Complexity Control in Rule Based Models for Classification in Machine Learning Context

Han Liu[1], Alexander Gegov[1] and Mihaela Cocea[1]

Abstract A rule based model is a special type of computational models, which can be built by using expert knowledge or learning from real data. In this context, rule based modelling approaches can be divided into two categories: expert based approaches and data based approaches. Due to the vast and rapid increase in data, the latter approach has become increasingly popular for building rule based models. In machine learning context, rule based models can be evaluated in three main dimensions, namely accuracy, efficiency and interpretability. All these dimensions are usually affected by the key characteristic of a rule based model which is typically referred to as model complexity. This paper focuses on theoretical and empirical analysis of complexity of rule based models, especially for classification tasks. In particular, the significance of model complexity is argued and a list of impact factors against the complexity are identified. This paper also proposes several techniques for effective control of model complexity, and experimental studies are reported for presentation and discussion of results in order to analyze critically and comparatively the extent to which the proposed techniques are effective in control of model complexity.

Keywords: Machine Learning, Rule Based Models, Model Complexity, Complexity Control, Rule Based Classification

[1] Han Liu

School of Computing, University of Portsmouth, Buckingham Building, Lion Terrace, Portsmouth, PO1 3HE, United Kingdom, Email: Han.Liu@port.ac.uk

Alexander Gegov,

School of Computing, University of Portsmouth, Buckingham Building, Lion Terrace, Portsmouth, PO1 3HE, United Kingdom, Email: Alexander.Gegov@port.ac.uk

Mihaela Cocea

School of Computing, University of Portsmouth, Buckingham Building, Lion Terrace, Portsmouth, PO1 3HE, United Kingdom, Email: Mihaela.Cocea@port.ac.uk

1 Introduction

A rule based model is a special type of computational models, which can be used for the purpose of knowledge discovery and predictive modelling. A rule based model consists of a set of rules, which can be built by using expert knowledge or by learning from real data. From this point of view, rule based modelling approaches can be categorized into expert based approaches and data based approaches. Due to the vast and rapid increase in data, the latter approach of modelling has become increasingly popular. The data based approach typically involves learning of rules for building of rule based models. In practice, rule based models can be used for different tasks such as classification, regression and association. From this point of view, rules extracted from a rule based model can be categorized into classification rules, regression rules and association rules. Both classification and regression rules can be viewed as a special type of association rules, due to the fact that these two types of rules represent the relationship between multiple independent variables and a single dependent variable, whereas association rules represent the relationship between multiple independent variables and multiple dependent variables. The main difference between classification rules and regression rules is that the output attribute on the right hand side must be discrete for the former and continuous for the latter [1]. Therefore, classification rules are generally used for categorical predictions whereas regression rules are used for numerical predictions.

In machine learning research, rule based models can be evaluated in three main dimensions namely accuracy, efficiency and interpretability. One of the important characteristics of rule based models is referred to as model complexity, which usually impacts on the above three main dimensions. As described in [2], complex models are usually less generalized than simple models, which are likely to result in overfitting. This problem typically results in loss of accuracy for predictive modelling and decrease of the level of reliability for knowledge extracted from a rule based model. On the other hand, as analyzed in [3], complex models usually lead to less efficient prediction on test instances and poor interpretability to people for the purpose of knowledge discovery. On the basis of the above description, this paper focuses on theoretical and empirical analysis of complexity of rule based models and how the complexity can be controlled effectively.

The rest of this paper is organized as follows: Section 2 argues why complexity control is important for rule based models to be applied in real world. Section 3 identifies a list of impact factors that affect complexity of rule based models as well as analyze in depth how these identified factors impact on model complexity. Section 4 introduces two main techniques, namely scaling up algorithms and scaling down data, towards effective complexity control in rule based models. In particular, scaling up algorithms involves proper use of statistical heuristics for rule generation and effective assistance from rule simplification, and scaling down data involves effective pre-processing of training data, which includes feature selection, feature extraction and attribute discretization. Section 5 describes setup of the experimental

studies, and results are presented and discussed critically and comparatively in order to show the extent to which the techniques used for complexity reduction are effective. Section 6 summaries the contributions of this paper and provides some suggestions for further directions towards advances in this research area.

