Reducing Complexity of Rule Based Models via Meta Mining

Lukasz A. Kurgan
Department of Electrical and Computer Engineering, University of Alberta, Edmonton, Canada
lkurgan@ece.ualberta.ca

Abstract

Complexity, or in other words compactness, of models generated by rule learners is one of often neglected issues, although it has a profound effect on the success of any project that utilizes the rules. Researchers strive to propose learners that are characterized by excellent accuracy, and sometimes also low computational complexity, but the size of the data model generated by the learners is often not even reported. While the model size can be disregarded from the research point of view, it is very important from the end user's perspective. Quite often the generated model is too complex to be manually analyzed or inspected, which prohibits from using it in a real-world setting. To fill this gap, the paper proposes a novel framework, which is designed to address problem of complexity reduction of rule based models. The framework is based on a Meta Mining concept, and can be applied to enhance several of existing rule learners. Its main goal is to reduce complexity, in terms of reducing size and number of generated rules, without sacrificing accuracy of the rules. The paper proposes the framework, and tests it on a set of benchmark datasets using two well known rule learners: C5.0 and DataSqueezer. The results are encouraging, and show that 50% complexity reduction can be achieved virtually without any loss of accuracy.

Keywords: Rule Complexity, Inductive Machine Learning, Data Mining, Meta Mining, Rule Learner, C5.0, DataSqueezer.

1. Introduction

Machine Learning (ML) field provides a number of very popular and highly useful tools that are used to perform data mining tasks. This work focuses on a class of ML algorithms called rule induction systems. Induction is a technique that infers data models, or knowledge, by searching for regularities among the data. A rule induction system takes as input a set of supervised training examples. The output often takes the form of IF-THEN rules, or decision trees that can be converted into rules.

Rule induction systems have a number of advantages, over other data mining methodologies, which resulted in attracting significant research attention:

- They generate rules that are relatively easy for people to understand [5]. People often learn from the generated hypotheses, and even participate in the learning process, provided that the hypothesis is in a human-comprehensible form.
- The generated rules can easily be translated to a first-order logic representation, or embedded within knowledge-based or expert systems [14].
- The rules can be easily inspected and modified because of their modularity, i.e. a single rule can be understood without reference to other rules [23], which is very important when a decision maker needs to understand and validate the generated rules, as in medicine [28] [30].
- Certain types of prior knowledge were found to be easily communicated to rule learners [13] [35].

One of the main advantages of rule induction systems is generation of human comprehensible and simple models. On the other hand most of research in the field of inductive ML emphasizes high correctness of the generated models, in terms of high accuracy, and sometimes also low computational complexity [14] [16] [32]. The issue of comprehensibility is often neglected, and very rarely reported in the literature. One of very important dimensions for rule comprehension is complexity expressed by the number and length of generated rules. As a simple quantitative measure of rule complexity, researchers report total number of selectors (logical conditions) for each rule set [10] [31] [32] [39]. Rule complexity has implications for ease of rule interpretation by domain experts, ease of implementing the rules via computer programming or querying language, and for ease of explanation to non-technical decision makers [39].

Obviously, having two rule based models that are characterized by the same accuracy, the end user would prefer the less complex models among them. There are two main reasons for that preference:

1. Less complex model is easier to analyze and understand since it is more compact.
2. Less complex model is better generalized than more complex model, assuming that they both achieve comparable accuracy. This difference is most clearly seen if both models are of the same type. In this case the more complex model contains additional information, over the less complex model, which does not improve its quality, and thus should be removed.

Often the complexity of the model relates to overfitting, where learning method generates data models that agree too closely with, or in other words perfectly imitate, the input data. This is done with expense of generalization on unseen, test data, and results in either lowering the accuracy or generation of very complex models. Other properties of input data, such as inconsistencies and noise, may also result in overfitting.

To this end, this paper proposes a novel framework that addresses the issue of complexity reduction. The framework can be used to reduce complexity of rule based models generated by many modern rule induction systems. The proposed framework is based on a Meta Mining concept, explained later, and provides a solution for the situations where low complexity models are necessary, e.g. in case of designing medical support systems that need to be manually evaluated by physicians.

