A Statistical Learning Method to Fast Generalised Rule Induction Directly from Raw Measurements

Thien Le, Frederic Stahl, Chris Wrench
Department of Computer Science
University of Reading
Reading, United Kingdom
Email: t.d.le@reading.ac.uk
F.T.Stahl@reading.ac.uk
c.wrench@pgr.reading.ac.uk

Mohamed Medhat Gaber
School of Computing and Digital Technology
Birmingham City University
Birmingham, United Kingdom
Email: mohamed.gaber@bcu.ac.uk

***Abstract*—Induction of descriptive models is one of the most important technologies in data mining. The expressiveness of descriptive models are of paramount importance in applications that examine the causality of relationships between variables. Most of the work on descriptive models has concentrated on less expressive approaches such as clustering algorithms or rule-based approaches that are limited to a particular type of data, such as association rule mining for binary data. However, in many applications its important to understand the structure of the produced model for further human evaluation. In this research we present a novel generalised rule induction method that allows the induction of descriptive and expressive rules directly from both categorical and numerical features.**

I. Introduction

Several descriptive techniques exist, such as neural network based techniques i.e. KT and RULEX [1] and cluster analysis i.e. k-means and agglomerative clustering [2]. Cluster analysis and neural network based techniques are more or less black box approaches that can categorise or group data based on the relationship of features. Generalised Rule Induction techniques aim to address this problem by inducing rules that are more expressive by describing these relationships between features.

To the best of our knowledge, there is no single optimised method to deal with numerical features for descriptive rule induction algorithms. Many algorithms, including Apriori [3], were developed to work only for categorical values. Most of the time, the numerical feature needs to be converted into ones with a discrete set of values. The method of discretisation is often decided by the user and their justification of the suitability of the data and the chosen algorithm. This prior discretisation or binarisation process results in a loss of information or over-sensitivity, and thus a loss of expressiveness of the rule set induced by these algorithms [4]. In this research, we propose a new method for inducing generalised descriptive rules that can be induced directly from numerical data without the loss of expressiveness by inducing expressive rule terms from numerical data directly, utilising the probability density of Gaussian distribution.

This paper is organised as follows. **Section II** describes our proposed expressive rule based method for descriptive analytics and **Section III** provides an empirical evaluation. Lastly, **Section IV** provides some concluding remarks and directions for future work.

II. Induction of Generalised Rules

This section outlines the details of our proposed algorithm to induce generalised rules directly from training examples. The induced rules are represented in the form of a set of modular and expressive '**IF** *BODY* **THEN** *HEAD*' rules, where HEAD represents the right hand side of the rule, and BODY is the left hand side of the rule. The format of BODY and HEAD will be explained later in this section.

The algorithm consists of three main processes. These steps include generating all possible HEADs first and then generating all possible BODYs for each HEAD, and finally ranking the rules by a chosen criteria.

Typically, both HEAD and BODY are represented by a conjunction of feature-value pairs $(\alpha_{1,j}, \alpha_{2,j}, ..., \alpha_{n,j})$, where each $\alpha_{i,j}$ is a value of feature α_j and n is the number of possible values of the feature α_j. A feature can have either a finite (categorical) or infinite (numerical) set of values.

In this paper, we use the phrase 'rule term' as well as $\alpha_{i,j}$ as a definition for a feature-value of the feature α_j to refer to the logical test describing the properties of the data instances. The terms for a categorical feature can be presented in the form of $\alpha_j = v$ where $v \in \{\alpha_{1,j}, ..., \alpha_{n,j}\}$. A numerical feature is represented in form of $v_l < \alpha_j \leq v_h$, where $v_l, v_u \in \{\alpha_{1,j}, ..., \alpha_{n,j}\}$ as a result of using probability density based on a Gaussian distribution approach to induce a rule term for a numerical feature as described in **Section II-B**. The algorithm then appends the rule term that maximises the conditional probability of a particular HEAD.

A. Covering Strategy

Theoretically, if all possible rules are systematically generated from a training dataset and each rule is individually checked for the consistency and validity, then a consistent and complete model representation of the data should be realised. At first glance, this seems to be a simple approach to learning rules from the training data but, unlike classification, the proposed algorithm in this paper aims to uncover possible

rules to describe an underlying model of the data, and thus does not limit the right-hand-side or HEAD part to a single feature.