2 Significance of Complexity Control

As mentioned in Section 1, model complexity usually impacts on accuracy, efficiency and interpretability. This section justifies why it is important to effectively control the complexity of a rule based model.

As mentioned in [3], rule based models can be used in practice for the purpose of knowledge discovery and predictive modelling. For the latter purpose, rule based models are used in a black box manner, which means that the emphasis is on the mapping from inputs to outputs without interpretation of the reasons, i.e. to predict the values of the outputs on the basis of the values of the inputs. In this context, rule based models need to be both accurate and efficient in predicting unseen instances. On the other hand, for the purpose of knowledge discovery, rule based models are used in a white box manner which should allow the interpretation of the reasons for the mapping. In this context, rule based models need to be both accurate and interpretable for people to use knowledge extracted from the models, i.e. to see a list of causal relationships by going through a set of rules. On the basis of the above description, model complexity can have a significant impact on the accuracy, efficiency and interpretability of rule based models.

In terms of accuracy, on the basis of the same data, more complex models usually have lower generality than simpler models, which are likely to result in models performing well on the training data but poorly on the testing data. The above case is commonly known as overfitting. As mentioned in [2], one of the biases that arise with rule based models is referred to as overfitting avoidance bias [4, 5], which means that rule learning algorithms prefer simpler rules to more complex rules under the expectation that the accuracy on the training data is lower but that on the testing data it would be higher.

In terms of efficiency, more complex models are usually less efficient than simpler models in predicting unseen instances. This is because of the fact that predictions by a rule based model are made through checking the rules extracted from the model [3]. In this context, a model that consists of a large number of rule terms is considered as a complex model whereas a model that is made up of a small number of rule terms is considered as a simple model. In the worst case, it always takes longer to make a prediction using a more complex model than using a simpler model, if the two models are represented in the same structure [3]. Section 3 will give more details on complexity analysis in terms of rule representation.

In terms of interpretability, more complex models are usually less interpretable for people to read and understand knowledge extracted from the rule based models.

This is because of the fact that people need to read each of the rules extracted from a particular model in order to see any causal relationships between the inputs and the outputs. In this context, a model that consists of a large number of complex rules is considered as a complex model whereas a model that is made up of a small number of simple rules is considered as a simple model. In other words, a model that consists of a large number of complex rules is like an article that is made up of a large number of long paragraphs, which usually makes it difficult and cumbersome for people to follow.

On the basis of the above description, model complexity needs to be considered as an important impact on accuracy, efficiency and interpretability, and thus needs to be controlled effectively.

3 Impact Factors for Model Complexity

Section 2 justified the significance of complexity control for rule based models towards generation of accurate, efficient and interpretable models in practice. This section identifies a list of impact factors for model complexity and justifies how these factors would affect the complexity of rule based models. In particular, the strategy involved in a learning algorithm and the characteristic of a data set are viewed as two main impact factors as already identified in [6]. Also, ways to impact on the model complexity are analyzed in the context of rule based classification.

3.1 Learning Strategy

In terms of learning algorithms, the strategy of rule generation usually significantly affects the model complexity. As mentioned in [7, 8], the generation of classification rules can be divided into two categories: 'divide and conquer' [9] and 'separate and conquer' [2]. The former is also referred to as Top-Down Induction of Decision Trees (TDIDT) due to the fact that this learning approach aims to generate classification rules in the form of a decision tree. The latter is also referred to as covering approach because of the fact that this approach aims to learn a set of if-then rules sequentially, each of which covers a subset of training instances that are normally removed from the current training set prior to the generation of the next rule.