Following, first necessary background, and architecture of the proposed framework are described. Next, the framework is applied on a set of benchmarking datasets showing that for some rule induction systems reduction of complexity by as much as 50% is possible, virtually without any loss of accuracy. We end the paper with summary and conclusions.

1.1. Meta Mining

The proposed framework uses a Meta Mining (MM) concept, which is related to higher order data mining. Its main characteristic is generation of data models, called meta-models (often meta-rules), from the already generated data models (usually rules, called meta-data) [41] [43]. In general, MM based systems work in two main steps. First, they divide the input data into subsets and generate a data model for each subset. Next, they take the generated data models and generate the meta-model from them. There are several advantages to using MM:

- MM based systems generate results from already mined data models, and therefore the generated results are different, and often more compact, than results of regular mining. Some researchers argue that meta results provide more interesting knowledge than regular mining systems [1] [31] [43].
- Improved scalability. The MM based systems analyze many small datasets (i.e. subsets of original data, and the data models) instead of one large input dataset. This results in reduction of computational time for worse-than-linear rule induction systems, and most of all in ability to implement them in parallel or distributed fashion [31].

The MM often applies the same base-learner (rule induction system) on the data to produce a hypothesis, but performs it in two steps where the outcome is generated from results of the first step [31]. In contrast, the meta-learning aims to discover the best learning strategy through continuing adaptation of the learning algorithms at different levels of abstraction, like for example through dynamic selection of bias [44]. We note that MM already has found applications, such as in association rule generation [1] [40].

1.2. Rule Induction Systems

The proposed framework applies standard rule induction systems in the MM setting. Therefore, a short survey of relevant systems follows.

In general rule induction systems can be divided into rule learners, decision tree learners, and their hybrids [7]. Rule learners are distinct from decision trees since they use different induction techniques. The decision tree learners primarily produce a decision tree, which is used to extract the rules, while rule learners generate the rules directly. The direct generation of rules results in independent rules, i.e. the rules are not biased towards sharing selectors. In case of decision trees, rules are generated by traversing a tree from a leaf to a root node. One rule is generated for every leaf node. This implies that different rules share the same selectors, as illustrated in Figure 1.

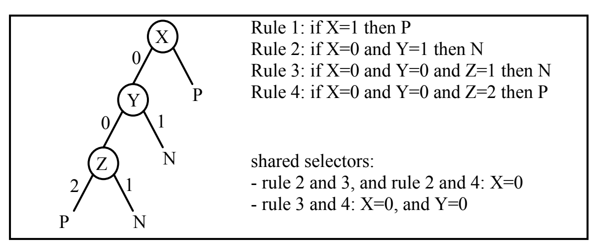

Figure 1. Rule generation using decision tree learners

Example decision tree learners are CART [4], C4.5 [38], T1 [24], and C5.0 [42]. Example rule learners are the AQ family of algorithms [25], FOIL [37], REP [12], IREP [19], RISE [17], RIPPER [14] [15], SLIPPER [16], LAD [3], LERILS [6], and DataSqueezer [28] [31] [32]. Hybrid learners are represented by the CN2 [11], and CLIP family of algorithms [8] [10]. A survey of relevant inductive ML learners can be found in [20].

1.3. Other Relevant Work

The proposed framework aims to reduce complexity of generated rule based models. This issue was usually tackled by researchers who develop a new standalone learner that provides smaller set of rules when compared with other learners. A good comparative paper was published by Lim et al. [33].

Two other related works were published. A method for simplifying comprehension of rules by deriving a new representation was proposed. The resultant representation was in terms of directed acyclic graph, but did not provide any complexity reduction [21]. An alternative approach for complexity reduction is to reduce the size of the input data. It was observed that decision tree learners generate much larger and no more accurate trees when using the entire available training data, when compared to trees generated by using a small subset of the training instances. Given the discovered strong relationship between tree size and training set size, the authors claim that any technique that removes training instances prior to tree construction could result in smaller trees [34]. We note that latter work concerns only decision tree learners, while the rule and hybrid learners were not considered.

