An Evaluation of Discretization Methods for Learning Rules from Biomedical Datasets.

CONFERENCE PAPER · JANUARY 2008

Source: DBLP

CITATIONS
3

READS
65

4 AUTHORS, INCLUDING:

Jonathan Lustgarten
Red Bank Veterinary Hospital, Tinton Falls, N…
15 PUBLICATIONS **178** CITATIONS
SEE PROFILE

Shyam Visweswaran
University of Pittsburgh
73 PUBLICATIONS **401** CITATIONS
SEE PROFILE

Vanathi Gopalakrishnan
University of Pittsburgh
56 PUBLICATIONS **505** CITATIONS
SEE PROFILE

An Evaluation of Discretization Methods for Learning Rules from Biomedical Datasets

J.L. Lustgarten, S. Visweswaran, H. Grover, and V. Gopalakrishnan
Department of Biomedical Informatics, University of Pittsburgh, Pittsburgh, PA, USA

Abstract *- Rule learning has the major advantage of understandability by human experts when performing knowledge discovery within the biomedical domain. Many rule learning algorithms require discrete data in order to learn the IF-THEN rule sets. This requirement makes the selection of a discretization technique an important step in rule learning. We compare the performance of one standard technique, Fayyad and Irani's Minimum Description Length Principle Criterion, which is the defacto discretization method in many machine learning packages, to that of a new Efficient Bayesian Discretization (EBD) method and show that EBD leads to significant gains in performance especially as the complexity of the rule learner increases.*

Keywords: Rule Learning, Discretization, Data Mining, Machine Learning, Biomedical Datasets

1 Introduction

In machine learning and data mining, models that consist of a set of IF-THEN rules have been extensively used for knowledge discovery [1, 2]. A set of IF-THEN rules learned from data captures and expresses the knowledge contained in the data. A major advantage of rules is that they can be easily understood by human experts [3, 4]. Many rule learning algorithms have been described in the literature and typically such algorithms require discrete data. Some rule learning algorithms, such as those that learn decision trees, can handle continuous data by using a discretization technique internally during learning. Researchers have extensively studied discretization of continuous-valued attributes [5-8]. A commonly used discretization method is Fayyad and Irani's Minimum Description Length Principle Criterion (MDLPC) algorithm based on the Minimal Description Length Principle [8].

The use of an appropriate discretization method is important when learning rules in high dimensional genomic and proteomic datasets. In this paper, we evaluate a Bayesian discretization algorithm called Efficient Bayesian Discretization (EBD) for rule learning and compare its performance to that of MDLPC on 23 biomedical datasets. We show that EBD performs better than the commonly used MDLPC discretization method. The remainder of this paper is structured as follows. Section 2 briefly reviews the discretization algorithms and the different rule learning techniques. Sections 3 and 4 give the experimental methods and the results of evaluation of the EBD and MDLPC discretization methods. Section 5 discusses the results and some directions for future work.

2 Discretization Methods and Rule Learners

2.1 Discretization

Discretization is the process of transforming a continuous-valued attribute into a discrete one by creating a set of contiguous intervals (or equivalently a set of cutpoints) that together spans the range of the attribute's values. A discretization algorithm returns a discretization model (a set of cutpoints) for a continuous-valued predictor attribute from a dataset that contains values for that attribute. Discretization methods fall into two distinct categories: unsupervised, which do not use information about the class (target) attribute, and supervised methods, which do. Unsupervised techniques are either typically simple methods that split the range of predictor attribute values into a user specified number of intervals (Equal-Width) or a user specified number of instances per interval (Equal-Frequency). Supervised methods are more sophisticated and derive the best discretization from the data automatically, although, sometimes the user may be required to specify the maximum number of intervals. They typically utilize a score to measure the goodness of a discretization model and heuristically search the space of discretizations for a good scoring model.