As introduced in [10, 11], Prism, which is a rule induction method that follows the 'Separate and Conquer' approach, is likely to generate fewer and more general rules than ID3, which is another rule induction method that follows the 'Divide and Conquer' approach. The above phenomenon is due mainly to the strategy of rule learning. As mentioned in [10], the rule set generated by the TDIDT needs to have at least one common attribute in order to be represented in the form of a decision tree. The same also applies to each of the subtrees of a decision tree, which requires

to have at least one common attribute represented as the root of the subtree. Due to this requirement, the TDIDT is likely to generate a large number of complex rules with many redundant terms such as the replicated subtree problem [10] illustrated in Fig.1 and thus results in a model with high complexity.

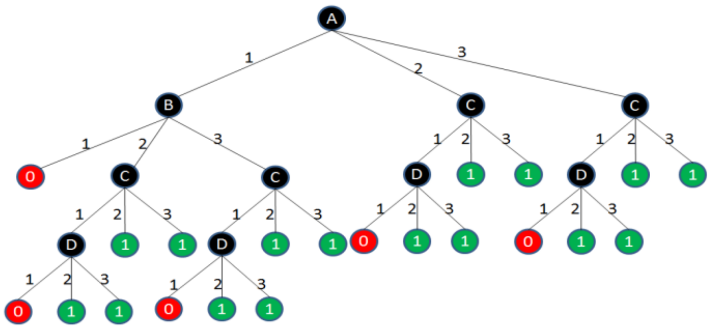

Fig. 1. Cendrowska's replicated subtree example [3]

3.2 Data Characteristics

As mentioned in Section 3.1, different algorithms involve different strategies of learning and thus generate rule based models with different levels of complexity. In this sense, when the same data set is used, different learning algorithms would usually lead to different levels of model complexity. However, for the same algorithm, data of different size would also usually result in the generation of models with different levels of complexity. The rest of this subsection justifies the potential correlation between data size and model complexity.

As mentioned earlier, rule learning methods involve the generation of rule based models. The complexity of a rule based model is determined by the total number of rule terms, which is dependent upon the number of rules and the average number of terms per rule. However, the total number of rule terms is also affected by the data size in terms of both dimensionality (number of attributes) and sample size (number of instances). For example, a data set has n attributes, each of which has t values, and its sample contains m instances and covers all possible values for each of the attributes. In this example, the model complexity would be equal to $\Sigma\, t^{i}$, for i=*0*, *1*, *2*…n, in principle, but no greater than $m \times n$ in the worst case in practice. This indicates that a rule based model consists of a default rule, which is also referred to as the 'else' rule, and t^{i} rules, each of which has i terms, for i=*0*, *1*, *2*…n respectively. However, each rule usually covers more than one instance and the rule based model is expected to cover all instances. Therefore, the number of rules from a rule based model is usually less than the number of instances from a data set. As also justified above, each rule would have up to n (the number of attributes) terms due

to the requirement that each attribute can only appear once comprising one of its possible values in any of the rules. On the basis of the above description, the complexity of a rule based model is up to the product of dimensionality and sample size of a data set. In addition, the complexity of each attribute also impacts on the complexity of the rule based model, especially for continuous attributes.

4 Techniques for Control of Model Complexity

Section 3 identified two main impact factors for model complexity – learning algorithms and data characteristics, and analyzed theoretically in what way the two factors impact on the complexity of a rule based model. This section presents several techniques towards effective control of model complexity. In particular, these techniques follow one of the two approaches namely scaling up algorithms and scaling down data.

4.1 Scaling Up Algorithms

As introduced in [6], scaling up algorithms for complexity reduction can be achieved through proper employment of rule generation methods or proper use of rule pruning algorithms.

In terms of rule generation, the learning approaches can be categorized into divide and conquer and separate and conquer as mentioned in Section 3.1. In particular, examples of the divide and conquer approach include ID3 [12] and C4.5 [9] and examples of the separate and conquer approach include IEBRG [7] and Prism [10]. ID3 and IEBRG both involve use of information entropy for generation of rules but with different strategies resulting in rule based models being represented in different forms and having different levels of complexity. The illustration of these methods are presented below using the contact-lenses data set [10] retrieved from the UCI repository [13].