2. Proposed Framework

There are three main requirements for application of a rule induction system within the proposed framework:

1. Rule induction system has to be able to cope with large amount of missing data. This is necessary to generate the meta-rules in the second step of a MM procedure; see requirement #3 for explanation.
2. Rule induction system should not apply so called default hypothesis. Some learners, such as RIPPER, SLIPPER, and C5.0 apply default hypothesis. In this case, examples that are not covered by any rule are assigned to the class with the highest frequency in the training dataset (default class). This means that each example is always classified. On the other hand for highly skewed datasets, where one of the classes is in significant majority, it may lead to generation of the default hypothesis as the only "artificial" rule. This, in turn, may results in inability to execute the second step of a MM procedure.
3. Each rule generated by a rule induction system should involve no more than one selector (logical condition) per attribute. Some learners, such as CLIP, generate rule with multiple selectors per one attribute, while decision tree learners and for example DataSqueezer learner satisfy this condition. This property allows storing generated rules (meta-data) in a table that has identical structure as the original data table. This, in turn, allows using the same learner in the second step of the MM procedure. For example, for data described by attributes A, B, C, and D, and describing two classes 1, and 2, the following rule can be generated: IF A=1 and C=1 THEN 2. The rule can be written as (1, *, 1, *, 2), following the format of the input data, which defines values of attributes A,B, C, and D, and adds class attribute as the last attribute. The "*" symbol stands for a missing value (attributes B and D were not used in the rule).

We also note that if decision trees were to be used in a MM setting, the generated meta-rules can be possibly biased towards the shared selectors. Therefore, using decision trees may have some disadvantages when compared to rule learners, which generate independent rules. We note that there are several learners that satisfy requirements of the proposed framework, such as C4.5 and all other decision tree learners, C5.0 for all non-skewed datasets, RIPPER, SLIPPER, and DataSqueezer.

With the above assumptions in mind, details of the proposed framework are explained. The architecture of the proposed framework is shown in Figure 2. The framework applies a rule induction system within a MM setting. Meta-rules are generated from input supervised data in these two steps:

1. Data Mining step.

- First, the input data is divided into subsets. Selection of the proper number of subsets depends on the input data size. The number of subsets usually should be relatively small, so that the size of each input subset would allow generation of rule sets for each of the classes. In case of a large number of small subsets, the rule induction system may generate inaccurate rules because the amount of examples would be too small to generate correct data model. The simplest way to divide input data is to split it randomly. The input data can be also divided in a predefined way. For example, in case of temporal data, it can be divided into subsets corresponding to different time intervals, as it was performed in [30].
- Second, rules are generated for data in each of the subsets by a rule induction system.
- Third, a rule table, which stores the generated rules (meta-data) in a format that is identical to the format

of the original input data, is created from the generated rules.

2. Meta Mining step

- Meta-rules are generated using the rule table are the input. The meta-rules are generated by the same rule induction system, which was used in the Data Mining step. The meta-rules describe the most important patterns associated with the target concept over the entire original input dataset

