation of Grading Classifiers, in Hoffmann F. et al. (eds.), Advances in Intelligent Data Analysis, Proc. 4th International Conference, IDA 2001, Springer, 115–124.
9. Seewald A.K.: Meta-Learning for Stacked Classification (ext.vers.). Technical Report, Austrian Research Institute for Artificial Intelligence, Vienna, TR-2002-05, 2002.
10. Ting, K. M., Witten, I. H: Issues in stacked generalization. Journal of Artificial Intelligence Research *10* (1999) 271–289.
11. Wolpert, D. H: Stacked generalization. Neural Networks *5(2)* (1992) 241–260.

Knowledge-based Selection of Data Characteristics for Algorithm Recommendation Using Ranking Methods

Carlos Soares and Pavel Brazdil

LIACC/Faculty of Economics, University of Porto, R. Campo Alegre 823, 4150-180 Porto, Portugal, {csoares,pbrazdil}@liacc.up.pt

Abstract. We show that information about the past performance of algorithms can be used for algorithm recommendation with small loss in accuracy and significant savings in experimentation time, when compared to cross-validation. This result is obtained with a meta-learning approach that uses a set of data characteristics that were manually selected using our knowledge of the algorithms. We also demonstrate the advantage of providing recommendation in the form of a ranking.

1 Introduction

Ideally, we would like to be able to identify or design the single best algorithm to be used in all situations. However, both experimental results and theoretical work indicate that this is not possible. Therefore, the choice of which algorithm(s) to use depends on the data set at hand and systems that can provide such recommendations would be very useful. We could reduce the problem of algorithm recommendation to the problem of performance comparison by estimating the performance of all the algorithms on the data currently available, assuming that it is representative of future data. Cross-validation (CV) is the most accurate method available for that purpose. However, it is not usually feasible in practice because there are too many algorithms to try out, some of which may be quite slow. Another approach to algorithm recommendation involves the use of meta-knowledge, that is, knowledge about the performance of algorithms, which is followed here.

The performance and the usefulness of meta-learning for algorithm recommendation depends on several issues, namely the measures used to characterize data sets, the type of recommendation provided and the meta-learning method used. Here we focus on the former and, so, we must make appropriate choices for the others. Another important issue for algorithm recommendation is the criteria used to evaluate the algorithms (e.g. accuracy, interpretability of models). Here we will concentrate on accuracy. One example of a multicriteria meta-learning approach can be found in [1].

Concerning the type of recommendation provided, we opted for a ranking of the algorithms. This approach is more flexible than the recommendation of a single algorithm or of a small set of algorithms which is expected to perform not

significantly worse than the best one, which are commonly used in meta-learning [2, 3]. Flexibility is important in the algorithm recommendation setting because it is not known beforehand how many alternatives the user will actually take into account. Furthermore, suppose that a single algorithm is recommended and that this algorithm fails, e.g. because it has a bug or uses too much memory. Given that no information regarding the expected performance of the other algorithms is provided, the user is left without guidance. Such a situation will not occur with a ranking because the user may simply try the next algorithm. Algorithm recommendation using meta-learning was first handled as a ranking task by [4]. Recently, there has been a growing interest in this approach (e.g. [5]).

The choice of ranking for the type of recommendation has limited our choice of meta-learners. We adopted an IBL framework that uses the k-Nearest Neighbor algorithm combined with a method to aggregate and rank performance information is selected. A few alternative ranking methods have been described in [5]. Here we have opted by the average ranks method (AR), which is simple and competitive [5]. This method consists of calculating the average rank of each algorithm on the neighbor datasets and ranking the algorithms accordingly. More details can be found in [1].

We start by describing the data characteristics selected (Section 2), then we present a comparative evaluation of this subset against the original set of measures (Section 2.1). Finally, we present some conclusions.

2 Selection of Data Characteristics

The most important issue in meta-learning is probably data characterization. We need to extract measures from the data that characterize relative performance of the candidate algorithms and that can be computed significantly faster than running those algorithms. It is known that the performance of different algorithms is affected by different data characteristics. For instance, the performance of k-Nearest Neighbor will suffer if there are many irrelevant attributes. Most work on meta-learning uses general, statistical and information theoretic (GSI) measures or *meta-attributes* [4]. Recently, other approaches to data characterization have been proposed, namely *landmarkers* [3] and model-based characterization [6], which we will not address here.

Many measures have been proposed for data characterization in the GSI approach. However, some of them may be irrelevant, others may not be adequately represented (e.g., the proportion of numeric attributes is probably more informative than the number of numeric attributes), while some important ones may be missing. Furthermore, given that the performance information available typically includes relatively few examples (data sets), a large number of meta-features creates the danger of overfitting. However, in most previous meta-learning work, not enough effort has been dedicated to selecting an appropriate subset of data characteristics. An exception is the work of [7], who applied a wrapper-based feature selection method to a large number of meta-features (and combinations of meta-features). The drawback of this approach is that many hypotheses are

tested when compared to the number of examples. This increases the probability of finding a subset of the data characteristics that obtains good performance merely by chance.

Here, we follow a knowledge engineering approach. Based on our expertise on the learning algorithms used and on the properties of data that affect their performance, we select and combine existing GSI measures to define *a priori* a small set of meta-features that are expected to provide information about those properties. The measures and the properties which they are expected to represent are presented in Table 1. All three proportional features proposed (2nd to 4th features) represent combinations of previously defined data characteristics. The number of examples represents one aspect of scalability, which is also affected by other data characteristics, like number of attributes and number of values in symbolic attributes. The need for a measure that combines all these characteristics remains. A numeric attribute is considered to have outliers, possibly due to noise, if the ratio of the variances of mean value and the α-trimmed mean is smaller than 0.7. We have used $\alpha = 0.05$.

Table 1. Properties of algorithms which affect relative performance and data characteristics that affect those properties. More details about the basic features used here can be found in [8].

Property	Measure
Scalability	Number of examples
Preference for symbolic or numeric attributes	Proportion of symbolic attributes
Robustness to missing values	Proportion of missing values
Robustness to outliers	Proportion of numeric attributes with outliers
Number of classes	Class entropy
Class frequency	Class entropy
Useful information in symbolic attributes	Average mutual information of class and symbolic attributes
Useful information in numeric attributes	Canonical correlation of the most discriminating single linear combination of numerical attributes and the class distribution

2.1 Empirical Evaluation and Comparison

Our meta-data consists of 53 data sets mostly from the UCI repository [9] but including a few others from the METAL project[1] (SwissLife's Sisyphus data and a few applications provided by DaimlerChrysler). Ten algorithms were executed on those data sets[2]: two decision tree classifiers, C5.0 and Ltree, which is a decision

[1] Esprit Long-Term Research Project (#26357) *A Meta-Learning Assistant for Providing User Support in Data Mining and Machine Learning* (www.metal-kdd.org).

[2] References for these algorithms can be found in [3].

tree that can introduce oblique decision surfaces; the IB1 instance-based and the naive Bayes classifiers from the MLC++ library; a local implementation of the multivariate linear discriminant; two neural networks from the SPSS Clementine package (Multilayer Perceptron and Radial Basis Function Network); two rule-based systems, C5.0 rules and RIPPER; and an ensemble method, boosted C5.0. Results were obtained with 10-fold cross-validation using default parameters on all algorithms.

To assess whether the subset of meta-features yields better rankings, we use a methodology for ranking evaluation and comparison that has been proposed earlier for meta-learning [5]. The rankings recommended by the ranking methods are compared against the true observed rankings using Spearman's rank correlation coefficient. We note that the performance of two or more algorithms may be different but not with statistical significance. To address this issue, we exploit the fact that in such situations the tied algorithms often swap positions in different folds of the N-fold cross-validation procedure which is used to estimate their performance. Therefore, we use N orderings to represent the true ideal ordering, instead of just one. The correlation between the recommended ranking and each of those orderings is calculated and its score is the corresponding average. Leave-one-out was used to estimate meta-level performance.

The mean average correlation for increasing number of neighbors obtained by the AR ranking method using the original set of 25 measures and the manually selected subset (Section 2) is shown on the right-hand side of Figure 1. We observe that the results are better with the reduced set than with the extended set. In fact, the combination of Friedman's test and Dunn's Multiple Comparison Procedure (significance levels of 5% and 25%, respectively), which is appropriate for this kind of comparison, shows that the AR with the reduced set of measures is significantly better than the same ranking method with the full set of measures and the baseline method of aggregating all performance information. More details about this comparison methodology can be found in [5]. We also observe that the quality of the rankings obtained with the reduced set decreases as the number of neighbors increases. This is not true when the extended set is used. This indicates that the space of meta-features is an approximation of the space of relative performance of the algorithms. This means that we were the measures selected are indeed representative of properties that affect relative algorithm performance. The shape of the curves also indicates that the extended set probably contains many irrelevant features, which, as is well known, affects the performance of the k-NN algorithm used at the meta-level.

One may ask how do these results reflect in terms of the quality of the advice provided, in the perspective of the user. Figure 1 shows a comparison of our meta-learning method with the cross-validation strategy (left-hand side), which is the most accurate algorithm selection method (an average accuracy of 89.93% in our setting) but it is very time consuming (approximately four hours on average, in our setting) and boosted C5.0, which is the best algorithm on average (87.94%) and also very fast (less than two min.). One could argue that, with such a small margin for improvement (2%), it is not worthwhile to

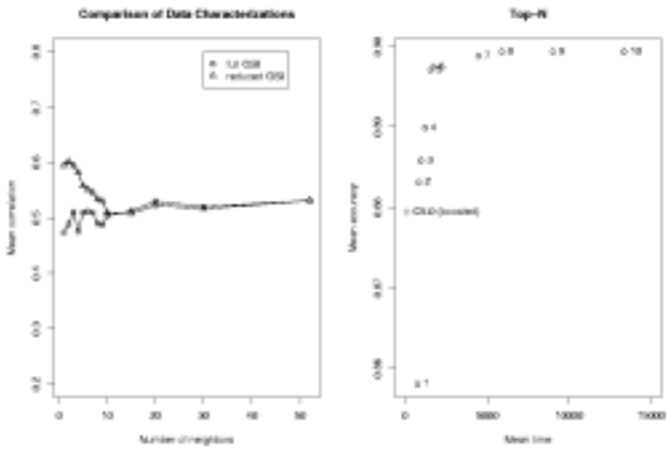


Fig. 1. Mean correlation obtained by AR ranking method for increasing number of neighbors using two sets of GSI data characteristics: full and reduced (on the right). Average accuracy versus average execution time for the strategy of executing the top-N algorithms in the recommended ranking, for all possible values of N, and for the strategy of always executing boosted C5.0 (on the left).

do algorithm selection: choosing boosted C5.0 will provide quite good results on average. However, in some applications (e.g. cross-selling in a website that sells thousands of items daily), an improvement of 2% or even less may be significant from a business point-of-view. The strategy of executing the algorithm ranked in the first position is worse than always executing boosted C5.0. However, if we use the full potentially of a ranking method, and execute the Top-2 algorithms in the ranking, the time required is larger than boosted C5.0's, although still acceptable in many applications (less than 15 min.) but the loss in accuracy would be only 1.62%. Running two more algorithms another algorithm would provide further improvement in accuracy (1.35% and 0.95% losses) while taking only a little longer (16 and 20 min.).

3 Conclusions

We have investigated the effect of careful selection of meta-features on the quality of rankings generated by an IBL meta-learning approach for ranking. We considered a large set of general, statistical and information-theoretic meta-features, commonly used in meta-learning, and selected a subset, containing measures that represent properties of the data that affect algorithm performance. This selection has significantly improved the results. Although the average difference in accuracy between cross-validation and the best algorithm is only 2%, which makes the goal of improving the result of the latter very hard, our meta-learning

approach is able to reduce this difference to less than 1%. Although, this is a positive result, we plan to investigate how much improvement is achieved in the worst case, where the difference between CV and the best algorithm is 36%. We also plan to compare the subset of selected measures with new data characterization approaches, like landmarking.

Acknowledgments We thank the anonymous reviewers for useful comments. Thanks also to all the METAL partners for a fruitful working atmosphere, in particular to Johann Petrak for providing the scripts to obtain the meta-data. We also thank DaimlerChrysler and Guido Lindner for providing us the data characterization tool. Finally, we thank Rui Pereira for implementing part of the methods. The financial support from ESPRIT project METAL, project ECO under PRAXIS XXI, FEDER, Programa de Financiamento Plurianual de Unidades de I&D and from the Faculty of Economics is gratefully acknowledged.

References

1. Soares, C., Brazdil, P.: Zoomed ranking: Selection of classification algorithms based on relevant performance information. In Zighed, D., Komorowski, J., Zytkow, J., eds.: Proceedings of the Fourth European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD2000), Springer (2000) 126–135
2. Todorovski, L., Dzeroski, S.: Experiments in meta-level learning with ILP. In Rauch, J., Zytkow, J., eds.: Proceedings of the Third European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD99), Springer (1999) 98–106
3. Pfahringer, B., Bensusan, H., Giraud-Carrier, C.: Tell me who can learn you and i can tell you who you are: Landmarking various learning algorithms. In Langley, P., ed.: Proceedings of the Seventeenth International Conference on Machine Learning (ICML2000), Morgan Kaufmann (2000) 743–750
4. Brazdil, P., Gama, J., Henery, B.: Characterizing the applicability of classification algorithms using meta-level learning. In Bergadano, F., de Raedt, L., eds.: Proceedings of the European Conference on Machine Learning (ECML-94), Springer-Verlag (1994) 83–102
5. Brazdil, P., Soares, C.: A comparison of ranking methods for classification algorithm selection. In de Mántaras, R., Plaza, E., eds.: Machine Learning: Proceedings of the 11th European Conference on Machine Learning ECML2000, Springer (2000) 63–74
6. Bensusan, H., Giraud-Carrier, C., Kennedy, C.: A higher-order approach to meta-learning. In: Proceedings of the ILP'2000 (Work in Progress Track). (2000)
7. Todorovski, L., Brazdil, P., Soares, C.: Report on the experiments with feature selection in meta-level learning. In Brazdil, P., Jorge, A., eds.: Proceedings of the Data Mining, Decision Support, Meta-Learning and ILP Workshop at PKDD2000. (2000) 27–39
8. Henery, R.: Methods for comparison. In Michie, D., Spiegelhalter, D., Taylor, C., eds.: Machine Learning, Neural and Statistical Classification. Ellis Horwood (1994) 107–124
9. Blake, C., Keogh, E., Merz, C.: Repository of machine learning databases (1998) http:/www.ics.uci.edu/~mlearn/MLRepository.html.