As mentioned in [14], ID3 makes attribute selection based on average entropy, i.e. ID3 is an attribute oriented learning method and the calculation of entropy is for a whole attribute on average. In addition, IEBRG makes selection of attribute-value pairs based on conditional entropy, i.e. IEBRG is an attribute-value oriented learning method and the calculation of entropy is for a particular value of an attribute. For each of the methods, the detailed illustration can be seen in [14]. As mentioned in [14], ID3 makes attribute selection based on average entropy, i.e. ID3 is an attribute oriented learning method and the calculation of entropy is for a whole attribute on average. In addition, IEBRG makes selection of attribute-value pairs based on conditional entropy, i.e. IEBRG is an attribute-value oriented learning method

and the calculation of entropy is for a particular value of an attribute. For each of the methods, the detailed illustration can be seen in [14].

Table 1. Contact-lenses Data Set [14]

age	prescription	astigmatic	Tear production rate	class
young	myope	no	reduced	no lenses
young	myope	no	normal	soft lenses
young	myope	yes	reduced	no lenses
young	myope	yes	normal	hard lenses
young	hypermetrope	no	reduced	no lenses
young	hypermetrope	no	normal	soft lenses
young	hypermetrope	yes	reduced	no lenses
young	hypermetrope	yes	normal	hard lenses
pre-presbyopic	myope	no	reduced	no lenses
pre-presbyopic	myope	no	normal	soft lenses
pre-presbyopic	myope	yes	reduced	no lenses
pre-presbyopic	myope	yes	normal	hard lenses
pre-presbyopic	hypermetrope	no	reduced	no lenses
pre-presbyopic	hypermetrope	no	normal	soft lenses
pre-presbyopic	hypermetrope	yes	reduced	no lenses
pre-presbyopic	hypermetrope	yes	normal	hard lenses
presbyopic	myope	no	reduced	no lenses
presbyopic	myope	no	normal	soft lenses
presbyopic	myope	yes	reduced	no lenses
presbyopic	myope	yes	normal	hard lenses
presbyopic	hypermetrope	no	reduced	no lenses
presbyopic	hypermetrope	no	normal	soft lenses
presbyopic	hypermetrope	yes	reduced	no lenses
presbyopic	hypermetrope	yes	normal	hard lenses

In accordance with the illustration in [14], the complete decision tree generated is the one as illustrated in Fig.2 and the corresponding if-then rules are represented as follows: *if tear production rate = reduced then class= no lenses*; *if tear production rate = normal and Astigmatic = yes then class= Hard lenses; if tear production rate = normal and Astigmatic= no then class= Soft lenses.*

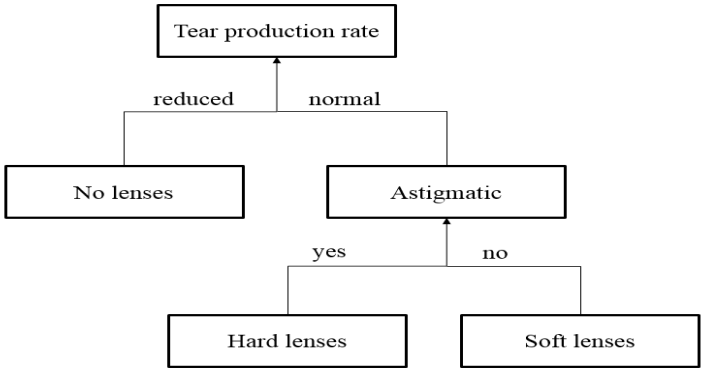

Fig. 2. Complete decision tree

For IEBRG, the first rule generated is also the same as the one represented above: *if tear production rate = reduced then class= no lenses*; this is because the conditional entropy E(*tear production rate= reduced)=0* is the minimum and indicates that there is no uncertainty any more for classifying instances covered by this rule. The same can be seen from Table 1 as the class is always *no lenses*, while *tear production rate = reduced*. All the subsequent rules are generated in the same way as the first rule by appending rule terms on the left hand side of the rule by iteratively selecting the attribute-value pair with the lowest conditional entropy for discriminating different classes. In particular, the rest of the rules generated are the following: *if Astigmatic = yes then class= Hard lenses; if Astigmatic = no then class = Soft lenses.*