A rule can have up to *n* rule terms where *n* is the number of features in the dataset. Each feature can only appear once in a single rule term per rule and the HEAD and BODY must each have at least 1 rule term. Therefore, if a HEAD is limited up to a fixed length l, then the solution of the algorithm for searching all possible HEADs is polynomial in the number of steps to find a solution.

In this paper covering or 'Separate-and-Conquer' search strategy is used to induce new rule terms for both BODY and HEAD. The concept of covering algorithms was first utilised in the 1960s by Michalski [5] for the AQ family algorithms. The *separate* part induces a rule that covers a part of the training data and the *conquer* part recursively learns another rule that covers some of the remaining data examples until no more data examples remain.

B. Using Gaussian Distribution for Inducing Rule Terms from Numerical Features

For each numerical feature in a dataset, we generate a Gaussian distribution to represent all possible values of that numerical feature for a given HEAD. This method is inspired by earlier works of one the author's on classification rule induction algorithms [6]. The generation of possible HEADs is explained in **Section II-C**.

Assume a dataset with a set of generated HEADs, $h_1, h_2, ..., h_i$. If we have a vector of feature-values then we can compute the feature-value that is the most relevant one to a particular HEAD based on the Gaussian distribution of the values associated with this HEAD. The Gaussian distribution is calculated for a numerical feature α_j with mean μ and variance σ^2 from all feature-values associated with HEAD, h_i. The conditional density probability is then given by:

$$P_{density}(\alpha_j = v|h_i) = p(\alpha_j = v|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} exp(-\frac{((\alpha_j = v) - \mu)^2}{2\sigma^2}) \quad (1)$$

Then a heuristic measurement of the posterior class probability, $p(h_i|\alpha_j = v)$, or equivalently $log(p(h_i|\alpha_j = v))$ can be calculated and used to determine the probability of a target class for a valid value of a numerical feature.

$$log(p(h_i|\alpha_j = v)) = log(p(\alpha_j = v|h_i)) + log(p(h_i)) - log(p(\alpha_j = v)) \quad (2)$$

We calculate the probability of regions Ω_i for these feature-values such that if $v \in \Omega_i$ then v belongs to a h_i. This approach may not necessarily capture the full details of the intricate continuous distribution, but it is highly efficient in computation and memory perspectives. The reason is that the Gauss distribution only needs to be calculated once and can then be updated when data instances are deleted by simply recalculating mean μ and variance σ^2. The real values of a numerical feature are assumed to be normally distributed in this paper. As stated in central limit theorem [7], [8], if the sample size is large enough (> 30 or 40) then the sampling distribution tends to be normal, regardless of the actual underlying distribution of the data.

C. Inducing Possible Rule Terms for the HEAD

Let *n* be the number of features in the data and *k* be the number of all possible terms from all features. The users or domain experts should determine the real usefulness of HEAD length and may decide to select a group of the most interesting rules according to some subjective measure of interestingness. Thus *n* can also be a subset of the features defined by the user. The number of possible rule term conjunctions in the HEAD is $\frac{(n+k-1)!}{k!(n-1)!}$.

Note that a feature can only be used once in a rule in order to avoid contradictory rule terms. Producing every single possible conjunction of feature-value pairs can guarantee to uncover all the underlying knowledge, but this can be computationally very expensive. Thus, for each feature, only one value is considered for the HEAD. In the current implementation we simply choose the candidate rule term of a feature with the highest coverage of the training data.

D. Induce Complete Generalised Rules

For each HEAD generated in, all possible BODYs (conditional parts) are searched by a covering strategy as described in **Section II-A**. The conditional probability with which the BODY covers a given HEAD is used as a metric (to be maximised) to select a new rule term for the BODY. The algorithm always tries to induce a complete rule, this may lead to a low coverage and thus overfitting. This is because the usefulness of a complete rule with low coverage may be very limited in the case of prediction. However, this may not be the case for descriptive rules.