Discretization methods can also be categorized into univariate and multivariate methods. Univariate methods search for the best discretization of each continuous-valued predictor attribute individually. Multivariate methods take into consideration the possible interaction of the predictor attribute being discretized with all other predictor attributes in the domain. In this paper, we focus on univariate supervised discretization methods since univariate methods are computational less expensive than multivariate methods and supervised methods tend to perform better than unsupervised methods on high-dimensional data [5] . We first briefly describe Fayyad and Irani's Minimum Description Length Principle Criterion (MDLPC) method, which is currently widely used, followed by the Efficient Bayesian Discretization (EBD) method that we have developed.

2.1.1 Fayyad and Irani's MDLPC

Fayyad and Irani's MDLPC discretization algorithm [8] is a univariate supervised discretization method that combines entropy and the minimum description length

principle to determine the best cutpoint for splitting an interval. This technique is recursive with a time complexity of $O(n \log n)$ where n is the number of instances in the dataset.. It is not optimal in that it does not examine all possible discretizations for an attribute in a given dataset; instead, it uses a greedy procedure. MDLPC is one of the most commonly used discretization methods in machine learning [9].

2.1.2 Efficient Bayesian Discretization (EBD)

Boullé developed a univariate supervised discretization method called the Minimum Optimal Description Length (MODL) algorithm based on the minimal description length (MDL) principle [7]. The MODL algorithm scores all possible discretization models and selects the one with the highest score. This is an optimal algorithm in that it examines all possible discretizations of an attribute given a dataset of values for the attribute and the corresponding target attribute values. MODL has been shown to have better performance than MDLPC on a large number of UC Irvine Machine Learning Repository (UCI) datasets. The optimal MODL algorithm as described by Boullé runs in $O(n^3)$ time where n is the number of instances in the dataset.

We have developed a new supervised discretization method called the Efficient Bayesian Discretization (EBD) algorithm that uses a Bayesian score to evaluate a discretization model. The Bayesian score is a generalization of the score used in the MODL algorithm. EBD, like MODL, is also an optimal algorithm but runs faster: in $O(n^2)$ time where n is the number of instances in the dataset.

2.2 Rule Learners

2.2.1 Conjunctive Rule Learner, RIPPER, and C4.5

These three rule learner algorithms represent common rule generation algorithms and are available in the Waikato Environment for Knowledge Analysis (WEKA) package [9]. Conjunctive Rule Learner is a simple rule learner that creates simple conjunctive rules that optimize the coverage and predictive accuracy. It uses a technique called Reduced Error Pruning (REP) [10] to trim an initial set of rules to the smallest and simplest subset of rules. Repeated Incremental Pruning to Produce Error Reduction (RIPPER), implemented as JRIP within WEKA, was developed by Cohen [11] which uses the REP technique, but performs multiple runs and has been shown to be effective in natural language processing [12]. C4.5 is a decision tree learner developed by Quinlan [13] that is an extension of the basic decision tree learner ID3.. These improvements include parameterization of the depth of the decision tree, post-pruning of rules, ability to handle continuous-valued attributes, and improved computational efficiency.

2.2.2 Rule Learner (RL)

Rule Learner (RL) is a knowledge-based inductive rule learner that has roots in the Dendral and Meta-Dendral programs [14]. Dendral can be considered the first expert system because it automated problem-solving and decision-making for organic chemists, by helping them analyze the mass spectra of unknown organic molecules using its knowledge of chemistry. MetaDendral is a learning program that generates hypotheses or models (chemical structures that correlate with observed mass spectrum), that Dendral uses for evaluation. This system later evolved into a data mining and rule discovery system with the creation of RL4 by Clearwater et al. [15]. RL4 was further improved in efficiency and speed through the use of Breadth First Marker Propagation [16] which allowed it to collect necessary statistics through a single pass over the data.. This resulted in a large speed up of RL4 and gave rise to the currently used RL algorithm. RL has been used in multiple domains including novel biomarker discovery and is considered to be a low variance and low bias algorithm [17]. We implemented RL in Java so that it can be used in conjunction with the WEKA package.