It can be seen that ID 3 generates a decision tree which contains 3 rules and 5 terms whereas IEBRG generates a set of if-then rules which contains 3 rules and 3 terms. The difference in the resulted complexity is due to the presence of the redundant term generated (*tear production rate = normal*) in the decision tree illustrated in Fig.2. The essential reason is that the ID3 method is attribute oriented for measuring uncertainty and must have the current training subset split on a particular attribute at each iteration, whereas IEBRG is attribute-value oriented for measuring uncertainty and only needs to separate some instances from the current training subset through selection of a particular attribute-value pair at each iteration. On the basis of the above statement, a rule based model generated by the decision tree learning approach must have at least one common attribute and the same also applies to each of its subtrees. However, a model generated by the if-then rules learning approach does not have such a constraint, which usually results in a lower level of complexity than that of a model generated by the other learning approach. Therefore, it has been recommended in the research literature [2, 10, 14] that the if-then rules learning approach should be used instead of the decision tree learning approach towards generation of simpler rules.

On the other hand, use of pruning algorithms can also manage to reduce the complexity of a rule based model as mentioned above. As introduced in [2], pruning methods can be categorized into pre-pruning and post-pruning.

While decision tree learning methods are used for rule generation, pre-pruning aims to stop a particular branch in a tree growing further whereas post-pruning aims to simplify each of the branches in a tree after the whole tree has been generated. In particular, the tree needs to be converted into a set of if-then rules before the pruning action is taken. In addition, post-pruning can also be done through replacing a subtree with a leaf node. A popular method used for pruning of decision trees is referred to as Reduced Error Pruning (REP) [15] which follows the strategy of post pruning.

While if-then rules learning methods are used for rule generation, pruning is taken per single rule generated in contrast to tree pruning. In other words, each single rule is pruned prior to the generation of the next rule rather than posterior to the completion of the generation of a whole rule set. In this context, pre-pruning aims to stop the specialization of the left hand side of a rule. Post-pruning aims to simplify the left hand side of a rule after its generation has been completed. An example

of the methods for pruning of if-then rules is referred to as Jmid-pruning [8], which follows the strategy of pre-pruning.

Section 5 will report experimentally more detailed analysis of model complexity controlled through scaling up algorithms in terms of both rule generation and rule pruning.

4.2 Scaling Down Data

As mentioned in Section 3, the size of data may also affect the complexity of a rule based model. In other words, if a data set has a large number of attributes with various values and instances, the generated model is very likely to be more complex.

The dimensionality issue can be resolved by using feature selection techniques such as Correlation Based Feature Selection (CFS) [16]. In other words, the aim is to remove those irrelevant attributes and thus make a model simpler. In addition, the issue can also be resolved through feature extraction methods, such as Principal Component Analysis (PCA) [17]. In other words, the aim is to transform the data set to a lower dimensional space through combining existing attributes.

Besides, in some cases, it is also necessary to remove some attribute values as they may be irrelevant. For example, in a rule based model, an attribute-value pair may never be involved in any rules as a rule term. In this case, the value of this attribute can be judged irrelevant and thus removed. In some cases, it is also necessary to merge some values for an attribute in order to reduce the attribute complexity especially when the attribute is continuous with a large interval. There are some ways to deal with continuous attributes such as ChiMerge [18] and use of fuzzy linguistic terms [19].

As analyzed in Section 3.2, dimensionality reduction can effectively reduce the average number of rule terms per rule. This is because each single rule can have only up to n rule terms, where n is the data dimensionality. As also analyzed in Section 3.2, reduction of the complexity for each input attribute can effectively reduce the number of rules. For example, three attribute a, b, c have 2, 3 and 4 values respectively. In this case, the number of first order rules (with one rule term) is $2+3+4=9$; the number of second order rules (with two rule terms) is $2\times3+2\times4+3\times4=26$; the number of third order rules (with three rule terms) is $2\times3\times4=24$.