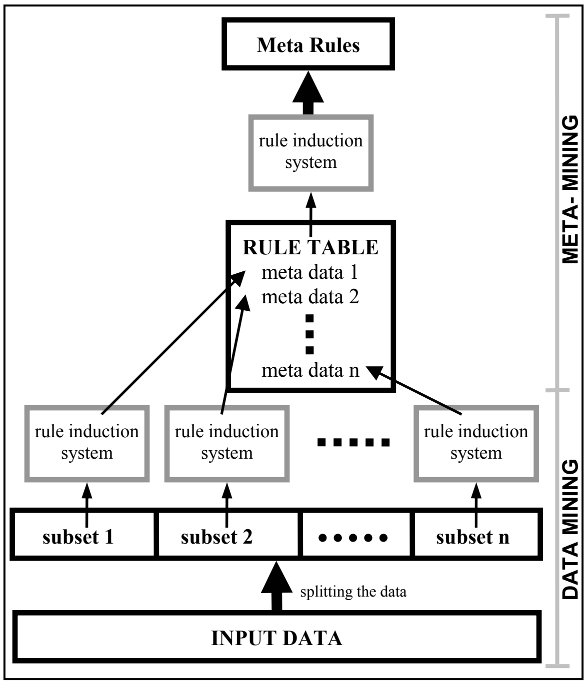

Figure 2. Architecture of the proposed framework

The idea of splitting the dataset, learning a rule set on each split, and combining the results has been previously introduced [18], but the proposed framework is the first that combines the rule sets in a separate, second learning phase. The proposed framework was previously used with the DataSqueezer learner to propose a MetaSqueezer learning system [31].

Next, a set of benchmarking tests that verify usefulness of the proposed framework is presented. The tests aim to show the complexity reduction when using the proposed framework vs. using a standalone rule induction system. At the same time accuracy of the generated rules is also recorded and compared.

3. Experiments

The proposed framework was tested with two rule induction systems: a rule learner DataSqueezer, and a decision tree learner C5.0. The test were performed using six publicly available benchmarking datasets, characterized by the size of training datasets between 600 and 200K examples, the size of testing datasets between 1190 and 100K examples, the number of attributes between 7 and 61, and the number of classes between 2 and 10. They constitute a set of larger datasets selected from the UCI ML repository [2]. The testing uses a single split of the input data into training and testing subsets, which is identical to the data submitted by the owners (donors). We note that 10 fold cross validation experiments will be performed, as a future work, to validate of the presented results. The datasets were randomly divided into a number of equal size subsets, depending on the size of the data, to be used as an input to the DM step. All continuous attribute were discretized by a supervised discretization algorithm CAIM [26] [27] [29] before they were applied in the tests. The main reason is that the DataSqueezer learner receives only discrete numerical or nominal data as its input. To make the comparison fair, the C5.0 also uses the discretized data. The complete description of the datasets is shown in Table 1.

Table 1. Major properties of datasets considered in the benchmarking tests

Properties	Datasets					
	cid	dna	led	sat	thy	wav
# classes	2	3	10	6	3	3
# examples	299285	3190	6000	6435	7200	3600
# test examples	99762	1190	4000	2000	3428	3000
# attributes	40	61	7	37	21	21
# subsets	10	8	10	10	6	3

The results, in term of accuracy of the rules, and the number of rules and selectors, are compared between the standalone learner, and the learner used in the proposed framework. The DataSqueezer within the framework is called MetaSqueezer, while the C5.0 is called MetaC5.0. In addition, sensitivity and specificity are also reported. These are a standard used in medicine where sensitivity and specificity analysis is used to evaluate confidence in the results [9]. For multi-class problems, the sensitivity and specificity are computed for each class separately, and the average values are reported. Table 2 shows the accuracy, sensitivity, and specificity comparison for both DataSqueezer, and C5.0 learners, while Table 3 shows comparison in terms of the rule complexity. The latter is measured in terms of the number of rules, the numbers of selectors, and the number of selectors per single rule.

Table 2. Comparison of accuracy, sensitivity, and specificity for the benchmarking tests

set	Reported			DataSqueezer			MetaSqueezer			C5.0			MetaC5.0		
	max	min	ref.	accuracy	sensitiv.	specific.	accuracy	sensitiv.	specific.	accuracy	sensitiv.	specific.	accuracy	sensitiv.	specific.
dna	95	62	[33]	92	92	97	90	89	95	94	94	97	53	33	67
led	73	18	[33]	68	68	97	69	69	97	74	74	97	62	61	96
sat	90	60	[33]	80	78	96	74	73	95	85	82	97	42	48	88
thy	99	11	[33]	96	95	99	96	86	99	98	89	98	98	86	99
wav	85	52	[33]	77	77	89	77	76	89	75	75	88	52	51	76
cid	95	77	[22]	91	94	45	90	93	49	95	99	32	94	99	17
MEAN	89.5	46.7	---	84.0	84.0	87.2	82.7	81.0	87.3	86.8	85.5	84.8	66.8	63.0	73.8