Collaborative Data Mining and Data Exchange: A Case Study

Olga Štěpánková, Jiří Kléma, Petr Mikšovský

Department of Cybernetics, CTU Prague,
Technicka 2, 166 27 Prague 6, Czech Republic,
{step,klema,miksovsp}@labe.felk.cvut.cz

Abstract. The paper wraps up our experience gained during a collaborative data mining project solved using RAMSYS methodology. We pay special attention to some difficulties, which have appeared during the collaborative data mining and we try to identify their reasons. Finally, we raise several suggestions how to ensure efficient function of organizational memory necessary to support concise and transparent information exchange among all participating partners.

1 Introduction

The aim of data mining is to extract non-trivial and actionable knowledge from large or very large databases [10]. No one doubts that data mining is a very complex activity. Many different technologies have to be combined to accomplish its goal – experience is needed in handling large databases, in usage of various machine learning techniques, in statistics, etc. An institution handling a data mining project has to establish a team consisting of several specialists, usually. If the institution is a virtual one [6], i.e. its members do not share the location of their office but their relation is expressed in more subtle or indirect way, it can easily happen, that the members of the team are distributed on very distant places all over the world and connected by Internet, only. Is it possible to achieve useful collaboration on a data mining project even under these conditions?

This question is crucial for the SolEuNet project [11], which aims to develop virtual enterprise producing data mining and decision support services. The SolEuNet project partners belong both to academic institutions and to companies across Europe. Each data mining process follows the CRISP-DM principles [4] dividing the DM process into six interrelated phases: Business understanding, Data understanding, Data preparation, Modeling, Evaluation and Deployment. Moreover, efficient cooperation among the partners has to be ensured. To minimize management and coordination efforts RAMSYS methodology [7] has been designed. It is based on the following six principles: *"Light management, Start and stop at any time, Problem solving freedom, Knowledge sharing* and *Security.*" In SolEuNet, the groupware system called ZENO [9] was used to implement a tool supporting RAMSYS methodology. This paper wraps up experience gained when using this tool for the Spa DM project.

2 SPA Data Mining Project: Resource Allocation in Spa Facilities

A company running a spa facility offers a rich set of health procedures to heal medical problems of the patients who are arriving into the health farm for a restricted period. Obviously, each patient is supposed to obtain an individual treatment, i.e. a set of procedures assigned to him/her by the spa physician, who bases his recommendation on results of careful inspection of the patient upon his arrival. But the written recommendation of the spa physician is not enough to ensure that the patient really gets exactly the recommended procedures. The second necessary condition is that necessary resources (personnel/technical equipment) are available in appropriate quantity.

All over the fact that the groups of patients occupying the spa are changing frequently, the spa aims to be able to ensure the appropriate individual treatment for each of its patients. How can such a goal be achieved? It is vital for the spa administration to know in advance the total requirements of a group of patients for all individual procedures offered by the spa. That is why the company running the spa administrative system decided to start a data mining project (SPA Project) to be solved within the SolEuNet Project using the available history data as the basic data source.

2.1 SPA Data Mining Task and CRISP Phases

The SPA project happened to be one of the first collaborative DM projects where RAMSYS methodology could be verified. Four teams (CTU, KUL, BRI and LIACC) took part in this exercise – the first three started almost simultaneously, while LIACC joined 6 months later. These teams used 2 basic communication channels: e-mail and ZENO. ZENO played a role of organizational memory – place, where intermediate products of CRISP phases achieved by the partner teams were made public to be shared with all the participating partners. Some of these results were used later by another participating team – in this way collaborative DM really took place. This happened e.g. in the case of **data preprocessing**: its significant part was ensured by the CTU team using the data preprocessing tool SumatraTT [2].

In the **modeling phase**, all the participating teams decided to approach a prediction goal on their own using different ML tools. The intermediate results have been published on ZENO and this on-line information source highly supported competition among the partners. Two modeling directions have been considered in the project:

- *The individual centered approach* starts by predicting all procedures to be prescribed to a single patient. The total for one week is obtained as a sum of predictions for individual patients actually present.
- *The aggregated approach* tries to predict usage of resources for a whole group of patients at once.

Table 1 provides condensed **evaluation** of the results obtained by the participating teams. Only after a careful inspection we have found out that objectivity of the result comparison is hindered by the fact that the DM goal has been slightly modified or

improved by some of the teams. Modification of a DM goal is understood as a natural part of the CRISP methodology, which counts with loops. Any DM process has to be understood as pursuit of a moving target – to be successful one has to modify his/her goal according to the obtained results. At the present state of the RAMSYS implementation, any information concerning the goals considered by different participating teams had to be mined from the text reports provided by these teams with significant efforts. To overcome this problem we suggest simple refinement of RAMSYS structure in the next section 2.2.

Let us illustrate the upper mentioned claim on the SPA example: at early stages of business understanding phase it was not discussed how far in advance the prediction has to be generated. Everyone could thus assume that the prediction for the week no. W has to be available just at the beginning of that week. This timing gives a chance to use information contained in the data collected during the previous week ($W - 1$) to improve the prediction accuracy for the week W. Later on it was revealed by the domain expert that the prediction should be available at least four weeks in advance. The team, which joined the problem solving party as the last one, omitted this information due to the fact it was buried deep among other less important details. Thus they have been solving a modified goal without making this explicit - this finally caused certain incompatibility of the results obtained by different teams.

Table 1. The results reached by the individual teams. The individual cells show the total number of procedures (max. 35) that satisfy the specified condition. The first column shows how often the result exceeds the customer's required relative error (RE). The second column shows number of procedures for which the given team delivered the best result. The last column shows for how many times the given team was "close" to the best one.

Team	RE>20%	Best	RE-RE(best)<5%
BRI	5	2	18
KUL	12	1	14
LIACC	2	29	33
CTU	9	3	19

2.2 RAMSYS Principles and Organizational Memory

The main aim of the six RAMSYS principles (*Light management, Start any time, Stop at any time, Problem solving freedom, Knowledge sharing* and *Security*) is to ensure cooperative environment supporting competition and creativity of partners while not restricting their academic freedom. How did we succeed to follow these principles during the SPA experiment? Most of the RAMSYS principles caused no problems. The only exception is the **knowledge sharing** principle. It proved to be the most difficult part in the considered project. All the partners did their best to provide all necessary information, everybody made available his/her results (e.g. ideas, evaluation results) as well as the modified, extended or transformed datasets on ZENO. But sometimes the provided description happened to be difficult to follow and the rest of partners preferred not to rely on the results of others but did most of the work on their own. We believe that one of the reasons is lack of tools and standards

supporting this aspect of DM process. The organizational memory has to have transparent structure, which ensures direct access to knowledge characterizing the considered task. The core knowledge to be shared has to specify precise description of:

1. all the considered DM goals (new goals are being suggested often during the course of the project, the domain expert can assign them with his priorities etc.),
2. scripts designed for preprocessing the treated data.

The first item can be handled easily if the structure of ZENO for RAMSYS is extended by a direct access to a place to present all the considered DM goals. This **Review of DM goals** could be situated e.g. in Business understanding section. Each goal has to be presented in a clear structured way including a unique name, time the goal was defined, set of its considered input attributes (and used preprocessing), evaluation of the goal importance given by the domain expert, as well as the list of all generated models. Appropriate extensions of PMML could be applied for that purpose. The next section is devoted to problems of knowledge sharing in the data preprocessing tasks.

3 Centralized Support and RAMSYS

Collaborative data mining has to be situated in distributed problem solving environment supporting sharing and re-use of resource-intensive results. Processes involved in data transformation and model evaluation certainly belong among resource-intensive activities. If the collaborating subjects want to share knowledge it has to be expressed in a form understandable to all of them. It is very important to ensure standardization of knowledge to be exchanged. While PMML becomes a standard data mining model representation, the standard description of data pre-processing tasks is still missing. This is a serious drawback: if we want to support *start-any-time* and *stop-any-time* RAMSYS principles we necessarily have to be able to reconstruct problem-solving path of any of collaborating subjects.

3.1 Data Transformation Standard

Knowledge sharing would be significantly improved if all data transformations applied in a certain DM or DSS problem are described in a single format shared by all partners. Let us call it **Data Transformation Mark-up Language** (DTrML). Its obvious advantage is transparent description of all data modifications and transformations applied when solving the given task. Moreover, as soon as DTrML becomes operational, we get a powerful universal data pre-processing tool, which can ensure the centralized data transformation phase of collaborative DM (see Fig. 2B). No doubt, such a centralization results in efficient use of available resources (time and computational capacity).

We believe that design of SumatraTT [2], a data pre-processing tool developed at CTU, contains the basic components of DTrML including the metadata concept for data transformation description [1]. In this way, it integrates descriptive metadata

with the operational ones. The metadata is stored in XML format (similarly as PMML), which can help in the case of conversion into another format. Moreover, the resulting kind of standardization brings, similarly as Java does, self-documenting effect which simplifies human understanding. To support the last claim let us describe the Sumatra-based data transformation. There are input and output data sources the structure and location of which is described in metadata. Then a certain sequence of transformation templates is applied. Every transformation template consists of a documentation describing what input/output is expected and how to set-up parameters – it is not usable without such documentation.

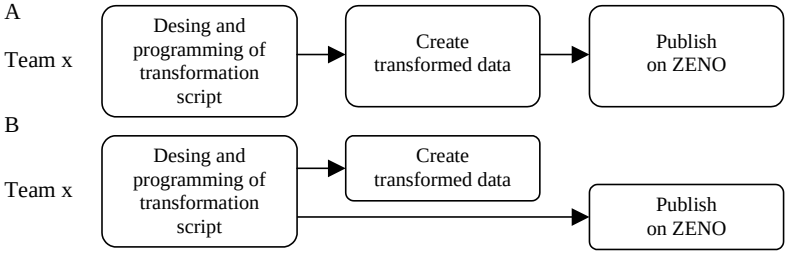


Fig. 2. Two options for exchange of data transformation results. The case A represents situation where each collaborating subject prepares data by a different data transformation tool. It means that they can share/publish transformed data sets only. Whereas in the case B the standardized data transformation script can be shared/published. It is not only more operational but besides others it saves disk space on ZENO.

3.2 Centralized Model Evaluation

The SPA experiment has been reviewed in [3] from the point of view of model evaluation, which authors divide into 5 sequential steps: tune, build, predict, evaluate and publish. The authors have noted number of problems: "*... the evaluation criterion may change, people tent to report the measures that their tools compute but are reluctant to implement measures themselves ...*" and they present centralized model evaluation as the remedy for these problems. The centralized model evaluation can appear in 4 options depending on the specific point in the tune ... publish sequence when the centralized approach is started. In the option 1, the only centralized step is the last one "publish" – this evaluation mode was applied in the SPA experiment and it proved to cause some misunderstandings among the cooperating subjects. That is why more complex options seem preferable. The authors conclude "*In the longer run, under assumption that PMML is general enough to describe any kind of model that could be submitted and that interpreters are available, it seems desirable to shift to Option 3 (which ensures the prediction on test data centrally)*." On the first glance, this mode seems simple: the collaborating subjects send their working solution in the form of the PMML description to a single node which ensures centrally their use for prediction, evaluation and finally publish the results. This sounds easy until we realize that most often the models created as the result of DM do not rely directly on the original data. They use data, which are pre-processed, aggregated, complemented by new derived attributes, transformed to fit the requirements of the model and of the

DM task. Consequently, the working solution must include DTrML part beside the PMML parts. That is why development of DTrML has to be considered as an integral part of the plan to incorporate centralized model evaluation process into RAMSYS.

4 Conclusions

Our experience in the SPA project pointed to the problems of knowledge sharing in all steps of the collaborative DM process. Existence of tools and standards, which support organizational memory seem to be a prerequisite for reaching the expected effects of collaborative data mining and efficient collaboration. We are suggesting additional tools and standards which range from a simple knowledge structure designed in the Section 2.2 up to very ambitious goal to standardize language used for description of data transformations suggested in 3. Introduction of DTrML can have significant impact on re-use of the obtained results as it simplifies creation of a universal functional repository of solved cases containing solutions which can be reconstructed or re-applied on new data at any time. The suggested standardization will simplify transfer of DM results into the practice.