On the basis of the above description, feature selection, feature extraction and reduction of attribute complexity are all generally effective towards reduction of model complexity. More detailed experimental results are reported in Section 5 to analyze the extent to which the model complexity can be effectively controlled through scaling down data.

5 Experimental Studies

This section presents the validation of the proposed techniques mentioned in Section 4 for effective control of model complexity towards advances in model efficiency and interpretability. In particular, the validation includes five parts: rule generation, rule pruning, feature selection, feature extraction and attribute discretization. The first two parts are in line with scaling up algorithms and the rest of them are in line with scaling down data.

In terms of rule generation, ID3 is chosen as an example of the divide and conquer approach and IEBRG is chosen as an example of the separate and conquer approach. This is based on the fact that both methods involve use of information entropy as the heuristic for uncertainty measure. The rule based models generated by using the above two methods are compared in terms of model complexity. In particular, for the purpose of advancing model efficiency, the models generated by ID3 and IEBRG are compared in terms of the total number of rule terms generated. This is because the computational complexity of a rule based model in predicting the class of an unseen instance is typically measured by using the BigO notation and considering the worst case. As introduced in [3], if a rule based model is represented in the form of a decision tree, then the prediction is made by going through the tree from the root node to a leaf node in a divide and conquer search. The computational complexity is O (log (n)), where *n* is the tree size, i.e. the number of nodes in the tree. If a rule based model is represented in the form of a set of if-then rules, then the prediction is made by linearly going through the whole rule set until the firing rule is found. The computational complexity is O (n), where *n* is the total number of rule terms in the rule set. In this experimental study, the models generated by ID3 are all converted from the form of decision trees to the form of if-then rules in order to make consistent comparisons with models generated by IEBRG. This is because models represented in different forms cannot be compared consistently, and it is straightforward to convert from a decision tree to a set of if-then rules but much more difficult the other way around. On the other hand, for the purpose of advancing model interpretability, the models generated by ID3 and IEBRG are compared in terms of the number of rules and average number of rule terms per rule in a rule set. In this context, models generated by ID3 need to be converted from the form of decision trees to the form of if-then rules with the same reason as mentioned above. However, in general, models represented in the form of decision trees can be checked in terms of height and width, i.e. the length of the longest branch of a tree and the number of branches/leaf nodes respectively. This is because of the fact that the two popular search strategies are depth first search and breadth first search.

In terms of rule pruning, C4.5 is chosen as an example of the divide and conquer approach with the use of REP for tree pruning. The decision is based on the fact that C4.5 is a popular decision tree learning method and REP has been proven effective in reduction of overfitting of models generated by C4.5 towards improvement of

accuracy [15]. In addition, Prism is chosen as an example of the separate and conquer approach with the use of Jmid-pruning for pruning of if-then rules. The decision is based on the fact that Prism is a representative method for learning of if-then rules and Jmid-pruning has been proven effective in reduction of overfitting of models generated by Prism towards improvement of accuracy [8, 14]. In the case of decision tree pruning, the comparisons on model complexity are in terms of tree size, height and width, whereas the comparisons in the case of pruning of if-then rules are in terms of the total number of rule terms, number of rules and average number of rule terms per rule.

In terms of feature selection, feature extraction and attribute discretization, CFS, PCA and ChiMerge are used respectively to assist C4.5 for the purpose of data preprocessing. This is in order to reduce the level of difficulty for rule based modelling towards effective control of model complexity. In particular, models generated by C4.5 on the basis of the original data are compared in terms of tree size, height and width, with those ones generated by the same method on the basis of the processed version of the data by CFS for feature selection, PCA for feature extraction, and ChiMerge for attribute discretization.

All parts of the validation mentioned above are undertaken by using data sets retrieved from the UCI repository and the characteristics of these data sets can be seen in [13]. The results for the rule generation part are presented in Table 2 and 3.