3 Experimental Setup

3.1 Biomedical Datasets

The performance of EBD and MDLPC were evaluated on 20 publicly available genomic datasets, and two publicly available proteomic datasets from the Surface Enhanced Laser/Desorption Ionization Time of Flight (SELDI-TOF) mass spectrometry platform, and one University of Pittsburgh proteomic dataset, which is a diagnostic dataset from a Amyotrophic Lateral Sclerosis (ALS) study on the SELDI-TOF platform. The datasets along with their types, number of instances, number of attributes, and the fraction of the data covered by the most abundant class are given in Table 1.

3.2 Machine Learning Techniques and Statistical Analysis

For our experiments, we used three rule learners (Conjunctive Rule, C4.5, RIPPER) [9, 11, 13] as implemented in WEKA version 3.5.6 [9]. For the fourth rule learner, we used our Java implementation of RL [2] in conjunction with WEKA. We used the WEKA implementation of the MDLPC and implemented EBD in Java so that it can be used in conjunction with WEKA.

We conducted experiments for each discretization method using 10-fold cross-validation done ten times. The discretization algorithms were evaluated using the following measures: classification accuracy and relative classifier information (RCI). Relative Classifier Information (RCI) is an entropy-based performance measure that quantifies how much the uncertainty of a decision problem is reduced by a classifier relative to classifying using only the prior probabilities of each class [18]. RCI's minimum value is 0% denoting the worst

Table 1. Datasets used for the discretization experiments. In the Type column G stands for genomic and P for proteomic. In the P/D column, P signifies prognostic and D signifies diagnostic. #A is the number of attributes in the dataset. M is the fraction of the data covered by the most abundant class.

Dataset	Dataset name	Type	P/D	# Classes	# Instances	# A	M
1	Alon et al	Genomic	Diagnostic	2	61	6584	0.651
2	Armstrong et al	G	D	3	72	12582	0.387
3	Beer et al	G	Prognostic	2	86	5372	0.795
4	Bhattacharjee et al	G	D	7	203	12600	0.657
5	Bhattacharjee et al	G	P	2	69	5372	0.746
6	Golub et al	G	D	4	72	7129	0.513
7	Hedenfalk et al	G	D	2	36	7464	0.500
8	Iizuka et al	G	P	2	60	7129	0.661
9	Khan et al	G	D	4	83	2308	0.345
10	Nutt et al	G	D	4	50	12625	0.296
11	Pomeroy et al	G	D	5	90	7129	0.642
12	Pomeroy et al	G	P	2	60	7129	0.645
13	Rosenwald et al	G	P	2	240	7399	0.574
14	Staunton et al	G	D	9	60	7129	0.145
15	Shipp et al	G	D	2	77	7129	0.506
16	Singh et al	G	D	2	102	12599	0.746
17	Su et al	G	D	13	174	12533	0.150
18	Veer et al	G	P	2	78	24481	0.562
19	Welsch et al	G	D	2	39	7039	0.878
20	Yeoh et al	G	P	2	249	12625	0.805
21	Petricoin et al	Proteomic	D	2	322	11003	0.784
22	Pusztai et al	P	D	3	159	11170	0.364
23	Ranganathan et al	P	D	2	52	36778	0.556

performance while the best performance is 100%, which signifies perfect discrimination. It is similar to area under the ROC curve (not equivalent) in that it measures the discrimination power of the classifier..

For comparing the performance of EBD and MDLPC we used the Wilcoxon paired samples signed rank test and the paired samples t-test. The Wilcoxon paired samples signed rank test is a non-parametric procedure used to test whether there is sufficient evidence that the median of two probability distributions differ in location. In evaluating algorithms, it can be used to test whether two algorithms differ significantly in performance on a specified measure. Being a non-parametric test, it does not make any assumptions about the form of the underlying probability distribution of the sampled population.