Table 3. Comparison of rule complexity for the benchmarking tests

set	Reported		DataSqueezer			MetaSqueezer			C5.0			MetaC5.0		
	median # of rules	ref.	# rules	# select	# select / rule	# rules	# select	# select / rule	# rules	# select	# select / rule	# rules	# select	# select / rule
dna	13	[33]	39.0	97.0	2.5	34.0	53.0	1.6	40	107	2.7	0	0	0
led	24	[33]	51	194	3.8	51	141	2.8	20	79	4.0	10	21	2.1
sat	63	[33]	57	257	4.5	55	104	1.9	118	413	3.5	4	4	1.0
thy	12	[33]	7	28	4.0	6	6	1.0	5	9	1.8	4	4	1.0
wav	16	[33]	22	65	2.9	17	18	1.0	37	128	3.5	2	2	1.0
cid	---	---	15	95	6.3	6	34	5.7	146	412	2.8	14	14	1.0
MEAN	---	---	31.8	122.7	4.0	28.2	59.3	2.3	61.0	191.3	3.1	5.7	7.5	1.3

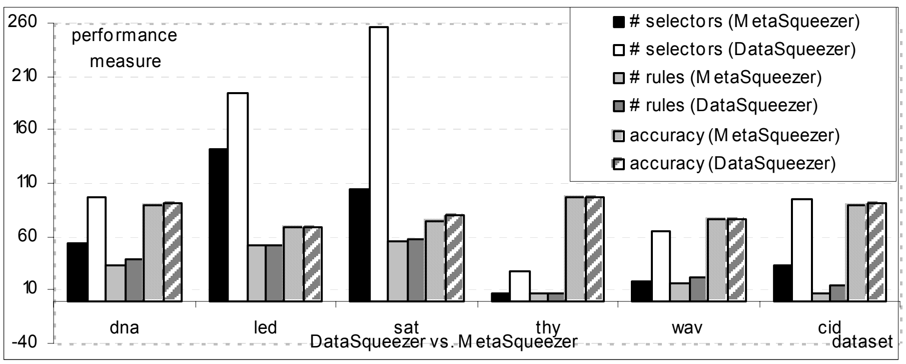

Figure 3. Side-by-side comparison between DataSqueezer and MetaSqueezer induction system