5 References

1. Aubrecht, P., Kouba, Z.: Meta-Data Driven Data Transformation. Proc. of the 5th World Muti-conference on Systemics, Cybernetics and Informatics, 2001.
2. Aubrecht, P., Zelezny, F., Miksovsky, P., Stepankova, O.: SumatraTT: Towards a Universal Data Preprocessor. Proc. of the 16th European Meeting on Cybernetics and Systems Research – Vienna 2002, pp. 818-823, Austrian Society for Cybernetic Study.
3. Blockeel, H., Moyle, S.: Collaborative data mining needs centralized model evaluation, submitted to DMLL-2002 at ICML-2002
4. Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C. and Wirth, R.: CRISP-DM 1.0: Step-by-step data mining guide. CRISP-DM consortium, 2000.
5. Jorge, A., Moyle, S., Richter G., Voß, A.: Remote Collaborative Data Mining Through Online Knowledge Sharing, submitted to PROVE-2002.
6. Lavrac, N., Urbancic, T., Orel, A.: Virtual Enterprise For Data Mining And Decision Support: A Model For Networking Academia And Business, submitted to PROVE-2002.
7. Moyle, S., Jorge, A.: RAMSYS – A Methodology for Supporting Rapid Remote Collaborative Data Mining Projects, IDDM'01, ECML/PKDD Workshop notes, 2001.
8. Stepankova, O., Lauryn, S., Aubrecht, P., Klema, J., Miksovsky, P., Novakova, L., Palous, J.: Data Mining for Resource Allocation: A Case Study. In: Intelligent Methods for Quality Improvement in Practice, Prague, CTU FEE, Department of Cybernetics, pp. 94-105, 2002.
9. Voss, A., Gartner, T., Moyle, S.: Zeno for Rapid Collaboration in Data Mining Projects. Proceedings of the ECML/PKDD Workshop on Integration of Data Mining, Decision Support and Meta-Learning (pp. 43-51). ECML/PKDD'01 Workshop notes, 2001.
10. Witten, I., Frank, E.: Data Mining - Practical Machine Learning Tools and Techniques with Java Implementations. Morgan Kaufmann, 1999.
11. Data Mining and Decision Support for Business Competitiveness: A European Virtual Enterprise, SolEuNet pages available at http://soleunet.ijs.si/.

Qualitative Clustering of Short Time-Series: A Case Study of Firms Reputation Data

Ljupčo Todorovski[1], Bojan Cestnik[2], Mihael Kline[3], Nada Lavrač[1], and Sašo Džeroski[1]

[1] Department of Intelligent Systems, Jožef Stefan Institute
Jamova 39, SI-1000 Ljubljana, Slovenia
{Ljupco.Todorovski, Nada.Lavrac, Saso.Dzeroski}@ijs.si
[2] Temida, Grassellijeva 20, SI-1000 Ljubljana, Slovenia
Bojan.Cestnik@temida.si
[3] University of Ljubljana, Faculty of Social Sciences,
Kardeljeva pl. 5, SI-1000 Ljubljana, Slovenia
Mihael.Kline@uni-lj.si

Abstract. In this paper, we propose a clustering approach to the analysis of time series data about reputation of firms. A standard hierarchical clustering method is used and a new measure of distance between time series is proposed. The newly introduced measure is based on qualitative analysis of time series data. The approach is evaluated on a task of clustering Slovenian firms according to the pattern of change of their reputation through years.

1 Introduction

Undoubtedly, reputation of an firm is a dynamical concept. It is usually measured once per year, but most of the analysis methods explore the time local relation of reputation to various financial and performance indicators of the firm. However, time local analysis gives no insight into dynamic change of reputation through years. The temporal change of reputation indicators is usually analyzed with different regression tools [4].

In this paper, we aim to analyze time series data about the dynamic change of the reputation of firms through years. The presented approach to analysis of reputation data is based on a clustering methodology. Following the clustering methodology, groups of firms with similar patterns of temporal change of their reputation are created. Thus, the key concept in clustering is the measure of distance between firms, or more precisely distance measure between time series that reflects temporal changes of reputation. Most commonly used distance measures for time series analysis are based on the correlation coefficient [7, 8]. However, the correlation coefficient is very problematic in cases when we are dealing with very short time series, measured at ten or less time points. To address this problem, we propose the use of an alternative measure of qualitative distance between time series.

We evaluate the performance of the proposed approach on the task of clustering Slovenian firms based on their reputation. The dataset contains data about 113 firms in Slovenia from 1996 to 2000. The data have been collected using the standard Computer Assisted Telephone Interviewing (CATI) method. A quota sample of 760 to 840 Slovenian managers are interviewed about their subjective perception of the reputation of firms. The dataset also includes data about yearly financial performance and advertising investments from the FIPO [1] and IBO [2] databases.

The paper is organized as follows. The hierarchical clustering methodology is presented in Section 2. Section 3 introduces the measure of qualitative distance between short time series and provides an illustrative example of its advantage over a correlation based distance measure. The results of clustering time series data about the reputation of Slovenian firms are presented and analyzed in Section 4. Section 5 concludes the paper with a brief summary and directions for further work.

2 Hierarchical clustering

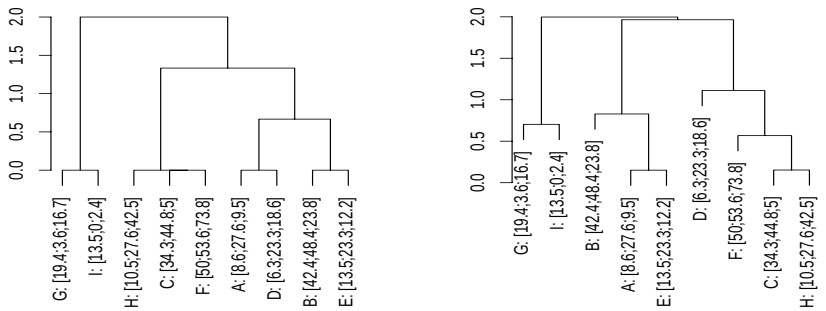
Clustering is an unsupervised learning method. Given data about a set of objects, a clustering algorithm creates groups of objects following two criteria. First, objects are close (or similar) to the other objects from the same group (internal cohesion) and distant (or dissimilar) from objects in the other groups (external isolation).

A particular class of clustering methods, studied and widely used in statistical data analysis [9, 5] are hierarchical clustering methods. The hierarchical clustering algorithm starts with assigning each object to its own cluster, and iteratively joins together the two closest (most similar) clusters together. The distances between objects are provided as input to the clustering algorithm. The iteration continues until all objects are clustered into a single cluster. The output of a hierarchical clustering algorithm is a hierarchical tree or dendrogram.

Two examples of dendrograms obtained by clustering 9 objects are presented in Figure 1. Dendrogram is a binary tree where the initial clusters, consisting of one element only, form the leaves of the tree. Each internal node represents a cluster that is formed by joining together objects from the two clusters corresponding to the children nodes. The height of the node is proportional to the distance between the joined clusters. For example, clusters $\{G\}$ and $\{I\}$ are joined together at height 0 in the dendrogram on the right-hand side of Figure 1, and at height 1 in the dendrogram on the right-hand side of Figure 1.

In the very last step of the clustering, a number of clusters are obtained from the dendrogram. This is done by cutting the dendrogram at a given height. Cutting a single dendrogram at different heights produces different numbers of clusters. For example, cutting the dendrogram on the left-hand side of Figure 1 at height 1.5 produces the following 3 clusters: $\{G, I\}$, $\{H, C, F\}$ and $\{A, D, B, E\}$. Analogously, cutting the dendrogram on the right-hand side of Figure 1 at height 1.4 produces the following 3 clusters: $\{G, I\}$, $\{B, A, E\}$ and $\{D, F, C, H\}$.

Fig. 1. Two example dendrograms obtained with clustering 9 time series of length 3 using a qualitative distance measure (left-hand side) and a correlation based distance measure (right-hand side).



The optimal "cut point" that produces clusters with maximal internal cohesiveness and minimal external isolation is where the difference between the height of two successive nodes in the dendrogram is maximal. The dendrogram on the left-hand side of Figure 1 has three equally good cut points (first between 0 and 1, second between 1 and 2 and third between 2 and 3), whereas the one on the right-hand side has only one optimal cut point (any height between between 1.25 and 1.75).

3 Distance measures for short time series

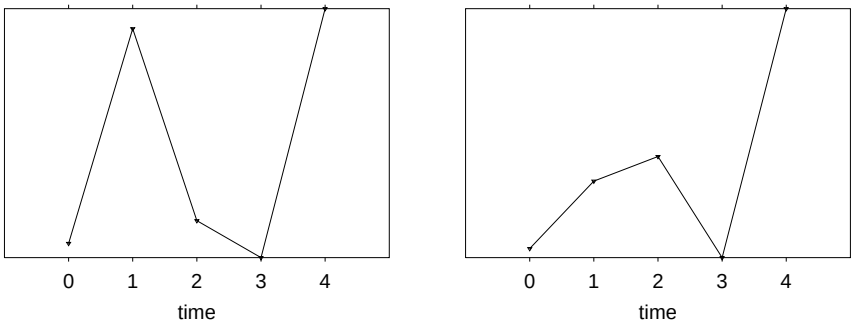
There is a very important question about clustering that has remained unanswered in Section 2: "how should we measure the distance between objects and between clusters of objects?" Following the fact that in this study we analyze time series data, we are interested in measure of distance between time series.

Most commonly used distance measures are the ones that define distance between two n-dimensional vectors of real numbers. Examples of these are Euclidean and Manhattan distances. These measures are not appropriate for clustering time series, because they mainly capture the difference in the scale and baseline (order of magnitude difference) between objects. Instead of that, in clustering time series, we are more interested in the shape of the time change of the value through time.

A better alternative is to use a correlation based distance measure. The correlation coefficient $r(X, Y)$ between two time series X and Y, calculated as

$$r(X,Y) = \frac{E[(X - E[X]) \cdot (Y - E[Y])]}{E[(X - E[X])^2] \cdot E[(Y - E[Y])^2]},$$

Fig. 2. Two example time series of length 5. The correlation based distance between them equals 0.694, whereas the qualitative distance equals 0.2.



where $E(V)$ is used to denote the expectation (i.e., mean value) of V, measures the degree of linear dependence between X and Y. Values of $r(X,Y)$ close to zero denote that there is a low degree of linear dependence between X and Y. On the other hand, values close to ± 1 denote a high degree of linear dependence. In terms of shapes of the X and Y, the value of $r(X,Y)$ has the following intuitive meaning. Values close to -1 means that X and Y have "mirrored" shapes, r close to 0 means that shapes are unrelated (and consequently dissimilar) and r close to 1 means that the shapes are very similar. Following this intuitive interpretation of correlation we can define the following distance measure between time series [8]:

$$D_r(X,Y) = \sqrt{2 \cdot (1 - r(X,Y))}.$$

However, this correlation based distance measure has two drawbacks. First, it is well known that the correlation coefficient is very poorly estimated when we have small number of observations (i.e., short time series). Second, it is capable of capturing only the linear aspect of dependence or relation between time series. Two time series that are non-linearly related to each other will be distant from each other, regardless of the similarity of their dynamic change through time.

The distance measure that we propose here is based on a qualitative analysis and comparison of the shape of the time-series. Consider the two time series X (left-hand) and Y (right-hand) of length five, presented in Figure 2. We choose a pair of time points i and j and we observe the qualitative change of the value of X and Y. Three possible values of qualitative change $q(X_i, X_j)$ (as well as $q(Y_i, Y_j)$) can be distinguished: increase, when $X_i > X_j$; no-change when $X_i = X_j$ and decrease $X_i < X_j$[1]. For example, consider the change between the first and third

[1] Note the strict definition of the no-change situation is used here for simplicity. In reality, we use a threshold ($|X_i - X_j| < \epsilon$) to test the equality of X_i and X_j. Alternatively, if more then one measurement for X_i is available, a statistical test of the significance of change can be applied for the same purpose.

time point in X and Y from Figure 2: they both increase. On the other hand, if we consider the change between the second and third point, we observe that while X decreases, Y increases. Now, we can calculate the qualitative difference between X and Y by summing up the differences for all the pairs of time points:

$$D_q(X,Y) = \frac{4}{N \cdot (N-1)} \cdot \sum_{i<j} \texttt{Diff}(q(X_i, X_j), q(Y_i, Y_j)),$$

where $q(V_i, V_j)$ is used to denote the qualitative change of V and `Diff` is a simple function that defines the difference between three possible values of qualitative change. The factor $\frac{4}{N \cdot (N-1)}$ is used to normalize the values of the distance measure in the range $[0, 2]$ which equals the range of values of the correlation based distance D_r.

Table 1. Definition of the `Diff` function.

		q_1	
$\texttt{Diff}(q_1, q_2)$	increase	no-change	decrease
increase	0	0.5	1
q_2 no-change	0.5	0	0.5
decrease	1	0.5	0

The `Diff` function is defined in Table 1. The definition simply specifies that the difference between increase and decrease values equals 1, whereas the difference between increase (or decrease) and no-change values equals 0.5.

Roughly speaking, D_q counts the number of disagreements of change of X and Y. It equals 0 if both time series increase and decrease at same time. In the qualitative reasoning methodology QSIM [6] this is denoted by using the qualitative constraint $M^+(X,Y)$. The maximal distance of 2 is obtained in cases when X decreases wherever Y increases. The QSIM notation for this situation is $M^-(X,Y)$.

The proposed qualitative distance measure does not have the drawbacks of the correlation based measure, mentioned above. First, it can be calculated on very short time series, without decreasing the quality of the estimate. On the other hand, calculating D_q for pairs of very long time series can be impractical for time complexity reasons.[2] Second, it captures the similarity between patterns of change of the time series, regardless of whether the nature of the dependence between them is linear or non-linear. An illustration of the advantage of using the qualitative distance measure is given in Figure 2. Although the pattern on the left-hand side is very similar to the pattern on the right-hand side, the distance between them is rather high according to the correlation based measure (0.694). It is three times higher than the one measured by the qualitative distance measure (0.231).

[2] Note that the time complexity of calculation of D_q is quadratic in the length of the time series, due to the fact that all possible pairs of time points are considered.

4 Clustering of Firms Reputation Data

We applied the presented approach to the task of clustering firms based on their reputation. In this section, we first describe the dataset and the methodology used for measuring the reputation and collecting other data about Slovenian firms. We then present the results of clustering and analyze the obtained clusters.

4.1 Experimental Methodology

The dataset contains data about 113 firms in Slovenia covering the period 1996 to 2000. For each firm, the dataset includes measurements of three groups of descriptors. The first group includes time-invariant general data such as the name of the firm and (industrial or service) area of the firm activity. The second group of features include estimates of the recognition and reputation of the firms. These estimates are based on the customers' subjective perception of the reputation. The data about this group of features have been collected using the standard Computer Assisted Telephone Interviewing (CATI) method. Further details about the CATI methodology are given below. Finally, the third group of features includes yearly financial performance indicators, such as the firm's equity, revenues, net profit and number of employees, as well as advertising investments data. The data about this group of features have been collected from the publicly available databases FIPO [1] and IBO [2]. The features in this group are measured once per year for the period 1996 to 1999.