Table 2. Total number of rule terms

Dataset	ID3	IEBRG
vote	333	**24**
zoo	12000	**5**
car	**54**	91
breast-cancer	167	**7**
kr-vs-kp	502	**9**
lung-cancer	52015	**3**
mushroom	6595	**9**
nursery	**85**	846
soybean	12257	**9**
splice	5815740	**10**
tic-tac-toe	**174**	605
trains	11048	**2**
contact-lenses	5	**3**
sponge	75240	**4**
audiology	14475	**7**

It can be seen from Table 2 that IEBRG outperforms ID3 in 12 out of 15 cases in terms of the total number of rule terms. The same phenomenon can also be seen from Table 3. In the three cases that IEBRG performs worse than ID3, the reason is that IEBRG cannot effectively learn consistent rules from the three data sets *car*, *nursery* and *tic-tac-toe*. As reported in [20], on the above three data sets, IEBRG generates a large number of inconsistent rules, each of which has already included all attributes on its left hand side but still covers instances that belong to different classes. In this case, the number of terms of an inconsistent rule is exactly the same

as the number of attributes of the data set, which is the maximum as analyzed in Section 3.2, and thus leads to a higher level of model complexity.

On the basis of the above description, methods (e.g. IEBRG) that follow the separate and conquer approach typically generate a smaller number of simpler rules in comparison with methods (e.g. ID3) that follow the divide and conquer approach, while the former ones can effectively learn consistent rules with high quality. Therefore, rule based models generated by the former type of methods are generally more efficient and interpretable.

Table 3. Number of rules and average number of terms per rule

Dataset	ID3		IEBRG	
	Count(rules)	Avg(terms)	Count(rules)	Avg(terms)
vote	31	10.74	**10**	**2.4**
zoo	1500	8.0	**5**	**1.0**
car	**16**	**3.38**	23	3.96
breast-cancer	43	3.88	**7**	**1.0**
kr-vs-kp	228	21.94	**9**	**1.0**
lung-cancer	1631	31.89	**3**	**1.0**
mushroom	521	12.66	**9**	**1.0**
nursery	**20**	**4.25**	121	6.99
soybean	576	21.28	**9**	**1.0**
splice	190680	30.5	**10**	**1.0**
tic-tac-toe	**31**	**5.61**	91	6.65
trains	565	19.55	**2**	**1.0**
contact-lenses	3	1.67	**3**	**1.0**
sponge	3344	22.5	**4**	**1.0**
audiology	314	46.1	**7**	**1.0**

Table 4. Tree size I- Tree Pruning

Dataset	unpruned C4.5	pruned C4.5
anneal	72	**60**
breast-cancer	179	**22**
breast-w	45	**27**
car	186	**112**
credit-a	135	**43**
credit-g	466	**64**
diabetes	43	**15**
ecoli	51	**11**
heart-c	77	**21**
heart-h	47	**8**
heart-statlog	61	**25**
hepatitis	31	**1**
ionosphere	35	**9**
vote	37	**9**
segment	101	**59**

The results for the rule pruning part are presented in Table 4 and 5 for decision tree pruning and in Table 6 and 7 for pruning of if-then rules.

In terms of tree pruning, it can be seen from Table 4 that the pruned decision tree has a smaller size than the unpruned decision tree in all cases. The same phenomenon can also be seen from Table 5. This is because of the fact that REP is a post-pruning method and the aim is to replace a subtree with a leaf node without affecting any other branches/subtrees after the whole tree has been generated. In addition, as also analysed in [2], for a decision tree, pruning one branch does not affect any other branches normally growing when using either pre-pruning or post-pruning. Therefore, in the context of decision tree learning, if there are any branches taken pruning actions, the tree is definitely simpler than the one without pruning taken and thus more efficient and interpretable**.**

Table 5. Tree complexity analysis I – Tree Pruning

Dataset	unpruned C4.5		pruned C4.5	
	Tree height	Count(leafs)	Tree height	Count(leafs)
anneal	13	53	**12**	**44**
breast-cancer	7	152	**2**	**18**
breast-w	9	23	**27**	**14**
car	**6**	134	**6**	**80**
credit-a	10	101	**9**	**30**
credit-g	11	359	**8**	**47**
diabetes	10	22	**6**	**8**
ecoli	9	26	**4**	**6**
heart-c	8	46	**5**	**14**
heart-h	7	29	**4**	**5**
heart-statlog	10	31	**7**	**13**
hepatitis	10	16	**1**	**1**
ionosphere	12	18	**5**	**5**
vote	9	19	**5**	**5**
segment	15	51	**11**	**30**