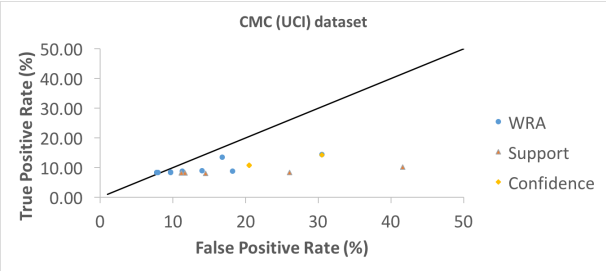
A complete rule can be an indication of something that is unusual and interesting for the analyst. Therefore, a descriptive model will retain all the rules, and the user can rank the rules by their coverage, accuracy, interestingness, or other metrics. The algorithm is not forced to produce complete rules all the time, and a pre-pruning can be applied to generalise the induced rules and avoid overfitting. In our proposed algorithm, a pre-pruning method was used to avoid overfitting during rule construction.

III. Experimental Evaluation

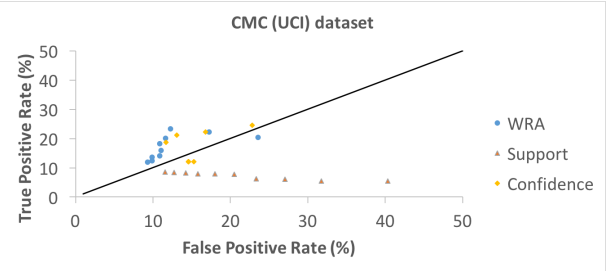
A. Experimental Settings and Datasets

The purpose of this experimental evaluation was to find out whether rule evaluation measures based on different constraints can be used to select *K* best rules from the proposed algorithm. To this end, we used ROC (Receiver Operating Characteristic) [9] space to evaluate a set of rules based on different criteria. There are two properties to determine the goodness of a ruleset statistically:

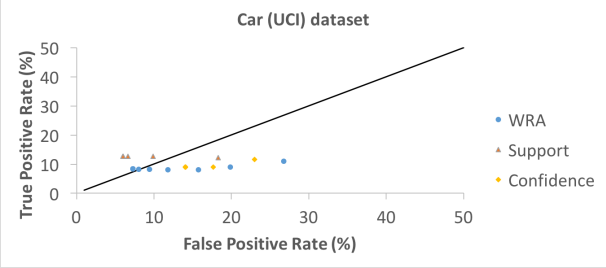
- **Completeness** - determines the number of positive data instances that are covered by the ruleset and this should be maximised.



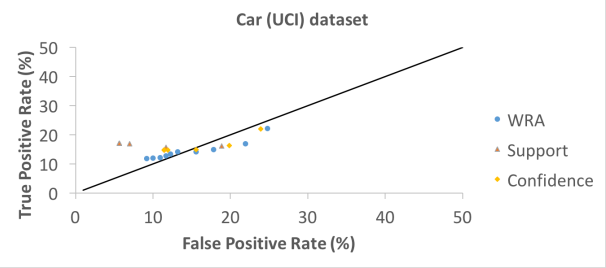
(a) Cmc dataset without pruning



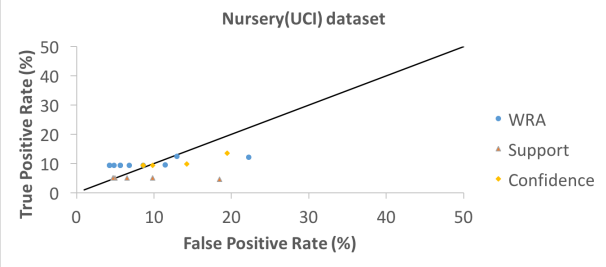
(b) Cmc dataset with pruning



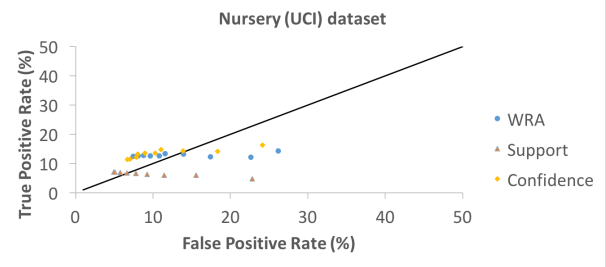
(c) Car dataset without pruning



(d) Car dataset with pruning



(e) Nursery dataset without pruning



(f) Nursery dataset with pruning

Fig. 1. Difference in accuracy compared with other classifiers for synthetic data streams.

- **Consistency** - determines the number of negative data instances that are covered by the ruleset and this should be minimised.

Generally, an induced rule labels a data instance as either positive or negative. Additionally, a rule can label a data instance as shown in **Table** I.