Since 1995, Kline and Kline marketing agency has conducted annual surveys that assess corporate reputation of 255 largest companies in Slovenia. The standard CATI methodology is used to colect answers from quota sample of 760 to 840 Slovenian managers. The corresponding data are stored in the computer program QA that was designed to facilitate the questioning process. After the data gathering process is completed, the program exports the collected data in a standardized format, so that it can be imported to other statistical or data mining software for further processing.

One of the most important features of the QA program is to lower the complexity of estimating the image of firms. Note that the idea of asking each individual to evaluate all of the firms would surely degrade his or her initiative to cooperate. Therefore, a careful experimental design is required before one can start questioning people. From our experience, as well as from theoretical psychology, we concluded that in order to obtain valuable information, each individual could subjectively evaluate at most 15 firms. As s result, QA program in each run randomly selects a permutation of 15 firms, balancing the overall frequency rate for each firm.

We performed two clustering experiments. The first is on a dataset for the period of five years from 1996 to 2000 that contains complete data for 82 firms. Since many fields describing the years 1996 and 1997 were missing, we repeated a simplified version of the first experiment on a dataset for the period of three years from 1998 to 2000 that contains complete data for 106 firms. In both experiments, hierarchical clustering with the qualitative distance measure was

used. For the purpose of comparison of results, we also used a correlation based distance measure.

4.2 Experimental Results

In both experiments, the presented approach generated 4 clusters of firms. The summary of the clusters for both experiments is presented in Table 2.

Table 2. Summary of the clusters generated on both experimental datasets.

first dataset	cluster 1	cluster 2	cluster 3	cluster 4	full dataset
size	10	29	34	33	106
distribution	9.43%	27.36%	32.07%	31.14%	100%
reputation	15.32	31.67	24.11	34.32	28.53
employees	690.12	657.33	912.08	1,192.05	911.78
equity	3,221,372.33	6,923,234.12	8,310,799.32	14,835,049.93	9,686,419.34
assets	8,728,310.46	11,743,432.20	13,426,307.40	22,726,639.07	15,727,797.17
equity on assets	42.68	56.24	61.96	66.68	60.04
revenues	9,758,372.67	13,750,380.12	12,131,758.02	25,038,866.90	16,877,261.45
net profit	1,683,917.50	378,695.03	436,917.19	1,049,282.17	759,206.09
net profit on rev.	145.88	3.89	4.46	5.86	17.32
advertising	8,329,958.17	107,488,647.38	31,849,193.61	92,239,627.24	69,261,171.31

second dataset	cluster 1	cluster 2	cluster 3	cluster 4	full dataset
size	13	30	25	14	82
distribution	15.85%	36.58%	30.49%	17.08%	100%
reputation	25.72	33.94	36.62	23.68	31.70
employees	747.64	1,081.46	972.47	1,320.76	1,032.58
equity	5,926,813.87	12,507,134.41	12,387,730.71	11,200,748.06	11,202,319.84
assets	11,828,923.13	20,833,307.33	16,633,921.88	18,131,153.82	17,799,067.92
equity on assets	53.31	58.67	73.87	64.04	62.86
revenues	10,466,048.83	21,506,881.25	23,223,478.82	14,270,215.67	18,954,032.25
net profit	575,692.47	881,447.59	819,185.20	470,787.58	744,411.52
net profit in rev.	4.20	4.99	6.78	3.52	5.09
advertising	80,785,072.90	121,001,002.85	71,855,421.89	32,405,566.90	84,672,149.32

The first two rows in each table are the sizes of clusters (the number of firms in each cluster) and the distribution of firms among clusters. Further down the rows in both tables, the average values of reputation and some financial and performance indicators are presented.

4.3 Preliminary Analysis of the Results

The first observation is that a small number of clusters is obtained in both cases. This compares favorably with the number of clusters generated with the

correlation based distance measure. The latter generated 16 clusters for the first dataset and 8 clusters for the second dataset.

The four clusters obtained are immediately recognizable in both cases. Of course, it is easier to recognize the patterns for the second dataset with shorter time series. In the first dataset, the patterns become more complex, but still they are compact enough to still be recognizable. In both cases, we can identify the following four important and relevant groups of firms. The first group, metaphorically named gazelles, includes firms with rapidly increasing growth of reputation. The firms in the second group of mugwumps have stable, but slower growth of reputation that is on average higher then the reputation of firms in the fist group. The third group of decays includes mostly firms with constant decrease of the reputation. The fourth group includes lingerers, i.e., firms with constant ups and downs. Of course, in each of the described groups there are exceptions from the general pattern. The analysis of exceptions is more complex and requires a separate analysis of data for each exceptional firm.

We also analyzed the distribution of values of other financial and performance indicators among clusters. From the table, we observe that the firms in the fourth cluster obtained in the first dataset (the one of the mugwumps) are highly reputable (with average reputation of 34.32 versus the overall reputation of 28.53), but also have higher average equity (14M vs. 9M), assets (22M vs. 5M) and equity on assets (66.68 vs. 60.04). It is interesting that the firms in this cluster are not the ones with the highest average advertising budget. The high advertising budget is typical for the firms in the second cluster of gazelles.

Please note that the observations presented here are preliminary. For more conclusive statements, a test of statistical significance of differences should be performed.

5 Conclusions and Further Work

In this paper, we present a clustering approach to analysis of time series data about the dynamic change of the reputation of firms through years. We use hierarchical clustering to obtain groups of firms with similar patterns of change of reputation through years. We proposed a novel measure of distance between time series that is especially suitable for very short time series, where the use of commonly used correlation based distance measure is not appropriate.

The preliminary analysis of the obtained clusters shows the usability of the approach. The number of clusters is small and they reflect representative groups of Slovenian firms. This is in contrast to clustering with a correlation based distance measure where many clusters were generated.

However, the analysis of the discussion of the results presented here is preliminary. Many aspects of the analysis should be improved. First, we observe considerable amount of variance of the values of different firms' features among clusters. The statistical significance of this variance should be tested. Also, the reasons for this variance should be further explored and explained.

Furthermore, the approach presented here can be a first step towards more extensive analysis of dynamic change of the reputation of firms. One line of extensions would be towards practical applications by building predictive models of reputation, that can be widely used by managers and analysts for prediction and planing. Another line of further work is to analyze the "buffer" hypothesis about the delay between reputation change and the consequential, delayed, change of other financial and performance indicators of the firm. For this purpose, the delay operator between time series should be taken into account in the further development of the methodology.

Finally, clustering methodology has already been used for analysis of reputation of German Firms in [3]. They use alternative approach, where each year firms are clustered according to their level of reputation. Then the dynamic change of cluster membership can be observed in order to analyze the dynamic change of reputation through years. One of the goal of further work is to compare this approach to the clustering time series approach presented in this paper.

Acknowledgments

The work reported in this paper was supported by the Slovenian Ministry of Education, Science and Sport, and the IST-1999-11495 project Data Mining and Decision Support for Business Competitiveness: A European Virtual Enterprise.

References

1. FIPO Database. Gospodarski Vestnik, Ljubljna, Slovenia, 2000. `http://www.gvin.com/FIPO/`.
2. IBO Database. Media Research Institute Mediana, Ljubljna, Slovenia, 2000. `http://www.mediana-irm.si/eng/02_02.html`.
3. R. L. M. Dunbar and J. Schwalbach. Corporate reputation and performance in Germany. *Corporate Reputation Review*, 3(2):115–123, 2000.
4. S. A. Hammond and Jr. J.W. Slocum. The impact of prior firm financial performance on subsequent corporate reputatation. *Journal of Business Ethics*, 15:159–165, 1996.
5. J. A. Hartigan. *Clustering Algorithms*. Wiley, New York, 1975.
6. B. Kuipers. Qualitative simulation. *Artificial Intelligence*, 29(3):289–338, 1986.
7. R. N. Mantegna. Hierarchical structure in financial markets. *The European Physical Journal B*, 11(1):193–198, 1999.
8. P. Ormerod and C. Mounfield. Localised structure in the temporal evolution of asset prices. In *New Approaches in Financial Economic Conference*, Santa Fe, NM, 2000. Available for download from `http://www.quantnotes.com/publications/volterra/sf21092000.pdf`.
9. R. R. Sokal and P. H. A. Sneath. *Principles of Numerical Taxonomy*. Freeman, San Francisco, 1963.

A KDDSE-independent PMML Visualizer

Dietrich Wettschereck[1]

University of Applied Sciences, Bonn-Rhein-Sieg, Grantham Allee 20, 53757 Sankt Augustin, Germany,
dietrich.wettschereck@fh-bonn-rhein-sieg.de

Abstract. Several knowledge discovery support engines (KDDSE) feature the export and in a few cases even the import of data mining models in the Predictive Modeling Markup Language (PMML) standard. A visualization tool for PMML models that is independent of a specific KDDSE is presented in this paper. An extension of the PMML model for association rules that allows the definition of propositional and first order rules is also presented in its document type description form (DTD).

1 Introduction

The emerging standard for the platform and system independent representation of data mining models PMML (*Predictive Markup Modeling Language* [1]) is currently supported by a number of commercial and non-commercial knowledge discovery support engines (KDDSE). Most of these systems can export one or several model types in PMML, some can even import models generated by other KDDSEs. The primary purpose of the PMML standard is to separate model generation from model storage in order to enable users to view, post-process, and utilize data mining models independently of the KDDSE that generated the model.

This paper makes two contributions that are only related through the fact that they both employ PMML: it proposes an extension to the PMML model for association rules (AssociationModel) for the definition of (first-order) classification and regression rules (Section 2) and it describes a KDDSE-independent PMML visualizer (Section 3). A short discussion of the utility of such a visualizer in Section 4 is followed by a description of planed extensions to the tool (Section 5).

2 PMML DTD for (first order) rules and subgroups

This section describes a DTD (document type description) for propositional and first order rules. Subgroups [8] can also be represented by this PMML model type. The proposed DTD closely follows the AssociationModel DTD that is part of the PMML standard.

The rule models in PMML allow for defining either a classification or prediction structure. Each Rule holds a logical predicate expression that defines the conditions under which a rule will fire and a similar expression for the conclusions that can be drawn from the rule.

```
<!ELEMENT RuleModel (Extension*, MiningSchema,
                ModelStats?, GeneralRuleItem*,
                Itemset*, Rule+,
                Extension*)>
```

The *RuleModel* element starts the definition of a rule model. It has a few optional slots (denoted by '*' and '?' and one required slot, the element *Rule*. *Extension*, *MiningSchema*, and *ModelStats* are standard PMML. *GeneralRuleItem*, *Itemset*, and *Rule* are extensions that are described in detail below. The attributes of the *RuleModel* are not shown as they are identical to the attributes of the *AssociationModel*.

The *Rule* element is an encapsulation for a propositional or a first order rule. Every rule contains an antecedent and a consequent. The antecedent is either a simple predicate, a compound predicate or a reference to an *Itemset*. A compound predicate combines simple predicates and *Itemsets*. An *Itemset* is a generalization of an association rule *Itemset*. It represents a literal in first order logic terminology. The consequent is either a simple predicate or an *Itemset*.

The element *GeneralRuleItem* is an element of an *Itemset* and denotes a generalization of *Item* as defined in *AssociationModel* or a *RuleItem* as defined in this model:

```
<!ELEMENT GeneralRuleItem (Extension*, (RuleItem|Item))>
<!ATTLIST GeneralRuleItem EMPTY>
```

RuleItems are contained in *Itemsets* and represent a field (variable).

```
<!ELEMENT RuleItem EMPTY>
<!ATTLIST RuleItem
     id                %ELEMENT-ID;      #REQUIRED
     field             CDATA             #REQUIRED
     mappedValue       CDATA             #IMPLIED   >
```

Attribute description:

id: An identification to uniquely identify an item.

field: This must point to a field that was previously defined in the *DataDictionary*

mappedValue: Optional, a value to which the internal field value is mapped. This should be kept empty, since this information is redundant to the information given in the *DataDictionary* (it is only included here for compatibility with *Item* in *AssociationModel*).

Itemsets are contained in compound predicates of rules or directly in the antecedent or consequent. They are a generalization of *AssociationModel Itemsets*:

```
<!ELEMENT Itemset (Extension*, ItemRef+, DisplayTerm?)>
<!ATTLIST Itemset
        id                %ELEMENT-ID;     #REQUIRED
        predicate         CDATA            #REQUIRED
        support           %PROB-NUMBER;    #IMPLIED
        numberOfItems     %INT-NUMBER;     #IMPLIED   >
```

Attribute description:

id: An identification to uniquely identify an *Itemset*

predicate: the name of the predicate that will be used to combine the arguments. This can be a simple predicate (equals, greaterThan, ...), but shouldn't, since a *SimplePredicate* is better in this case. The predicate is the name of a functor and the items are its arguments. Example: predicate="father_of" ... itemRef="1" .. itemRef="2" indicates that the person indicated by item #1 is the father of the person indicated by item #2.

support: The relative support of the *Itemset*

numberOfItems: The number of items contained in this *Itemset*

DisplayTerm: The *Itemset* (Literal)described in natural language. Placeholders within the *DisplayTerm* allow for insertion of actual values.

```
<!ELEMENT DisplayTerm EMPTY >
<!ATTLIST DisplayTerm value CDATA  #REQUIRED>
```

Attribute description:

value: The ItemSet described in natural language. A visualization of this model should use this term instead of the standard predicate(arg1, arg2, ..., argn) representation. Placeholders in this value are denoted by %0, %1, ... where %0 is replaced by the actual value of the TermRef at position one, %1 by the TermRef at position two, and so on. The order of the placeholders is arbitrary, and not all TermRefs must be listed.