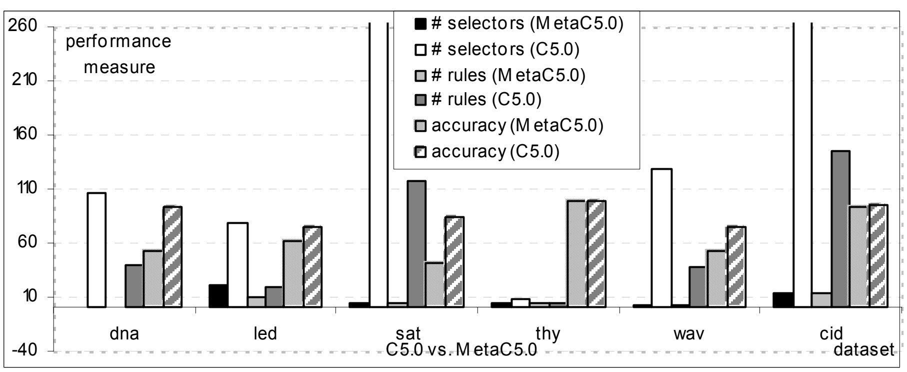

Figure 4. Side-by-side comparison between C5.0 and MetaC5.0 induction system

To easy the analysis, the results are also presented in Figure 3 and Figure 4. Figure 3 shows side-by-side comparison between DataSqueezer and MetaSqueezer, while Figure 4 shows the comparison between C5.0 and MetaC5.0 induction system. The results, in case of both MetaSqueezer and MetaC5.0, show significant reduction of rule complexity when compared with standalone learners. The 12% reduction in the number of rules and 52% reduction in the number of selectors were achieved by MetaSqueezer. This significant complexity reduction was achieved with only marginal loss of accuracy by 1.3%. The loss of accuracy was caused by 3% loss of sensitivity, and without any loss in terms of specificity. The 91% reduction in the number of rules and 96% reduction in the number of selector were achieved by the MetaSqueezer. Although a very significant complexity reduction was achieved, it came with large costs. The 20% loss of accuracy, 22.5% loss of sensitivity, and 11% loss of specificity resulted from applying the proposed framework. The main reasons for the weak performance of MetaC5.0 are generation of default hypothesis, as in case of *dna* dataset (zeros in Table 3 mean that only the default hypothesis was generated), problems with handling large amounts of missing data [32], and rule dependency that resulted in generation of too compact rule sets. Despite these flaws, for some datasets where underlying data models rely on only a few attributes, such as *thy*, *led*, and *cid*, the results are encouraging.

To summarize, the benchmarking results show that the proposed framework in tandem with some of rule induction systems can provide significant rule complexity reduction. Most importantly, we expect that for some learners, especially those that satisfy requirements stated in section 2, the rule complexity reduction will come with virtually no loss of accuracy, as it is shown in case of the DataSqueezer learner.

4. Summary and Conclusions

The paper introduces a novel framework that aims to reduce complexity of rule base models generated by rule induction systems. The framework is situated within the state-of-the-art in the inductive ML field. While many standalone ML learners generate rules directly from input data, the framework is based on the Meta Mining concept and generates meta-rules from previously generated meta-data. The main benefit of the reduced complexity is increased comprehension of the generated knowledge. Application of the Meta Mining concept also results in ability to successfully devise efficient parallel or distributed DM solutions to the rule induction process. The framework works by merging partial knowledge extracted from subsets of data, which can be computed in parallel by distinct serial programs on (possibly disjoint) subsets of the training data set [36]. The biggest advantage of the described system is that the application of the distributed approach to the rule generation does not result in penalizing accuracy of the results.

The framework was tested on a set of larger size, publicly available benchmarking datasets showing very promising results. Using the framework with a rule learner reduction by over 50% of rule complexity was achieved with virtually no penalty on the accuracy of the generated model. Application of the framework with a decision tree learner resulted in reduction by over 90%, but with a 20% loss of accuracy. A set of requirements, stated in section 2, was developed to select rule induction systems that are suitable to be applied within the proposed framework. Since the tested decision tree learner did not adhere to some of these requirements, the complexity reduction resulted in the accuracy loss. We expect that learners adhering to the requirements, such as DataSqueezer, will benefit the complexity reduction with only marginal loss of accuracy.

We note that this work shows preliminary results on a limited number of datasets. Future work will include more comprehensive analysis, which will accommodate more datasets, and incorporate cross validation experiments and t-test based comparison between relevant methods, to validate the presented results. We also note that although the proposed framework gives a desirable solution to problem of rule complexity, many questions remain open. Psychological studies of the nature of comprehensibility of generated knowledge structures are necessary to give substance to the intuitions that lie behind the work reported in this paper. The analysis to determine whether the meta-rules or rules generated by the standard inductive ML learners make sense to the user is outside of the scope of this paper.

Acknowledgements

This research was supported in part by the Natural Sciences & Engineering Research Council (NSERC) of Canada under grant number G121210953.

References

[1] Abraham, T., and Roddick, J. F., Incremental Meta-mining from Large Temporal Data Sets, Advances in Database Technologies, *Proceedings of the 1st International Workshop on Data Warehousing and Data Mining*, pp.41-54, 1999

[2] Blake, C.L., and Merz, C.J., 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, 1998

[3] Boros, E., Hammer, P.L., Ibaraki, T., Kogan, A., Mayoraz, E., and Muchnik, I., An Implementation of Logical Analysis of Data, *IEEE Transactions on Knowledge and Data Engineering*, 12:2, pp.292-306, 2000

[4] Breiman, L., Friedman, J., Olshen, R., and Stone C., *Classification and Regression Trees*, Chapman and Hall, 1984

[5] Catlett, J., Megainduction: a Test Flight, *Proceedings of the 8th International Workshop on Machine Learning*, Ithaca, NY, pp.596-599, 1991