TABLE I
COVERED AND UNCOVERED POSITIVE AND NEGATIVE DATA INSTANCES.

	BODY cover		**BODY not cover**		
HEAD cover	True positives $\hat{P}$	+	False Negatives $\bar{P}$	=	P
HEAD not cover	False positives $\hat{N}$	+	True Negatives $\bar{N}$	=	N

Consequently, most rule evaluation measures are calculated based on $\hat{P}, \hat{N}, P$ and N. The used evaluation measures in this section are *Support*, *Confidence* and *Weighted Relative Accuracy* (WRA). Support and Confidence are commonly used in the literature to evaluate association rules, while WRA is a potentially more suitable metric to measure the interestingness of a descriptive model: $WRA(R_{set}) = \frac{\hat{P}+\hat{N}}{P+N} * (\frac{\hat{P}}{\hat{P}+\hat{N}} - \frac{P}{P+N})$

We selected datasets with both numerical and categorical values from the UCI machine learning repository [10] in order to evaluate the efficiency of our Gaussian based technique. We preferred datasets with clear descriptions of the data and features to allow a better understanding of the rules. We used **Car**, **CMC** and **Nursery** datasets.

The measures used in the evaluation are primarily intended for ranking and filtering rules' output by induction algorithms. These heuristics are of particular importance in descriptive induction algorithms since they can easily output several thousands of rules. We studied the resultant rule sets when ranked by Support, Confidence and WRA. The K top rules, where $K = \{5, 10, 15, \ldots, 50\}$ were plotted in ROC space.

B. Results and Discussion

The primary objectives of our experimental evaluation are to study the behaviour and performance focussing on the following evaluation questions: *How the induced rules performed in ROC space? How pruning can effect the rule quality? How to pick the K best rules from the induced ruleset?*

Figure 1a, **1b**, **1c**, **1d**, **1e** and **1f** show the ratio between *Tpr* (True positive rate) and *Fpr* (False positive rate) for

TABLE II

Tpr AND DISTANCE FROM THE DIAGONAL LINE FOR CAR, CMC AND NURSERY DATASETS. BOTH BEFORE & AFTER MEASURES ARE INCLUDED.

	Support				**Confidence**				**WRA**			
	Non Prunning		**With Pruning**		**Non Prunning**		**With Pruning**		**Non Prunning**		**With Pruning**	
	Tpr	**p-distance**	**Tpr**	**p-distance**	**Tpr**	**p-distance**	**Tpr**	**p-distance**	**Tpr**	**p-distance**	**Tpr**	**p-distance**
	CMC											
10 rules	8.36	-12.52	5.50	-18.56	10.76	-6.89	22.24	3.85	13.40	-2.40	22.23	3.54
20 rules	8.1	-2.50	6.16	-12.16	10.76	-6.90	18.70	4.95	8.86	-3.66	20.13	**6.00**
30 rules	8.1	-2.15	7.93	-7.11	10.76	-6.90	12.10	-1.75	8.52	-0.84	15.87	3.39
40 rules	8.1	-2.15	8.27	-4.22	10.76	-6.90	12.10	-1.75	8.25	0.21	13.69	2.68
50 rules	8.1	-2.15	8.58	-2.12	10.76	-6.90	12.10	-1.75	8.25	0.36	11.93	1.87
	Car											
10 rules	12.66	1.98	15.74	2.86	9.06	-6.10	16.36	-2.47	8.98	-7.68	16.88	-3.56
20 rules	12.77	4.80	17.10	**8.07**	9.04	-3.54	14.78	2.01	8.16	-2.57	14.10	-1.03
30 rules	12.77	4.80	17.10	**8.07**	9.04	-3.54	14.78	2.36	8.28	0.18	13.38	0.78
40 rules	12.77	4.80	17.10	**8.07**	9.04	-3.54	14.78	2.36	8.34	0.75	12.13	0.84
50 rules	12.77	4.80	17.10	**8.07**	9.04	-3.54	14.78	2.36	8.34	0.75	11.85	1.87
	Nursery											
10 rules	5.04	-3.40	6.04	-6.73	9.90	-3.08	14.22	-2.91	12.47	-0.33	12.09	-7.45
20 rules	5.11	0.09	6.37	-2.03	9.44	0.58	14.83	2.66	9.44	0.61	13.19	-0.54
30 rules	5.11	0.26	6.69	0.04	9.44	0.58	13.54	3.26	9.40	2.65	12.69	1.32
40 rules	5.11	0.26	7.01	1.36	9.44	0.58	12.09	2.99	9.40	3.24	12.78	2.84
50 rules	5.11	0.26	7.01	1.46	9.44	0.58	11.38	3.33	9.40	**3.64**	12.52	3.57