Each *Rule* consists of:

```
<!ELEMENT Rule ( Extension*, ScoreDistribution*,
                 Antecedent, Consequent )>
<!ATTLIST Rule
     support          %PROB-NUMBER;    #REQUIRED
     confidence       %PROB-NUMBER;    #REQUIRED
     ruleId           CDATA     #IMPLIED         >
```

Attribute description:

support: The relative support of the rule

confidence: The confidence of the rule

ruleId: The id value of the rule

The standard PMML definition of *PREDICATE* is extended here by *ItemSetRef* indicating that a predicate can also be a more complicated operator as foreseen by the standard:

```
<!ENTITY % PREDICATE "( SimplePredicate | SimpleSetPredicate |
                        CompoundPredicate | ItemSetRef |
                        True | False  ) " >
```

Each *Antecedent* consists of:

```
<!ELEMENT Antecedent ( Extension*, (%PREDICATE;) )>
```

The antecedant has an empty attribute list. The definition of a *Consequent* is identical to the definition of an antecedent.

3 Visualization of PMML models

Data visualization methods have been part of statistics and data analysis research for many years. This research concentrated primarily on plotting one or more independent variables against a dependent variable in support of explorative data analysis [4, 6]. The visualization of analysis results, however, only recently gained some attention with the proliferation of data mining[2]. This recent interest was spawned by the often overwhelming number and complexity of data mining results.

The visualization of analysis results primarily serves four purposes: (1) to better illustrate the model to the end user, (2) to utilize comparison of models, (3) to increase model acceptance, and (4) to enable model editing and provide support for ”what-if questions”.

The tool presented in this section was designed by the author to address these four issues. It is a Java implementation that can be run as an application or as an Applet. [1]

[1] The software is available upon request from the author. See also: http://soleunet.ijs.si/website/other/pmml.html.

It allows for viewing of models by users that either do not have access to the actual KDDSE, want to avoid the overhead of starting a KDDSE or want to present their results in the internet. This visualization wizzard currently supports the following PMML models: decision and regression trees (Figure 1), association rules (Figure 2), propositional and first order rules (non-standard PMML, Figure 3), and subgroups (non-standard PMML, Figure 4).

Figure 1 shows a decision tree for the well known Iris domain. The tree is fully expanded and is normally shown in color, where different colors in the nodes denote the number of instances from each class contained in that node. The user can browse through the tree and open or close subtrees as needed.

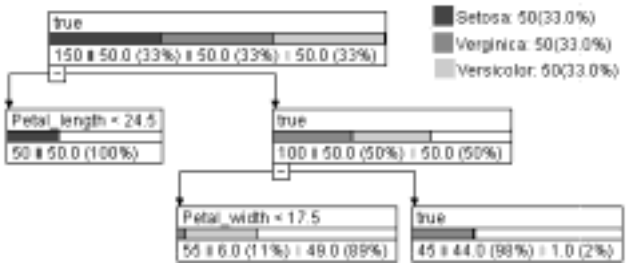


Fig. 1. Visualization of a decision tree for the Iris data set (courtesy of G. Meyer, IBM, visualized from PMML model exported from Intelligent Miner)

Figure 2 shows an interactive visualization for association rules. For each rule, confidence and support are displayed. The bar below each rule display graphically these two numbers where the length of the bar shows the support and the color of the bar its confidence (from red denoting low confidence to green denoting high confidence). The two sliders at the bottom of the display allow the restriction of the rules to be displayed to those that satisfy selected minimal confidence and support values.

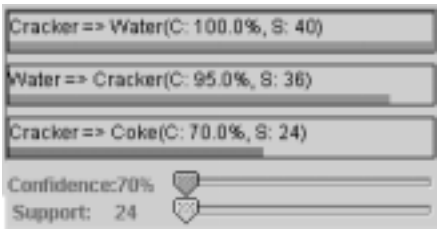


Fig. 2. Visualization of three association rules (courtesy DMG, slightly modified example for AssociationModel).

Figure 3 displays a modified set of rules learned by Aleph in the animal domain. The rules for each class are summarized by the bars at the right of the figure. The left bar shows the number of instances correctly covered, and the right bar the number of exceptions covered by the rule.

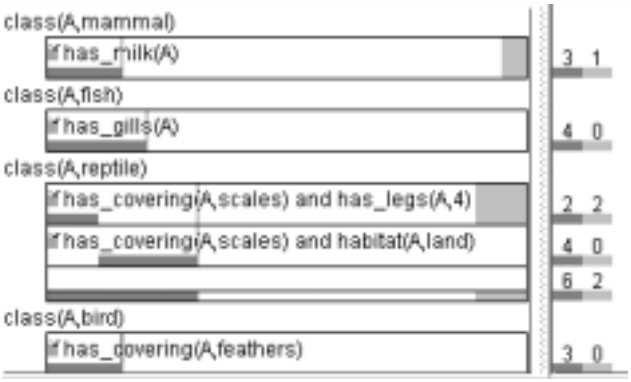


Fig. 3. Visualization of first order rules for the animals task (courtesy S. Moyle, Oxford University)

Figure 4 displays a set of subgroups discovered by Midos [8] in the Cleveland Heart Disease domain. Shown is the size of each subgroup, how it compares to the entire population and the distribution of the target values within each subgroup. Experience gained from working with non-technical end users has shown that a pie chart visualization is more appealing to these users because they more closely resemble business charts. Pie charts, however, often mislead the perception of the user due to difficulties with relating the size of pie slices to actual values. Hence, alternative visualizations are possible (see, for example [3]).

4 Discussion

The visualization tool presented is a simple, yet powerful tool that can function as a dissemination tool for data mining results. Its simplicity ensures that non-KDD users can operate the tool and interpret the results obtained by a data mining expert. Java technology ensures that platform issues are secondary and that results could even be part of online content management or workgroup support systems.

The proposed extension to the PMML AssociationModel should be seen as a first proposal of a first order PMML rule model. It does not at this time utilize the extension mechanism that can be used in PMML models. The reason for this deviation from the proposed extension procedure is that first order rules models are sufficiently generic data mining models to justify the existence of a distinct model type. However, significant effort has been extended to stay as close as possible to the terminology employed by the AssociationModel.

5 Future Work

The tool presented should be enhanced in three directions: (1) Addition of visualization methods for the other PMML models that are supported by the current standard.

Fig. 4. Visualization of selected subgroups for the Cleveland Heart Disease domain (generated by Midos, exported from Kepler)

(2) Addition of functionality that enables the user to edit PMML models. Straight forward editing operations are the deletion of entire rules or subgroups or of conditions within these. More complex editing operations are the modification of existing rules or the addition of entirely new rules. Likewise, the editing of decision and regression trees will be supported. (3) Addition of a model evaluator. A common request voiced by current users of the visualizer is to be able to evaluate single records or entire tables on the model that is displayed (and possibly modified by the user). However, in order to avoid a significant increase in tool complexity, it is envisioned to realize the PMML model evaluator as a separate tool.

ACKNOWLEDGMENT

This work has been supported in part by the EU funded project Data Mining and Decision Support for Business Competitiveness: A European Virtual Enterprize (IST-1999-11495). I am grateful to G. Meyer and the members of the data mining group for providing sample PMML models. S. Moyle generated the first order rules for the Animals domain. The visualizations presented here were developed by A. and G. Andrienko, AIS, FhG, Sankt Augustin, Germany.

References

1. Data Mining Group, see www.dmg.org
2. U.M. Fayyad, G.G. Grinstein, and A. Wierse, Information visualization in data mining and knowledge discovery. Morgan Kaufmann, (2002).
3. D. Gamberger, N. Lavrač, D. Wettschereck, Subgroup Visualization: A Method and Application in Population Screening. *ECAI 2002 Workshop on INTELLIGENT DATA ANALYSIS IN MEDICINE AND PHARMACOLOGY*, (2002).
4. H.Y. Lee, H.L. Ong, and L.H. Quek, Exploiting visualization in knowledge discovery. In *Proc. of the First Inter. Conference on Knowledge Discovery and Data Mining*, pp. 198-203, (1995).
5. S. Müller, diplom thesis, University of Magdeburg, (2000).
6. Workshop on visual data mining, PKDD 2001, Freiburg, Germany, (2001). http://www-staff.it.uts.edu.au/˜simeon/vdm_pkdd2001/
7. A. Unwin, Visualisation for data mining, (2000). http://www1.math.uni-augsburg.de/ unwin/
8. S. Wrobel, An algorithm for multi-relational discovery of subgroups. *Proc. First European Symposium on Principles of Data Mining and Knowledge Discovery*, 78–87, Springer, (1997).

Feature Selection with Labelled and Unlabelled Data

Shaomin Wu and Peter A. Flach

Department of Computer Science, University of Bristol,
Woodland Road, Bristol BS8 1UB, U.K
{shaomin,flach}@cs.bris.ac.uk

Abstract. Most feature selection approaches perform either exhaustive or heuristic search for an optimal set of features. They typically only consider the labelled training set to obtain the most suitable features. When the distribution of instances in the labelled training set is different from the unlabelled test set, this may result in large generalization error. In this paper, a combination of heuristic measures and exhaustive search based on both the labelled dataset and the unlabelled dataset is proposed. The heuristic measures concerned are two contingency table measures — Goodman-Kruskal measure and Fisher's exact test — which are used to rank the feature according to how well a feature predicts the class. Secondly, an exhaustive search is employed: by using test for goodness-of-fit, information on both the labelled dataset and the unlabelled dataset is applied to choose a better combination of features. We evaluate the approaches on the KDD Cup 2001 dataset.

1. Introduction

Feature selection aims at finding a feature subset that can describe the data for a learning task as good as or better than the original dataset. It is of importance for both data mining and machine learning, in particular for high-dimensional data. Most algorithms for feature selection perform either heuristic or exhaustive search [1]. Heuristic feature selection algorithms estimate the feature's quality with a heuristic measure, for instance, information gain [2], Gini index [3], discrepancies measure [4] and chi-square test [5]. Other examples of heuristic algorithms include the Relief algorithm [6] and its extension, the PRESET algorithm [7]. Exhaustive feature selection algorithms search all possible combinations of features and aim at finding a minimal combination of features that is sufficient to construct a model consistent with a given set of instances, for example, the FOCUS algorithm [8].

In supervised learning we use a labelled training set to obtain a model, which is then executed on an unlabelled test set to obtain predictions. However, the model developed from the labelled dataset may not perform well on prediction for the unlabelled dataset because of differences in class distribution and cost distribution between the labelled dataset and the unlabelled data. For classification, ROC analysis [9] can be used to choose the best model from a model set if distribution of positives

and negatives and distribution of misclassification costs for the unlabelled dataset are given. Whereas misclassification costs may be given, it is often impossible to obtain the class distribution of positives and negatives for the unlabelled data. What can be obtained from the unlabelled data is information about the distribution of instances. For instance, the transduction technique [10] aims at maximizing the classification margin on both the labelled and the unlabelled data.

Algorithms for both heuristic and exhaustive feature selection in the literature, however, only focus on the labelled dataset. This may lead to large generalization error when the instance distribution in the labelled dataset is different from that of the unlabelled data. This paper introduces two feature selection approaches: feature selection based on the Goodman-Kruskal measure and feature selection based on both labelled and unlabelled datasets. The Goodman-Kruskal measure is used to select a subset of features, which is then exhaustively searched for a sub-subset with similar distributions in both the labelled and the unlabelled datasets. Experimental evaluation shows that the proposed approach performs well compared with other feature selection approaches.

The paper is organized as follows. Section 2 introduces two contingency table measures — the Goodman-Kruskal measure and Fisher's exact test — to rank the importance of features. Section 3 proposes a new feature selection approach based on the unlabelled dataset and the Chi-squared test for goodness-of-fit. Section 4 evaluates the approach on the KDD Cup 2001 dataset [11]. Section 5 concludes with a discussion and the main conclusions.

2. Heuristic measures for feature selection

Heuristic feature selection algorithms search the feature set with a heuristic measure such as information gain. Assume that the input features are independent of each other, we can compare associations between each input feature and the class to select important features. Let the value of the class be P (positive) or N (negative), and the value of an input feature be C_1…, C_r, then a contingency table can be built up as follows.

	Class = P	Class = N	
Input Feature = C_1	n_{1P} (μ_{1P})	n_{1N} (μ_{1N})	n_{1*}
…	…	…	…
Input Feature = C_r	n_{rP} (μ_{rP})	n_{rN} (μ_{rN})	n_{r*}
	n_{*P}	n_{*N}	n

Table 1. An $r \times 2$ contingency table

In table 1, n_{ij} is the number of instances for which the value of a feature is C_i and the value of the class is j. $n_{*j} = \sum_{i=1}^{r} n_{ij}$, $n_{i*} = n_{iP} + n_{iN}$, $n = n_{*P} + n_{*N}$, and

$\mu_{ij} = \frac{n_{*j} n_{i*}}{n}$, where n is the number of instances in the labelled data, $i = 1, \cdots, r$ and $j=P, N$. The table has r–1 degrees of freedom.

2.1 Chi-squared measure

Most methods to measure the association between two features in a contingency table are based on the Chi-squared test [5]. The Chi-squared measure can be used to measure the association between class and input feature. In the 2-by-r case in Table 1 it is defined as

$$c^2 = \sum_{i=1}^{r} \left(\frac{(n_{iP} - m_{iP})^2}{m_{iP}} + \frac{(n_{iN} - m_{iN})^2}{m_{iN}} \right) \quad (1)$$

This value is compared with a threshold value corresponding to a confidence level. For instance, if r=2 (1 degree of freedom) the χ^2 value at the 5% level is 3.84 — if our χ^2 value is larger than that, the probability is less than 5% that discrepancies this large are attributable to chance, and we are led to reject the null hypothesis of independence.