The paired samples t-test is a parametric procedure used to determine whether there is a significant difference between the average values of the same performance measure for two different algorithms. The test assumes that the paired differences are independent and identically normally distributed. Although the measurements themselves may not be normally distributed, the pair wise differences often are.

4 Results

The average accuracies for EBD and MDLPC for the four rule learners are given in Table 2. For each dataset, the average accuracy is obtained from 10-fold cross-validation done ten times for a total of 100 folds. For each dataset, the higher accuracy is shown in bold font.

There is variation from learner to learner with the accuracy increasing from the simpler to the more complex learners. For the Conjunctive Rule Learner, EBD has higher accuracies on 4 datasets, MDLPC has higher accuracies on 9 datasets and there are 10 ties. For Ripper, EBD has higher average accuracy on 11 datasets, MDLPC has higher accuracies on 9 datasets and there are 3 ties. For C4.5, EBD has higher accuracies on 9 datasets while MDLPC has higher accuracies on 10 datasets and there are 4 ties. RL shows the largest performance gain with EBD; for RL, EBD has higher accuracies on 16 datasets, MDLPC has higher accuracies 5 datasets and there are 2 ties. For the first three rule learners, the difference between EBD and MDLPC with respect to accuracy is not statistically significant on either the Wilcoxon rank test or the t-test at the 5% significance level (Table 4). For RL, EBD produces significantly higher accuracies than MDLPC on the Wilcoxon rank test but not on the t-test at the 5% significance level (Table 4).

The average RCIs for EBD and MDLPC across the multiple rule learners are given in Table 3. For each dataset, the average RCI is obtained from 10-fold cross-validation done ten times for a total of 100 folds. For each dataset, the higher RCI is shown in bold font.

Table 2. Average accuracies for EBD and MDLPC discretization algorithms for four rule learners. The average accuracies are based on a total of 100 folds. The higher accuracy is in bold font.

Dataset	Conjunctive Rule		RIPPER		C4.5		RL	
	EBD	MDLPC	EBD	MDLPC	EBD	MDLPC	EBD	MDLPC
1	98.69	98.69	100.00	100.00	100.00	100.00	100.00	100.00
2	70.28	70.28	**94.72**	93.61	97.92	**98.06**	**99.96**	98.79
3	80.23	80.23	93.84	**95.35**	97.67	97.67	**100.00**	99.87
4	74.29	**62.17**	**96.21**	95.67	**97.00**	96.85	97.20	**98.35**
5	75.36	**25.50**	92.46	**93.48**	**95.22**	94.78	**99.35**	98.54
6	**69.17**	69.03	93.75	93.75	96.11	**96.53**	100.00	100.00
7	96.94	**99.72**	99.44	99.44	99.72	99.72	**94.74**	75.79
8	66.67	66.67	**91.50**	91.17	93.17	**94.00**	**98.79**	97.98
9	61.93	**62.05**	**95.90**	94.94	96.14	**96.51**	78.65	**82.65**
10	**55.60**	50.60	**88.40**	87.60	**95.20**	94.80	91.63	**94.80**
11	76.89	76.89	91.78	**93.33**	**95.56**	95.44	**94.37**	72.36
12	73.83	**74.17**	**92.17**	89.83	**94.17**	92.33	**98.23**	96.26
13	62.17	**63.88**	85.67	**86.54**	91.25	**91.96**	91.25	**94.96**
14	25.50	25.17	**76.50**	75.50	83.83	**84.50**	84.73	**85.50**
15	87.01	**88.83**	**96.49**	96.23	97.53	**97.92**	**74.56**	71.24
16	88.24	**88.92**	**95.49**	95.00	95.98	**96.08**	**95.14**	92.08
17	28.51	**28.85**	88.16	**87.18**	93.05	**94.37**	**89.05**	88.97
18	**82.56**	81.92	94.23	**94.49**	**96.79**	96.92	**81.79**	77.79
19	95.90	95.90	97.69	**98.21**	98.46	98.46	**99.10**	98.46
20	80.75	80.75	92.71	**93.02**	**93.38**	92.99	**94.27**	92.99
21	79.38	79.38	93.45	**94.50**	94.91	**95.03**	**97.64**	96.57
22	53.21	53.21	**70.19**	69.25	**82.14**	77.23	**96.53**	81.25
23	**87.88**	87.69	**94.42**	92.12	**95.58**	95.38	**94.62**	91.36
Average	**72.65**	70.02	**91.96**	91.75	**94.82**	94.68	**93.55**	90.72