each selected *K* best rules. Any points plotted in the ROC space that are above the diagonal line should be considered as useful because the probability with which the BODYs cover the HEADs is higher than random. In all cases, the rules with pruning are better compared with the un-pruned rules as shown in the figures. Please note the data for the plots in **Figure 1** can be found in **Table II**. We discovered that the selection criteria performed differently on different datasets, and there is no single best heuristic for filtering the *K* best rules. Therefore, we believe that there is no universal heuristic that can provide precise information about the quality of an induced model and the ruleset. Thus, instead of relying upon a specific selection criteria, we evaluate the induced rules in the ROC space and select the *K* best rules no matter what selection criteria was used. By using *Tpr* and *Fpr* we are also reassured that the measures do not depend on the total number of training examples. We determine if a set of rules is better than another when its corresponding point in the ROC space is further away from the diagonal line positively. Thus, for each point in ROC space we also calculate its perpendicular distance to the diagonal and called it 'p-distance', which is also listed for our experiments in **Table** II. As diagonal line is used, the equation for calculating 'p-distance' is revised as: $p - distance = \frac{|Ax_0+By_0+C|}{\sqrt{a^2+b^2}} = -\frac{x_0-y_0}{\sqrt{2}}$

The 'p-distance' can be positive and negative which indicates if the plotted point in the ROC space is above or below the diagonal line respectively. For example: if a 'p-distance' is positively greater than another then its ratio between *Tpr* and *Fpr* is better (which is desirable) compared with another point. In contrast, if a 'p-distance' is negatively smaller than another then this is undesirable.

IV. CONCLUSION AND FUTURE WORK

This paper presented the work on a novel descriptive rule induction algorithm that produces highly expressive rules. The rule induction method is based on a covering approach and the algorithm uses several selection heuristics to select the best *K* rules. The induced rulesets are evaluated using ROC space to estimate a rulesets' usefulness. The produced rulesets show a usefulness better than random. A basic pruning method has also been employed and compared with the algorithm's rulesets induced without using a pruning method. The results in **Section III** indicate that a pruning method can improve the quality of the induced rules.

Acknowledgements. This research has been supported by the UK Engineering and Physical Sciences Research Council (EPSRC) grant EP/M016870/1.

REFERENCES

[1] R. Andrews, J. Diederich, and A. B. Tickle, "Survey and critique of techniques for extracting rules from trained artificial neural networks," *Knowledge-based systems*, vol. 8, no. 6, pp. 373–389, 1995.

[2] M. Bramer, *Principles of data mining*. Springer, 2007, vol. 180.

[3] R. Agrawal and R. Srikant, "Fast Algorithms for Mining Association Rules in Large Databases," *Journal of Computer Science and Technology*, vol. 15, no. 6, pp. 487–499, 1994.

[4] J. Fürnkranz, D. Gamberger, and N. Lavrač, *Foundations of rule learning*, 2012, no. 2003.

[5] R. S. Michalski, "On the quasi-minimal solution of the general covering problem," 1969.

[6] T. Le, F. Stahl, J. B. Gomes, M. M. Gaber, and G. D. Fatta, "Computationally efficient rule-based classification for continuous streaming data," in *Research and Development in Intelligent Systems XXXI*. Springer International Publishing, 2014, pp. 21–34.

[7] D. G. Altman and J. M. Bland, "Statistics notes: the normal distribution." *BMJ (Clinical research ed.)*, vol. 310, p. 298, 1995.

[8] A. C. Elliott and W. A. Woodward, "Statistical Analysis Quick Reference Guidebook: With SPSS Example," p. 280, 2007.

[9] J. A. Hanley and B. J. McNeil, "The meaning and use of the area under a receiver operating characteristic (ROC) curve." *Radiology*, vol. 143, no. 1, pp. 29–36, 1982.

[10] M. Lichman, "UCI machine learning repository," 2013. [Online]. Available: http://archive.ics.uci.edu/ml