2.2 Fisher's exact measure

If one uses the Chi-squared measure to test whether an association exists between two random variables, m_{ij}>5 should be satisfied for each i and j. When $m_{ij} \leq 5$ and r=2, Fisher's exact test can be applied to test the association. Assume that $m_{1P} \leq 5$, $n_{1*} \leq n_{*P}$ and $n_{1*} \leq n_{*N}$. Below is Fisher's exact measure [13]

$$P_F = \sum_{k=n_{1P}}^{n_{1*}} \frac{n_{1*}! n_{2*}! n_{*P}! n_{*N}!}{k!(n_{1*} - k)!(n_{*P} - k)!(n_{*N} - n_{1*} + k)!(n_{1*} + n_{2*})!} \quad (2)$$

This measure is normalised between 0 and 1. When P_F is less than 0.05, we are led to reject the null hypothesis of independence (at the 5% level). It should be noted that Fisher's exact test can become computationally expensive for large n and r.

2.3 Goodman-Kruskal measure

The Chi-squared measure and Fisher's exact test can only measure the association between two features, as they are symmetric in the two features. Goodman and Kruskal [14] introduced an asymmetric measure λ that measures the predictivity of one feature with respect to another, say, predicting class with an input feature. The measure is

$$\lambda = \frac{\sum_{i=1}^{r} \max(n_{iP}, n_{iN}) - \max(n_{*P}, n_{*N})}{n - \max(n_{*P}, n_{*N})} \quad (3)$$

where $0 \le \lambda \le 1$. λ=0 means no predictive gain when using an input feature to predict the class, and λ=1 means perfect predictivity. If we want to select features that have strong association with the class in a dataset, both n_{*P} and n_{*N} which are the number of instances in which the class equals to P and N, respectively, will be constant. In this case, a simplified version of the Goodman-Kruskal measure is

$$\lambda_0 = \sum_{i=1}^{r} \max(n_{iP}, n_{iN}) \quad (4)$$

Section 4 in this paper will give examples that performance of models based on features selected with Goodman-Kruskal measure is sometimes better than those based on Chi-squared measure and information gain measure.

2.4 Information gain

For the sake of comparison, we use a feature selection approach based on information gain. Information gain is commonly used as a surrogate for approximating a conditional distribution in the classification setting [15]. Below is a simplified version of information gain for our problem (the remaining part only depends on the class).

$$I_{gain} = -\sum_{i=1}^{r} \left(\frac{n_{iP}}{n_{1*}} \log \frac{n_{iP}}{n_{1*}} + \frac{n_{iN}}{n_{2*}} \log \frac{n_{iN}}{n_{2*}}\right) \quad (5)$$

3. Feature selection based on labelled and unlabelled data

Heuristic measures like the above can be used to rank features. However, such a ranking does not consider that the probability distribution of the features in the labelled dataset may be different from those in the unlabelled dataset. In general, a model developed from the labelled dataset may have large generalization error if the probability distribution of the model's features in the labelled dataset is considerably different from their distribution in the unlabelled dataset. Assume M features, say, $x_1, x_2, \cdots, x_M$, are selected with a certain criterion from N features, where $M \le N$, and a model below is built up based on the M features.

$$y = f(x_1, x_2, \cdots, x_M) \quad (6)$$

where y represents the class. The probability distribution of $x_1, x_2, \cdots, x_M$ in the labelled dataset is expected to be close to the one in the unlabelled data. In other words,

the closer the probability distributions of $x_1, x_2, \cdots, x_M$ between the labelled dataset and the unlabelled data, the lower generalization error the model (6) has.

Let the probability distribution of $x_1, x_2, \cdots, x_M$ in the labelled dataset be $F(x_1, x_2, \cdots, x_M)$. According to the assumption of the heuristic search, $x_1, x_2, \cdots, x_M$ are independent of each other. Therefore, $F(x_1, x_2, \cdots, x_M)$ can be simplified as

$$F(x_1, x_2, \cdots, x_M) = F_1(x_1)F_2(x_2)\cdots F_M(x_M) \quad (7)$$

where $F_i(x_i)$ $(i = 1,2,\cdots,M)$ is the probability distribution of x_i in the labelled data. Similarly, let the probability distribution of $x_1, x_2, \cdots, x_M$ in the unlabelled dataset be $G(x_1, x_2, \cdots, x_M)$, we have

$$G(x_1, x_2, \cdots, x_M) = G_1(x_1)G_2(x_2)\cdots G_M(x_M) \quad (8)$$

where $G_i(x_i)$ $(i = 1,2,\cdots,M)$ is the probability distribution of x_i in the unlabelled data.

If the distribution function $F(x_1, x_2, \cdots, x_M)$ and $G(x_1, x_2, \cdots, x_M)$ come from the same distribution, the performance of the model on the labelled dataset and on the unlabelled dataset will be similar. If the probability distributions $F_i(x_i)$ and $G_i(x_i)$ are similar, the distribution functions $F(x_1, x_2, \cdots, x_M)$ and $G(x_1, x_2, \cdots, x_M)$ will be similar.

Assuming that x_i is a categorical feature, the Chi-squared test for goodness-of-fit can be used to estimate whether two random variables come from the same distribution. For the labelled data, let π_{ij} be the probability that the value of feature i falls in category C_{ij}, $j = 1,2,...,C$, which can be estimated as

$$\pi_{ij} = \frac{\text{the number of instances with feature } i \text{ falling in category } C_{ij} \text{ in labelled data}}{\text{the total number of instances in training dataset}} \quad (9)$$

Similarly for the unlabelled data, let θ_{ij} be the probability that the value of feature i falls in category C_{ij}, estimated as

$$\theta_{ij} = \frac{\text{the number of instances with feature } i \text{ falling in category } C_{ij} \text{ in unlabelled data}}{\text{the total number of instances in working dataset}} \quad (10)$$

The Chi-squared statistic for goodness-of-fit can be employed to measure the closeness between the distribution π_{ij} and θ_{ij} for $j = 1,...,C$:

$$\chi_i^2 = n\sum_{j=1}^{C}\frac{(\theta_{ij} - \pi_{ij})^2}{\pi_{ij}} \quad (11)$$

The smaller the value of χ_i^2 is, the more similar the distributions π_{ij} and θ_{ij} are. We average this over all features as follows:

$$c_{new} = \frac{1}{M}\sum_{i=1}^{M} c_i^2 \tag{12}$$

which measures the similarity between $F(x_1, x_2, \cdots, x_M)$ and $G(x_1, x_2, \cdots, x_M)$.

We can now formulate our proposed feature selection approach. We first use a heuristic measure to select the N^* best features, then apply a exhaustive search based on measure χ_{new} to select the combination of M features which minimizes the value of χ_{new}, where $M < N^* < N$.

4. Experimental Evaluation

The thrombin dataset from KDD Cup 2001 consists of 139351 features and 1909 instances and one class in the labelled data. All features and the class are binary. There are 42 instances labelled 'A' (standing for 'active', the positive class) and 1867 instances labelled 'I' (standing for 'inactive', the negative class). Below is the contingency table.

	Class Activity =A	**Class Activity =I**	
Input Feature = '1'	$n_{1A}(\mu_{1A})$	$n_{1I}(\mu_{1I})$	n_{1*}
Input Feature = '0'	$n_{0A}(\mu_{0A})$	$n_{0I}(\mu_{0I})$	n_{0*}
	42	1867	1909

Table 2. Contingency table.

A test dataset (below we call it the unlabelled data) with 634 unlabelled instances is given. We will use this dataset to test the performance of models. ROC analysis is used compare the performances of several classifiers within a ROC space. It allows, through the construction of the convex hull of a set of points, identification of classifiers that are optimal under certain parameter settings. Once the application context is known, say, distribution of positives and negatives and misclassification costs, the optimal classifiers can be determined from the convex hull. Only the classifiers on the convex hull are optimal under some circumstances.

4.1 Heuristic feature selection

Chi-squared measure, Fisher's exact measure, Goodman-Kruskal measure and information gain measure discussed above are used to select the features.

If the measure χ^2 in equation (1) is used to select features, 120941 features can be selected from the labelled dataset when a criterion $\chi^2 > 3.84$ is applied.

When $n_{1*} < 228$, μ_{1A} will be less than 5. Chi-squared test will not be suitable for the case and Fisher's exact measure P_F in section 2.2 can be used. In order to simplify the calculation, Fisher's exact test P_F in equation (2) is applied no matter whether n_{1*} is greater than or less than 228. 102326 features whose P_F values are all less than 0.05 can be selected.

By using Goodman-Kruskal measure in equation (4), we can select 51540 features whose λ are greater than zero.

If information gain measure in equation (5) is used to select the features, a set of features with ascending order of measure can be obtained. The set only shows the importance of each feature.

At the same time, the ID3 algorithm is used to build a decision based on the whole labelled dataset, and eight features occur in the tree.

Because we only focus on the comparison of different measures instead of the number of features to be selected, in order to compare and simplify our calculation, analogous to the number of features selected by the ID3 decision tree, eight features selected with other measures are chosen to develop models.

As it turns out, the first eight features selected with Fisher's exact measure are the same as those selected with the information gain measure. The order of features selected with Fisher's exact measure and information gain measure is very similar. Unfortunately, it is hard to prove that measure of P_F in equation (2) and the measure of I_{gain} in equation (5) can lead a similar result. Because the features selected by information gain and those by Fisher's exact measure are the same, we only compare the information gain measure with other measures below.

Based on the labelled data, logistic regression model, Naive Bayes model, IB1 model, support vector machine---sequential minimal optimisation (SMO) algorithm, Kstar model and ID3 decision tree are built. Descriptions of those approaches can be found in [12]. The Weka toolkit is used to build the models. We use ten-fold cross validation to estimate the error of a model on the labelled data. In order to explain the evaluation metrics used, let the confusion matrix be

	Positive examples	Negative examples
Instances predicted positive	a	b
Instances predicted negative	c	d

Table 3. A confusion matrix

Then the metrics are as follows:

$$\text{Accuracy} = \frac{a+d}{a+b+c+d},$$

$$\text{Recall Average} = \frac{1}{2}\left(\frac{a}{a+c} + \frac{d}{b+d}\right),$$

True Positive Rate= $\frac{a}{a+c}$, and False Positive Rate= $\frac{c}{b+d}$.

A model is expected to possess high accuracy, high recall average, high TPrate and low FPrate.

In Table 4, for instance, 0.833(0.991) in the second row and the second column means, recall average=0.991 and accuracy=0.833 for the logistic model on the labelled dataset (lbl). 0.484(0.416) in the third row and the second column means, recall average=0.484 and accuracy=0.416 for the logistic model on the unlabelled dataset (ulbl), and so on. ID3 in the first row means the feature set selected by the ID3 decision tree, Info Gain means information gain measure, Chi-squared means Chi-squared measure and Goodman means Goodman-Kruskal measure.

The underlined numbers indicate the maximum value in the same row in table 4. From the table, both accuracy and recall average are maximum for Naive Bayes model and support vector machine and accuracy for Prism model and ID3 decision tree are maximum for the unlabelled dataset when Goodman-Kruskal measure is used, and no performance for other measures is better than Goodman-Kruskal measure.

Model	ID3	Info Gain	Chi Squared	Goodman
Logistic (lbl)	0.833(0.991)	0.736(0.984)	0.749(0.986)	0.772(0.985)
Logistic (ulbl)	0.484(0.416)	0.509(0.626)	0.564(0.598)	0.546(0.597)
NaiveBayes (lbl)	0.838(0.979)	0.839(0.982)	0.820(0.988)	0.900(0.988)
NaiveBayes (ulbl)	0.480(0.402)	0.552(0.379)	0.538(0.435)	0.543(0.544)
SMO (lbl)	0.806(0.984)	0.806(0.984)	0.808(0.988)	0.819(0.987)
SMO (ulbl)	0.485(0.319)	0.500(0.457)	0.525(0.509)	0.564(0.615)
Prism (lbl)	0.812(0.989)	0.682(0.983)	0.756(0.987)	0.707(0.985)
Prism (ulbl)	0.501(0.497)	0.492(0.655)	0.577(0.587)	0.607(0.576)
ID3 (lbl)	0.867(0.990)	0.748(0.985)	0.808(0.988)	0.748(0.985)
ID3 (ulbl)	0.495(0.475)	0.542(0.642)	0.609(0.697)	0.618(0.651)
IB1 (lbl)	0.808(0.988)	0.795(0.986)	0.761(0.987)	0.748(0.985)
IB1 (ulbl)	0.548(0.555)	0.551(0.662)	0.541(0.584)	0.533(0.596)
Kstar (lbl)	0.702(0.986)	0.724(0.985)	0.774(0.990)	0.737(0.988)
Kstar (ulbl)	0.449(0.569)	0.534(0.632)	0.560(0.623)	0.544(0.645)

Table 4. Results based on different measures

Figure 1 is a ROC curve based on models for the unlabelled dataset. The X-axis and the Y-axis in the ROC curve represent FPrate and TPrate, respectively. The ROC curve shows that Naïve Bayes model based on features selected with information gain measure, both Prism model and ID3 decision tree based on features selected with Goodman-Kruskal measure and ID3 decision tree based on features selected with Chi-squared measure are on the convex hull. In other words, those models are the optimal models in certain circumstances.