Table 3. Average RCI for EBD and MDLPC discretization algorithms for four rule learners. The average RCI is based on a total of ten runs of 10 folds for a total of 100 folds. The higher RCI is in bold font.

Dataset	Conjunctive Rule		RIPPER		C4.5		RL	
	EBD	MDLPC	EBD	MDLPC	EBD	MDLPC	EBD	MDLPC
1	73.20	73.20	100.00	94.37	100.00	100.00	100.00	100.00
2	**27.06**	25.33	45.59	**50.41**	**66.75**	62.78	**93.25**	91.27
3	0.19	0.19	2.26	**3.38**	**10.49**	9.82	**100.00**	92.56
4	**18.65**	17.78	**42.72**	42.10	50.45	**51.58**	**68.26**	54.42
5	**1.03**	0.71	0.58	**1.04**	**0.82**	0.70	31.26	**43.90**
6	**29.15**	24.74	**45.98**	45.33	**67.24**	66.17	**78.12**	73.19
7	70.57	70.57	70.57	70.57	70.57	70.57	100.00	100.00
8	0.56	**0.82**	**1.60**	1.53	**0.90**	0.89	**94.74**	75.79
9	**16.31**	15.71	66.33	**71.64**	**60.83**	60.17	**84.13**	76.50
10	**28.82**	28.34	31.81	**36.11**	**34.16**	33.08	44.63	**54.40**
11	4.48	**5.01**	27.72	**29.61**	**25.23**	21.66	**33.37**	0.00
12	**4.97**	4.54	**1.28**	1.10	**4.26**	4.12	**22.46**	17.93
13	**0.38**	0.36	**1.84**	1.82	**2.72**	1.92	7.95	**8.51**
14	5.97	**8.01**	26.12	**27.14**	**26.92**	23.89	60.92	**64.05**
15	**18.36**	17.40	**24.11**	21.93	**27.03**	25.19	**13.15**	10.88
16	**35.02**	34.77	**36.23**	35.66	26.47	**32.90**	**66.10**	47.69
17	7.90	**8.08**	**57.67**	55.81	**69.41**	67.03	**19.71**	7.83
18	**18.69**	18.06	**26.60**	23.58	**24.37**	22.18	**14.08**	13.89
19	17.01	17.01	**23.55**	23.51	23.71	23.71	**35.68**	28.24
20	0.02	0.02	**0.79**	0.56	0.64	**1.20**	**5.64**	3.35
21	**0.50**	0.37	11.71	**13.62**	7.22	**8.97**	**22.28**	16.69
22	10.20	10.20	**18.97**	16.95	**15.20**	14.04	**70.42**	48.79
23	**11.92**	11.38	**9.47**	8.04	**12.34**	12.11	**15.96**	9.60
Average	**17.28**	17.07	29.28	**29.38**	**31.64**	31.07	**51.40**	45.19

Table 4. Results of the Wilcoxon paired samples signed rank test and the paired samples t-test, over all datasets, comparing accuracy and RCI of EBD to MDLPC. Each table shows the results of a different rule learning algorithm. P-values that are significant at the 0.05 level are shown in bold font.