[6] Chisholm, M., and Tadepalli., P., Learning Decision Rules by Randomized Iterative Local Search. *Proceedings of the International Conference on Machine Learning*, pp.75-82, 2002

[7] Cios, K.J., Pedrycz, W., and Swiniarski, R., *Data Mining Methods for Knowledge Discovery*, Kluwer, 1998

[8] Cios K.J. and Kurgan, L.A., Hybrid Inductive Machine Learning: An Overview of CLIP Algorithms, In: Jain L.C., and

Kacprzyk J. (Eds.) *New Learning Paradigms in Soft Computing*, Physica-Verlag (Springer), pp.276-322, 2001

[9] Cios, K.J., and Moore, G., Uniqueness of Medical Data Mining, *Artificial Intelligence in Medicine*, 26:1-2, pp.1-24, 2002

[10] Cios, K.J., and Kurgan, L.A., CLIP4: Hybrid Inductive Machine Learning Algorithm that Generates Inequality Rules, *Information Sciences*, Special Issue on *Soft Computing Data Mining*, Pal, S.K., & Ghosh, A. (Eds.), vol.163, no.1-3, pp. 37-83, 2004

[11] Clark, P., and Niblett, T., The CN2 Algorithm, *Machine Learning*, 3, pp.261 283, 1989

[12] Cohen, W., Efficient Pruning Methods for Separate-and-Conquer Rule Learning Systems, *Proceedings of the 13th International Joint Conference on Artificial Intelligence*, Chambery, France, pp. 988-994, 1993

[13] Cohen, W., Grammatically Biased Learning: Learning Logic Programs Using an Explicit Antecedent Description Language, *Artificial Intelligence*, 68, pp. 303-366, 1994

[14] Cohen, W., Fast Effective Rule Induction, *Proceedings of the 12th International Conference on Machine Learning*, pp.115-123, Lake Tahoe, CA, 1995

[15] Cohen, W., Learning Trees and Rules with Set-valued Features, *Proceedings of the 13th National Conference on Artificial Intelligence*, pp.709-716, Portland, OR, 1996

[16] Cohen, W., and Singer, Y., A Simple, Fast and Effective Rule Learner, *Proceedings of the 16th National Conference on Artificial Intelligence*, pp.335-342, 1999

[17] Domingos, P., The RISE System: Conquering Without Separating, *Proceedings of the 6th IEEE International Conference on Tools with Artificial Intelligence*, New Orleans, LA, pp.704-707, 1994

[18] Domingos, P., Using Partitioning to Speed up Specific-to-General Rule Induction, *Proceedings of the AAAI-96 Workshop on Integrating Multiple Learned Models for Improving and Scaling Machine Learning Algorithms*, Portland, OR, 1996

[19] Furnkranz, J., and Widmer, G., Incremental Reduced Error Pruning, *Machine Learning: Proceedings of the 11th Annual Conference*, New Brunswick, New Jersey, 1994

[20] Furnkranz, J. Separate-and-conquer Rule Learning, *Artificial Intelligence Review*, 13:1, pp. 3-54, 1999

[21] Gaines, B., Transforming Rules and Trees into Comprehensible Knowledge Structures, in *Advances in Knowledge Discovery and Data Mining*, U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, (Eds.) Menlo Park, CA: AAAI, 1996

[22] Hettich, S., and Bay, S. D., The UCI KDD Archive, http://kdd.ics.uci.edu, Irvine, CA: University of California, Department of Information and Computer Science, 1999

[23] Holsheimer, M., and Siebes, A.P., *Data Mining: the Search for Knowledge in Databases*, Technical report CS-R9406, 1994

[24] Holte, R.C., Very Simple Classification Rules Perform Well on Most Commonly Used Datasets, *Machine Learning*, 11, pp.63-90, 1993

[25] Kaufman, K.A., and Michalski, R.S., Learning from Inconsistent and Noisy Data: The AQ18 Approach, *Proceedings of the 11th International Symposium on Methodologies for Intelligent Systems*, Warsaw, Poland, pp.411-419, 1999