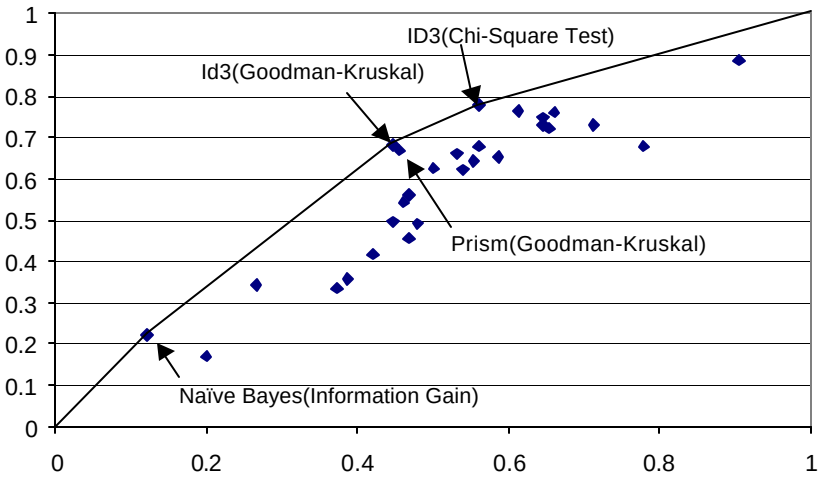


Fig. 1. ROC curve---comparison of different measures

4.2 Feature selection based on labelled and unlabelled data

By using Goodman-Kruskal measure, all features in the labelled dataset can be ranked in a descending order. The first twenty and thirty features from the ranked feature set are selected to form two feature sets, respectively. Then an exhaustive search for this two sets is performed to find the best combination of eight features to minimize χ_{new} in equation (12). Let F20 and F30 be the exhaustive search on the twenty-feature set and on the thirty-feature set, respectively. Table 5 shows the performances of models based on features selected with different measures.

The underlined numbers indicate the maximum values in the same row for the unlabelled dataset in table 5. From the table, only Naïve Bayes model based on features selected with information gain measure and ID3 decision tree based on features with Goodman-Kruskal measure have higher accuracy than F20 search or F30 search. Logistic model based on features selected with information gain measure, support vector machine model based on features selected with Goodman-Kruskal measure and ID3 decision tree model based on features selected with Chi-squared measure have higher recall average than F20 search or F30 search.

Figure 2 is a ROC curve based on prediction of models for the unlabelled dataset. The ROC curve shows that Naïve Bayes model based on features selected with information gain measure, Prism model on features selected with F20 are on the convex hull, IB1 on features selected with F20, Prism model on features selected with F30 and Kstar-kstar model on features selected with F30 are close to convex hull, which mean that F20 and F30 are better than other feature selection approaches.

Model	ID3	Info Gain	Chi Squared	Goodman	F20	F30
Logistic (lbl)	0.833(0.991)	0.736(0.984)	0.749(0.986)	0.772(0.985)	0.772(0.985)	0.784(0.987)
Logistic (ulbl)	0.484(0.416)	0.509(0.626)	0.564(0.598)	0.546(0.597)	0.559(0.623)	0.588(0.691)
NaiveBayes (lbl)	0.838(0.979)	0.839(0.982)	0.820(0.988)	0.900(0.988)	0.785(0.990)	0.820(0.990)
NaiveBayes (ulbl)	0.480(0.402)	0.552(0.379)	0.538(0.435)	0.543(0.544)	0.514(0.569)	0.579(0.691)
SMO (lbl)	0.806(0.984)	0.806(0.984)	0.808(0.988)	0.819(0.987)	0.761(0.986)	0.761(0.987)
SMO (ulbl)	0.485(0.319)	0.500(0.457)	0.525(0.509)	0.564(0.615)	0.604(0.577)	0.576(0.696)
Prism (lbl)	0.812(0.989)	0.682(0.983)	0.756(0.987)	0.707(0.985)	0.723(0.987)	0.774(0.989)
Prism (ulbl)	0.501(0.497)	0.492(0.655)	0.577(0.587)	0.607(0.576)	0.688(0.735)	0.614(0.722)
ID3 (lbl)	0.867(0.990)	0.748(0.985)	0.808(0.988)	0.748(0.985)	0.738(0.988)	0.773(0.988)
ID3 (ulbl)	0.495(0.475)	0.542(0.642)	0.609(0.697)	0.618(0.651)	0.644(0.667)	0.610(0.721)
IB1 (lbl)	0.808(0.988)	0.795(0.986)	0.761(0.987)	0.748(0.985)	0.785(0.988)	0.773(0.988)
IB1 (ulbl)	0.548(0.555)	0.551(0.662)	0.541(0.584)	0.533(0.596)	0.677(0.744)	0.569(0.689)
Kstar (lbl)	0.702(0.986)	0.724(0.985)	0.774(0.990)	0.737(0.988)	0.726(0.988)	0.691(0.986)
Kstar (ulbl)	0.449(0.569)	0.534(0.632)	0.560(0.623)	0.544(0.645)	0.571(0.716)	0.561(0.751)

Table 5. Comparisons of different approaches.

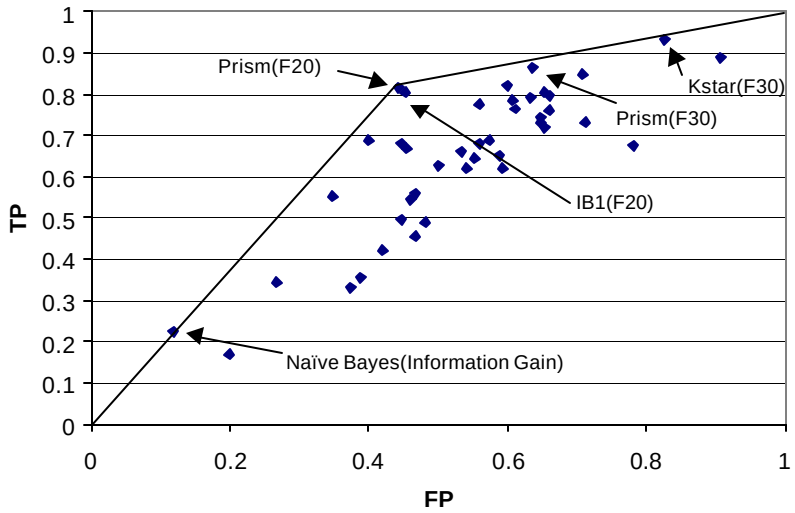


Fig. 2. ROC curve comparing different measures based on labelled and unlabelled data.

5. Concluding remarks

In the above experiments in section 4, we first selected candidate feature set (say, twenty and thirty features in this paper) from features selected with Goodman-Kruskal measure, then use χ_{new} in equation (13) to search a best combination of eight features

among the candidate feature set. However, if the candidate feature set is too large, some features with small Goodman-Kruskal measure may be included. Therefore, if more features are searched with χ_{new}, a smaller χ_{new} may probably be achieved, but prediction association between features and the class will degrade which means the performance of models based on features selected with χ_{new} will become poor. If few features are searched with χ_{new}, a larger χ_{new} is probably obtained. That means that the distribution of the labelled dataset and that of the unlabelled dataset may have a big difference, which will lead to larger generalization error. In other words, there is a trade-off between the size of search space and the value of χ_{new}.

Contingency table measures have been discussed by statisticians for a long time. The most well-known technique for analyzing the contingency table is the Chi-squared test. Furthermore, Fisher's exact test is used to test on contingency table with small expectations and Goodman-Kruskal measure is used to measure the prediction association. This paper firstly borrowed Fisher's exact test, Goodman-Kruskal measure to select feature. Below we summarize the results given in this paper

A. The rank order with Fisher's exact measure is similar to the one with information gain measure.
B. The performance of feature selection based on the Goodman-Kruskal measure is better than those based on other measures.
C. Feature selection with a measure based on the features from the labelled dataset and the unlabelled dataset has a lower generalization error than those based only on the labelled dataset.

Acknowledgements

This work is supported by the Esprit V project (IST-1999-11495) *Data Mining and Decision Support for Business Competitiveness: Solomon Virtual Enterprise*. We thank DuPont Pharmaceuticals Research Laboratories and KDD Cup 2001 for graciously providing this data set. Thanks are also due to the anonymous reviewers for their insightful comments and suggestions.

Reference:

[1] Blum, A., Langley, P., *Selection of relevant features and examples in machine learning*. Artificial intelligence, 97, 1997, pp.245-271

[2] Hunt, E., martin, J., and Stone, P. *Experiments in Induction*. Academic Press, New York, 1966

[3]Breiman, L., et. al., *Classification and regression trees*. Wadsworth Inc., Belmont, California, 1984

[4] Mantara, R., *ID3 revisited: A discrepancies based criterion for attribute selection*. In Proceedings of International Symposium Methodologies for Intelligent Systems, Charlotte, North Carolina, USA, 1989

[5] Lehmann, E. L., *Testing statistical hypothesis*. Springer, 199

[6] Kira, K., and Rendell, L., *The feature selection problem: Traditional methods and a new algorithm.* In Proceedings of the tenth National Conference on Artificial intelligence. Menlo Park: AAAI Press/The MIT Press, 1992, pp. 129-134

[7] Modrsejewski, M., *Feature selection using rough sets theory*, in P.B.Brazdil, ed., Proceedings of the European Conference on Machine Learning, Springer, 1993, pp. 213-226

[8]Almuallim, H., and Dietterich, T., *Learning boolean concepts in the presence of many irrelevant feature*. Artificial Intelligence, 69(1-2), 1994, pp. 279-305

[9] Provost, F., and Fawcett, T., *Robust Classification for Imprecise Environments, Machine Learning*, 42, 2001, pp. 203–231.

[10] Vapnik, V. N., *Statistical Learning Theory*. Wiley, 1998.

[11] Page, D, www.cs.wisc.edu/~dpage/kddcup2001/

[12] Witten, I. H., and Frank E., *Data Mining -Practical Machine Learning Tools and Techniques with Java Implementations.* Morgan Kaufmann, 2000

[13] Goodman, L. A. and Kruskal, W. H., *Measures of association for classifications.* J. Amer. Statist. Ass. 49, 1954, pp.732-764

[14] Agresti, A., *A Survey of Exact Inference for Contegency Tables.* Statitical Science, **7**, 1992, pp.131-153

[15] Cover, T. and Thomas, J., *Elements of Information Theory*, Wiley, 1991

Model Selection for Dynamic Processes

Shaomin Wu and Peter A. Flach

Department of Computer Science, University of Bristol,
Woodland Road, Bristol BS8 1UB, U.K
{shaomin,flach}@cs.bris.ac.uk

Abstract. In machine learning, ROC (Receiver Operating Characteristic) analysis is widely used in model selection when we consider both class distribution and misclassification costs that must be given at test time. In this paper we consider the case of a dynamic process, such that the class distributions are different in different time periods or states. The main problem is then to decide when to change models according to the different states of the generating process. In this paper we use a control chart to choose models for the process when misclassification costs are considered. Four strategies are considered and model selection approaches are discussed.

1. Introduction

In machine learning, ROC analysis for two classes measures the quality of models by studying the distribution of true positive rates and false positive rates of models. Both the class distribution and misclassification costs may be unknown during training time whereas they must be known at application time in order to select a suitable model. In practice, however, it may be difficult to know the exact class distribution which may change over time. In such cases, we need to know which point is a change point from one class distribution to another even when the class distributions in different periods may be known. Suppose instances $\{(\mathbf{X}_t, y_t), t=1,2,\ldots\}$ is a multivariate time series, where y_t is a binary class and $\mathbf{X}_t$ is a vector of independent features. From the ROC analysis point of view, we need to know the class distribution of y_t in order to choose a suitable model. In other words, we need to know where the change point from on state to another is.

The change-point detection has been discussed in [1,2,3]. A control chart, or cumulative count control chart (CCC-chart) can detect the change of class distributions that may be skewed in a process. This paper considers model selection by using CCC-chart.

This paper is organized as follows. Section 2 briefly reviews ROC analysis and CCC-chart. In Section 3, different situations from cost viewpoints will be taken into account and assumptions will be introduced. We distinguish four strategies for the different states of the process. Section 4 will give the costs for the four strategies and some analytical expressions for the average number of instances classified are derived. An example is given in section 5. Section 6 concludes.

2. ROC analysis and CCC-chart

ROC analysis [4, 5] studies the distributions of points (F,T) of models on a two-dimensional plane. Here, F stands for false positive rate (the ratio between the number of negative instances incorrectly classified and the total number of negative instances), and T stands for true positive rate (the ratio between the number of positive instances correctly classified and the total number of positive instances).

Assume that the relative frequency of negative instances in the test dataset is p. Assume that the cost for a correct classification is zero; the cost for classifying a positive instance to be a negative one is C_{pn} and the cost for classifying a negative instance to be a positive one is C_{np}. Then, the expected cost of applying model 1 with false positive rate and true positive rate (F_1,T_1) in the ROC space is $(1-p)(1-T_1)C_{pn}+pF_1C_{np}$. Similarly, the expected cost for model 2 is $(1-p)(1-T_2)C_{pn}+pF_2C_{np}$. Obviously, if $(1-p)(1-T_1)C_{pn}+pF_1C_{np}>(1-p)(1-T_2)C_{pn}+pF_2C_{np}$, then model 2 will be chosen. Otherwise, we shall choose model 1.

Assume labelled instances appear within a dynamic process one after another independently, and some candidate models can classify the instances into positives and negatives. An example would be a production line, where most items are manufactured correctly (positive) but some have production errors (negative). The number of positive instances until the next negative instance is observed is a geometric random variable. Let a process consist of two states S1 and S2 with relative frequencies of negative instances p1 and p2, respectively, where p1<p2. Let the probability of the event that the number of positive instances until a negative instance being observed is less than n_0 be α, or $P(n\le n_0)=\alpha$. Since n is a geometric random variable, we have $P(n\le n_0)= 1-(1-p_1)^{n_0}=a$ or $n_0=\log(1-\alpha)/\log(1-p_1)$ if the process is in S_1. Or if $n\ge n_0+1$, the process may be in S_1 with a probability $1-\alpha$ and n is called a type 1 signal (denoted as s_1). If $n\le n_0$, the process may have shifted to S_2 with a probability $1-\alpha$ and n is here called a type 2 signal (denoted as s_2). The approach here comes from CCC-chart methods[6, 7].