Conjunctive Rule		Avg.	Diff.	Wilcoxon p-value (Z-Score)	t-test p-value (t-Score)
Acc.	EBD MDLPC	72.65 70.02	2.63	0.875 (0.157)	0.251 (1.180)
RCI	EBD MDLPC	17.43 17.07	0.36	**0.039** (2.059)	0.132 (1.566)

RIPPER		Avg.	Diff.	Wilcoxon p-value (Z-Score)	t-test p-value (t-Score)
Acc.	EBD MDLPC	91.96 91.75	0.22	0.422 (0.803)	0.333 (0.989)
RCI	EBD MDLPC	29.28 29.38	-0.10	0.733 (0.341)	0.848 (-0.194)

C4.5		Avg.	Diff.	Wilcoxon p-value (Z-Score)	t-test p-value (t-Score)
Acc.	EBD MDLPC	94.82 94.68	0.14	0.717 (0.362)	0.577 (0.566)
RCI	EBD MDLPC	31.64 31.07	0.58	**0.037** (2.091)	0.205 (1.307)

RL		Avg.	Diff.	Wilcoxon p-value (Z-Score)	t-test p-value (t-Score)
Acc.	EBD MDLPC	93.55 90.72	2.83	**.048** (1.981)	.056 (2.020)
RCI	EBD MDLPC	51.40 45.19	6.20	**.007** (2.694)	**.008** (2.913)

Similar to the results seen with accuracy, there is variation from learner to learner with the average RCI increasing from the simpler to the more complex learners. For the Conjunctive Rule Learner, EBD has higher average RCI on 13 datasets, MDLPC has higher average RCI on 4 datasets and there are 6 ties. For Ripper, EBD has a higher average RCI on 13 datasets, MDLPC has higher average RCI on 8 datasets and there are 2 ties. With C4.5, EBD has higher average RCI on 16 datasets, MDLPC has higher average RCI on 4 datasets and there are 3 ties. With RL, EBD has higher average RCI on 17 datasets, MDLPC has higher average RCI on 4 datasets and there are 2 ties. On the Wilcoxon rank test, EBD has statistically significant higher RCIs on Conjunctive Rule, C4.5, and RL at the 5% significance level (Table 4). On the t-test, EBD has statistically significant higher RCIs on RL at the 5% significance level (Table 4).

Table 4 contains the results of the Wilcoxon paired samples signed rank test and the paired samples t-test over all the datasets comparing the accuracy and the RCI of EBD to MDLPC for all four rule learning techniques. All p-values are calculated using a two-tailed test. In all but one rule learning algorithm, regardless of the statistical test, accuracies were not statistically significantly different when comparing EBD to MDLPC. However, on RCI the difference between EBD and MDLPC was statistically significant in favor of EBD in three out of the four rule learners.

5 Discussion

Learning rules is an important method for knowledge discovery and induction of predictive models in high dimensional genomic and proteomic data. In this paper, we have examined the effects of a commonly used discretization algorithm, MDLPC, and a new Bayesian discretization algorithm, EBD, on several rule learners on a wide variety of biomedical datasets.

Our results show that overall EBD has better performance over MDLPC when evaluated on RCI but has similar performance when evaluated on accuracy. In addition, with learners that induce rules that are more complex (where there are more attributes in the antecedent) such as RL, EBD tends to perform better. One effect of a discretization method is the reduction in the number of attributes that are input to a rule learner, since attributes that are discretized to a single interval are effectively discarded by the learner. In our experiments, we observed that EBD tends to produce fewer one-interval attributes than MDLPC; this may explain in part why EBD does better with complex rule learners, which are able to effectively use more attributes..

In future work, we plan to extensively evaluate other discretization methods with rule learners. We also plan to develop and evaluate attribute-specific discretization methods where each attribute is discretized by a potentially different discretization method, which could lead to further improvement in performance.