[26] Kurgan, L.A., and Cios, K.J., Discretization Algorithm that Uses Class-Attribute Interdependence Maximization, *Proceedings of the 2001 International Conference on Artificial Intelligence*, Las Vegas, Nevada, pp.980-987, 2001

[27] Kurgan, L.A., and Cios K.J., Fast Class-Attribute Interdependence Maximization (CAIM) Discretization Algorithm, *Proceedings of the 2003 International Conference on Machine Learning and Applications (ICMLA'03)*, Los Angeles, pp.30-36, 2003

[28] Kurgan, L.A., *Meta Mining System for Supervised Learning*, Ph.D dissertation, the University of Colorado at Boulder, Department of Computer Science, 2003

[29] Kurgan, L.A., and Cios, K.J., CAIM Discretization Algorithm, *IEEE Transactions of Knowledge and Data Engineering*, 16:2, pp.145-153, 2004

[30] Kurgan, L.A., Cios, K.J., Sontag, M., and Accurso, F.J., Mining the Cystic Fibrosis Data, In: Zurada, J., and Kantardzic, M. (Eds.), *Novel Applications in Data Mining*, IEEE Press, in print, 2004

[31] Kurgan, L.A., and Cios, K.J., Meta Mining Architecture for Supervised Learning, *Proceedings of the 7th International Workshop on High Performance and Distributed Mining*, in conjunction with the *4th International SIAM Conference on Data Mining*, pp.18-26, Lake Buena Vista, FL, 2004

[32] Kurgan, LA.., Cios, K.J., and Dick S., Highly Scalable and Robust Rule Learner: Performance Evaluation and Comparison, *IEEE Transactions on Systems, Man and Cybernetics – Part B*, in review, 2004

[33] Lim, T.-S., Loh, W.-Y., and Y.-S. Shih, A Comparison of Prediction Accuracy, Complexity, and Training Time of Thirty-three Old and New Classification Algorithms, *Machine Learning*, 40, pp.203-228, 2000

[34] Oates, T., and Jensen, D. The Effects of Training Set Size on Decision Tree Complexity, *Proceedings of the 14th International Conference on Machine Learning*, pp.254-262, 1997

[35] Pazzani, M., and Kibler, D., The Utility of Knowledge in Inductive Learning, *Machine Learning*, 9:1, pp.57-94, 1992

[36] Prodromidis, A., Chan, P., and Stolfo, S., Meta-Learning in Distributed Data Mining Systems: Issues and Approaches, In: Kargupta H. & Chan P. (Eds.), *Advances in Distributed and Parallel Knowledge Discovery*, AAAI/MITPress, 2000

[37] Quinlan, J.R., Learning Logical Definitions from Relations, *Machine Learning*, 5, pp.239-266, 1990

[38] Quinlan, J.R., *C4.5 Programs for Machine Learning*, Morgan-Kaufmann, 1993

[39] Rajagopalan, B, and Isken, M.W., Exploiting Data Preparation to Enhance Mining and Knowledge Discovery, *IEEE Transactions on Systems, Man and Cybernetics, Part C*, 31:4, pp.460-467, 2001

[40] Rainsford, C.P., and Roddick, J.F., Adding Temporal Semantics to Association Rules, *Proceeding of the 3rd European Conference on Principles and Practice of Knowledge Discovery in Databases*, pp.504-509, 1999

[41] Roddick, J.F., and Spiliopoulou, M., A Survey of Temporal Knowledge Discovery Paradigms and Methods, *IEEE Transactions on Knowledge and Data Engineering*, 14:4, pp.750-767, 2002

[42] RuleQuest Research, C5.0 decision tree system, http://www.rulequest.com/see5-info.html, 2003

[43] Spiliopoulou, M., and Roddick, J. F., Higher Order Mining: Modelling and Mining the Results of Knowledge Discovery, *Data Mining I: Proceedings of the Second International Conference on Data Mining Methods and Databases*, pp.309-320, 2000

[44] Vilalta, R., and Drissi, Y., A Perspective View and Survey of Meta-Learning, *Artificial Intelligence Review*, 18:2, pp.77-95, 2002