Because signal s_1 and s_2 show the state with a probability, they may be false ones. In order to confirm if a signal is true, an investigation may be carried out to check the true state of the process, which raise the different strategies in section 3. We assume that an investigation can recover the true state of the process. In what follows we make the following assumptions:

(1) Model 2 is more suitable for S_2 and model 1 is more suitable for S_1.
(2) When the process is in S_1, it may shift to S_2 with a probability π_{12}. When the process is in S_2, it may shift to another state with a probability π_{23}.

3. Four Different Strategies

One may decide whether a control chart will be used to monitor the process for different situations. We consider four possible strategies to decide when to switch between the two models.

(1) Strategy 1: In this strategy, no control chart will be used for the process. Because

the true state is not known, either model 1 or model 2 can be used throughout.

(2) Strategy 2: In this strategy, no control chart will be used. In order to know the exact state of the system, investigations for each instance are needed and two different models will be used according to the results of the investigations.

(3) Strategy 3: In this strategy, model 2 is used as soon as a signal s_2 appears. Although the signal s_2 may be a false one, no investigation on this signal will be carried out. In this strategy, the following two events may occur. Event A_1 — Before the process shifts to S_2, a signal s_2 appears when the process is in S_1, and then model 2 is used, and Event A_2 — in S_1, no signal s_2 appears. After the process shifts to S_2, a signal s_2 occurs when the process is in S_2, and then model 2 is used.

(4) Strategy 4: In this case, whenever a signal s_2 occurs, an investigation will be carried out to check the true state of the process. In this strategy, the following two events may occur. Event A_3 — before the process shifts to S_2, several s_2's occur and investigations are carried out. Model 2 is used until the process is confirmed to be in S_2 after a signal s_2 appears, and Event A_4 — in state 1, no signal s_2 appears. After the process shifts to S_2, a signal s_2 occurs in S_2 and an investigation is carried out and then model 2 is used.

4. Costs for the four strategies

Let $Q_{1(i)}$ (i=1,2) be the probability for signal s_i to appear when the process is in state S_1 and model 1 is being used, and $Q_{2(i)}$ (i=1,2) be the probability for a type i signal to appear when the process is in state S_2 and model 1 is still being used. Let $q_1=(1-p_1)(1-T_1)+p_1(1-F_1)$ and $q_2=(1-p_2)(1-T_1)+p_2(1-F_1)$, then we have $Q_{1(i)} = \sum_{j\in Z_i} q_1(1-q_1)^{j-1}(1-p_{12})^j$, $Q_{2(i)} = \sum_{j\in Z_i} q_2(1-q_2)^{j-1}(1-p_{23})^j$, where i=1,2

The probability of observing a transition of the process from state S_1 to state S_2 since the process starts is $Q_{1,2} = \sum_{j=1}^{\infty} p_{12}(1-q_1)^{j-1}(1-p_{12})^{j-1}$.

Recall that T_1 and F_1 are the true positive rate and the false positive rate of model 1, respectively, and T_2 and F_2 are the true positive rate and the false positive rate of model 2, respectively. Let the cost for investigating a signal be C_{in} and the cost for maintaining the CCC-chart be C_{chart}. Then, we can derive the following expressions for the expected cost for each of our four strategies.

Lemma 1. The expected cost for strategy 1 using model i is

$$c_{1i} = \left(\frac{1}{p_{12}}(1-p_1)+\frac{1}{p_{23}}(1-p_2)\right)(1-T_i)C_{pn} + \left(\frac{1}{p_{12}}p_1+\frac{1}{p_{23}}p_2\right)F_i C_{np}.$$

Lemma 2. The expected cost for strategy 2 is

$$c_2 = \left(\frac{1}{p_{12}}(1-p_1)(1-\mathrm{T}_1)+\frac{1}{p_{23}}(1-p_2)(1-\mathrm{T}_2)\right)C_{pn}+\left(\frac{1}{p_{12}}p_1\mathrm{F}_1+\frac{1}{p_{23}}p_2\mathrm{F}_2\right)C_{np}+\left(\frac{1}{p_{12}}+\frac{1}{p_{23}}\right)C_{in}$$

Lemma 3. The expected cost for strategy 3 is

$$c_3 = P(A_1)\Big(E_1\big((1-p_1)(1-\mathrm{T}_1)C_{pn}+p_1\mathrm{F}_1C_{np}\big)+E_3\big((1-p_1)(1-\mathrm{T}_2)C_{pn}+p_1\mathrm{F}_2C_{np}\big)$$
$$+\frac{1}{p_{23}}\big((1-p_2)(1-\mathrm{T}_2)C_{pn}+p_2\mathrm{F}_2C_{np}\big)\Big)+P(A_2)\Big(E_2\big((1-p_1)(1-\mathrm{T}_1)C_{pn}+p_1\mathrm{F}_1C_{np}\big)$$
$$+E_4\big((1-p_2)(1-\mathrm{T}_1)C_{pn}+p_2\mathrm{F}_1C_{np}\big)+\big(\frac{1}{p_{23}}-E_4\big)\big((1-p_2)(1-\mathrm{T}_2)C_{pn}+p_2\mathrm{F}_2C_{np}\big)\Big)+C_{chart}$$

Lemma 4. The cost for strategy 4 is

$$c_4 = \frac{1}{p_{12}}\big((1-p_1)(1-\mathrm{T}_1)C_{pn}+p_1\mathrm{F}_1C_{np}\big)+E_4\big((1-p_2)(1-\mathrm{T}_1)C_{pn}+p_2\mathrm{F}_1C_{np}\big)$$
$$+\big(\frac{1}{p_{23}}-E_4\big)\big((1-p_2)(1-\mathrm{T}_2)C_{pn}+p_2\mathrm{F}_2C_{np}\big)+P(A_1)C_{in}E_5+C_{in}+C_{chart}$$

where, $L_{1(i)} = \sum_{j\in Z_i} jq_1(1-q_1)^{j-1}(1-p_{12})^j$, $L_{2(i)} = \sum_{j\in Z_i} jq_2(1-q_2)^{j-1}(1-p_{12})^j$, i=1,2.

$L_0 = \sum_{j=1}^{\infty}(j-1)p_{12}((1-q_1)(1-p_{12}))^{j-1}$, $p(A_1) = Q_{1(2)}/(1-Q_{1(1)})$, $p(A_2) = Q_{1,2}/(1-Q_{1(1)})$

$E_1 = (Q_{1(2)}L_{1(1)}+L_{1(2)}(1-Q_{1(1)}))/(1-Q_{1(1)})^2$, $E_2 = (Q_{1,2}L_{1(1)}+L_0(1-Q_{1(1)}))/(1-Q_{1(1)})^2$,

$E_3 = 1-p_{12}E_2P(A_2)/(p_{12}P(A_1))-E_1$, $E_4 = (Q_{2(2)}L_{2(1)}+L_{2(2)}(1-Q_{2(1)}))/(1-Q_{2(1)})^2$,

$E_5 = E_3/E_1$.

5. Example

Let p_1=0.002, p_2=0.008, T_1=0.995, T_2=0.990, F_1=0.004, F_2=0.002, C_{np}=1000, C_{np}=1 and α=0.05. It can be shown that we should use model 1 in S_1 and model 2 in S_2, respectively. When π_{12}=0.00002, π_{23}=0.00006, from Lemma 1, Lemma 2, Lemma 3 and Lemma 4, we can obtain

A. If C_{chart}=0 and C_{in}==0, then c_{11}=1265, c_{12}=1131, c_2=1081.5, c_3=1130.1 and c_4=1081.7, both strategy 2 and strategy 4 are the best cases;
B. If C_{chart}=0 and C_{in}=0.5, then c_{11}=1265, c_{12}=1131, c_2=34414, c_3=1130.1 and c_4=1111.6, strategy 4 is the best;
C. If C_{chart}==100 and C_{in}=0.5, then c_{11}=1265, c_{12}=1131, c_2=34414, c_3=1230.1 and c_4=1211.6, strategy 1 with model 2 used in both states S_1 and S_2 is the best.

To sum up, the data analyst can choose a strategy to minimize the cost. Say, when the cost for maintaining the CCC-chart is small or the cost for investigating the state of the system is small, strategy 3 or strategy 4 may be the best choice. In other words, maintaining the CCC-chart for the process is helpful in these cases.

6. Conclusions

When the class distributions of different states in the process are known and the change point of the states is not known, it is hard to apply different models for the different states. This paper combines both ROC analysis and CCC-charts to optimize the cost. Four different strategies have been considered and expressions for the expected costs for each of these strategies have been obtained. This aids the data analyst in deciding which strategy to choose under particular cost distributions.

Acknowledgement

This work is supported by the Esprit V project (IST-1999-11495) *Data Mining and Decision Support for Business Competitiveness: Solomon Virtual Enterprise*. Thanks are due to the anonymous reviewers for their comments and suggestions.

References

1. Wang, Y.: Change-point analysis via wavelets for indirect data. Statistica Sinica, 9 (1999) 103-118
2. Pignatiello, P., Samuelsabre, T., Estimation of the Change Point of a Normal Process Mean in SPC Applications. Journal of Quality Technology, 33(1) (2001), 82-95
3. Loader, C.: Change Point Estimation Using Nonparametric Regression, Ann. Statist., 24 (1996) 1667-1678
4. Ferri, C., Flach, P., Hernandez, J.: Learning decision trees using the area under the ROC curve. Nineteenth International Conference on Machine Learning. July 2002
5. Provost, F., Fawcett, T.: Robust classification for imprecise environments. Machine Learning, 42(3) (2001) 202-231
6. Calvin, T. W.: Quality control techniques for 'zero-defects'. IEEE Transactions on Components, Hybrid and Manufacturing Technology, CHMT-6(3), (1983) 323--328.
7. Goh, T. N.: A control chart for very high yield processes. Quality Assurance, 13(1) (1987) 18--22.

Appendix

It is easy to prove the following results. When the process is in state 1, the number of instances immediately before the process has shifted to state 2 is a geometric random variable with parameter π_{12}, and expectation $1/\pi_{12}$. The expected number of instances since the transition of the process from state S_1 to state S_2 until it shifts to another state, is $1/\pi_{23}$. Under event A_1, the expected number of instances since the start of the process until the appearance of the first signal s_2 in S_1 with model 1 being used is E_1. The probability of event A_1 is $p(A_1)$ and the probability of event A_2 is $p(A_2)$. Under event A_2, the expected number of instances since the start of the process until the time of the transition from state S_1 to state S_2 and no s_2 appearing during that time with

model 1 being used is E_2. Under event A_1 or A_3, the expected number of instances since the time of the first appearance of signal s_2 until the transition from state S_1 to state S_2 is E_3. Under event A_2, the expected number of instances since the time of the transition from state S_1 to state S_2 until the appearance of the first signal s_2 in state 2 with model 1 being used is E_4. Under event A_3, the expected number of signal s_1 since the appearance of the first signal s_1 until the signal confirmed to be in state S_2 is E_5.

Proof of Lemma 1: The expected cost of applying model 1 in state 1 is $(1-p_1)(1-T_1)C_{pn}+p_1F_1C_{np}$. Similarly, the expected cost of applying model 1 in state 2 is $(1-p_2)(1-T_1)C_{pn}+p_2F_1C_{np}$. From the definition of strategy 1, and the above statement, we can obtain Lemma 1.

Proof of Lemma 2: If model 1 is being used in state 1 and model 2 is being used in state 2, the cost is $((1-p_1)(1-T_1)/\pi_{12}+(1-p_2)(1-T_2)/\pi_{23})C_{pn}+(p_1F_1/p_{12}+p_2F_2/p_{23})C_{np}$. In order to know the exact state of the system, investigations for each appeared instance are needed, the total cost for the investigation is $(1/p_{12}+1/p_{23})C_{in}$, then, we can get Lemma 2.

Proof of Lemma 3: For strategy 3,

(1) Under event A_1, the number of instances appearing before the appearance of the first signal s_2 in state 1 is E_1 and model 1 is being used during that time. The cost for this time period is $E_1((1-p_1)(1-T_1)C_{pn}+p_1F_1C_{np})$. The number of instances appearing since the appearance of the first signal s_2 until the time of the transition from state 1 to state 2 is E_3, and model 2 is being used during this time. The cost for this time is $E_3((1-p_1)(1-T_2)C_{pn}+p_1F_2C_{np})$. The number of instances since the system has shifted from state 1 to state 2 is $1/\pi_{23}$, then, the cost for this time period is $((1-p_2)(1-T_2)C_{pn}+p_2F_2C_{np})/\pi_{23}$.

(2) Under event A_2, the number of instances appearing in state 1 since the start of the process until the time of the transition from state 1 to state 2 is E_2 with model1 being used during that time. The cost for this time period is $E_2((1-p_1)(1-T_1)C_{pn}+p_1F_1C_{np})$. The number of instances appearing since the time of the transition from state 1 to state 2 until the appearance of the first signal s_2 is E_4,. The cost for this time period is $E_4((1-p_2)(1-T_1)C_{pn}+p_2F_1C_{np})$. The number of instances since the appearance of the first signal s_2 is $1/\pi_{23}-E_4$, then, the cost for this time period is $(1/\pi_{23}-E_4)((1-p_2)(1-T_2)C_{pn}+p_2F_2C_{np})$

To sum the above results of (1) and (2), and consider the probability of event A_1 and A_2, we get Lemma 3.

Proof of Lemma 4: For strategy 4, from the start of the process until transition from state 1 to state 2, model 1 is used; Between transition from state 1 to state 2 and appearance of the first signal s_2, model 1 is used. The number of instances in this time period is E_4.

(1) In state 2, after the first signal s_2 appears, model 2 is used; the number of the instances in this time period is $1/\pi_{23}-E_4$.

(2) In state 1, the number of investigations on signal s_2 when event A_3 and A_4 occur are $P(A_1)E_5$ and 0, in state 2, respectively. The number of investigations on signal s_2 when either event A_3 or A_4 occurs is 1.

Then, we can obtain Lemma 4

Author Index