6 Acknowledgements

We would like to thank the Bowser Lab at the University of Pittsburgh for use of the proteomic dataset that was analyzed in Ranganathan et al. (2005). This research was funded by grants from the National Library of Medicine (T15-LM007059 and R01-LM06696), the National Institute of General Medical Sciences (GM071951), and the National Science Foundation (IIS-0325581).

7 References

[1] F. Provost, J. M. Aronis, et al., "Rule-space search for knowledge-based discovery," Stern School of Business, New York University, NY, NY 10012 1999.

[2] V. Gopalakrishnan, P. Ganchev, et al., "Rule Learning for Disease-Specific Biomarker Discovery from Clinical Proteomic Mass Spectra," *Springer Lecture Notes in Computer Science*, vol. 3916, pp. 93-105, 2006.

[3] D. Heckerman and E. J. Horvitz, "On the Expressiveness of Rule-Based Systems for Reasoning under Uncertainty," in *Proceedings of the National Conference on Artificial Intelligence*, Palo Alto, CA, 1987.

[4] D. R. Carvalho and A. A. Freitas, "A hybrid decision tree/genetic algorithm method for data mining," *Information Sciences*, vol. 163, pp. 13-35, 2004.

[5] H. Liu, F. Hissain, et al., "Discretization: An enabling technique," *Data Mining and Knowledge Discovery*, vol. 6, pp. 393-423, 2002.

[6] R. Kohavi and M. Sahami, "Error-Based and Entropy-Based discretization of continuous features," in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, Portland, Oregon, 1996, pp. 114-119.

[7] M. Boullé, "MODL: A Bayes optimal discretization method for continuous attributes," *Machine Learning*, vol. 65, pp. 131-165, May 9th, 2006 2006.

[8] U. M. Fayyad and K. B. Irani, "Multi-interval discretization of continuous-valued attributes for classification learning," in *Proceedings of the Thirteenth International Joint Conference on AI (IJCAI-93)*, Chamberry, France, 1993, pp. 1022-1027.

[9] I. H. Witten and E. Frank, *Data Mining: Practical machine learning tools and techniques*, 2nd Edition ed. San Francisco: Morgan Kaufmann, 2005.

[10] J. Furnkranz and G. Widmer, "Incremental Reduced Error Pruning," in *Proceedings of the Eleventh International Conference on Machine Learning*, New Brunswick, NJ, 1994, pp. 70-77.

[11] W. W. Cohen, "Fast Effective Rule Induction," in *Proceedings of the Twelfth International Conference on Machine Learning*, Tahoe City, CA, 1995, pp. 115-123.

[12] W. W. Cohen, "Learning to Classify English Text with ILP Methods," *Advances in Inductive Logic Programming*, pp. 124-143, 1996.

[13] R. Quinlan, "C4.5: Programs for Machine Learning,," *Machine Learning*, vol. 16, pp. 235-240, 1/17/2005 1994.

[14] B. G. Buchanan and E. A. Feigenbaum, "Dendral and meta-dendral: Their applications dimension," *Artificial Intelligence*, vol. 11, pp. 5-24, 1978.

[15] S. H. Clearwater and F. J. Provost, "RL4: A Tool for Knowledge-Based Induction," in *Proceedings of the Second International IEEE Conference on Tools for Artificial Intelligence (TAI-90)*, Herndon, VA, 1990, pp. 24-30.

[16] J. M. Aronis and F. J. Provost, "Increasing the efficiency of data mining algorithms with breadth-first marker propagation," in *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining*, Newport, CA, 1997, pp. 119-122.

[17] T. Dietterich and E. B. Kong, "Machine Learning Bias, Statistical Bias, and Statistical Variance of Decision Tree Algorithms," Oregon State University 1995.

[18] V. Sindhwani, P. Bhattacharya, et al., "Information theoretic feature crediting in multiclass support vector machines," in *Proceedings of the First SIAM International Conference on Data Mining*, Chicago, IL, 2001.