Table 6. Total number of rule terms by Prism

Dataset	Prism without pruning	Prism with Jmid-pruning
cmc	168	**112**
vote	157	**77**
kr-vs-kp	368	**116**
ecoli	45	**33**
anneal.ORIG	**25**	44
audiology	173	**106**
car	**2**	6
optdigits	3217	**1287**
glass	**74**	79
lymph	13	**10**
yeast	62	**30**
shuttle	116	**12**
analcatdataasbestos	8	**7**
irish	15	**14**
breast-cancer	12	**11**

In terms of pruning of if-then rules, it can be seen from Table 6 that the pruned rule based model is simpler than the unpruned one in 12 out of 15 cases. The similar phenomenon can also be seen from Table 7. For the three exceptional cases, the reason could be explained by the fact that for learning of if-then rules, pruning one rule could affect the generation of all subsequent rules. In other words, taking pruning actions can effectively make the current rule simpler, but may disadvantage learning of the subsequent rules leading to generation of more complex rules if the current pruning action is not appropriately taken. In this case, the model accuracy is also decreased as reported in [8, 14].

Table 7. Number of rules and average number of rule terms by Prism

Dataset	Prism without pruning		Prism with Jmid-pruning	
	Count(rules)	Avg(terms)	Count(rules)	Avg(terms)
cmc	36	4.67	**25**	**4.48**
vote	25	6.28	**15**	**5.13**
kr-vs-kp	63	5.84	**21**	**5.52**
ecoli	24	**1.88**	**17**	1.94
anneal.ORIG	16	**1.56**	**12**	3.67
audiology	48	3.60	**38**	**2.79**
car	**2**	**1.0**	3	2.0
optdigits	431	7.46	**197**	**6.53**
glass	26	**2.85**	**24**	3.29
lymph	**10**	1.3	**10**	**1.11**
yeast	37	1.68	**20**	**1.5**
shuttle	30	3.87	**12**	**1.0**
analcatdataasbestos	**5**	1.6	**5**	**1.4**
irish	**10**	1.5	11	**1.27**
breast-cancer	**11**	1.09	**11**	**1.0**

Table 8. Tree size II - Feature Selection

Dataset	C4.5		C4.5 with CFS	
	Attribute#	Tree size	Attribute#	Tree size
kr-vs-kp	37	82	8	**16**
ionosphere	35	35	15	**31**
sonar	61	35	20	**29**
mushroom	23	30	5	**21**
anneal	39	72	10	**70**
waveform	41	677	16	**621**
spambase	58	379	16	**229**
splice	62	3707	23	**555**
sponge	46	18	4	**6**
cylinder-bands	40	**432**	7	**432**
audiology	70	62	17	**59**
lung-cancer	57	12	9	**7**
spectf	45	**17**	13	19
credit-g	21	466	4	**30**
breast-cancer	10	179	6	**94**

The results for the rest of the parts are presented in Table 8 and 9 for feature selection, Table 10 and 11 for feature extraction and Table 12 and 13 for attribute discretization.

In terms of feature selection, it can be seen from Table 8 that the tree generated by using the pre-processed data is simpler than the one generated by using the original data in 13 out of 15 cases. The similar phenomenon can also be seen from Table 9. For the case on the cylinder-bands data set that the same tree is generated after the data dimensionality is reduced, the reason is typically that C4.5 does not select any irrelevant attributes for learning of a decision tree, while the data set is not pre-processed, and that the set of attributes removed by CFS does not contain any of the attributes that are supposed to be selected by C4.5 for generation of the tree. In addition, the other case on the *spectf* data set could normally be explained by the possible reason that there are a few relevant attributes removed posterior to the pre-processing of the data set, which disadvantages the learning of a tree by C4.5.

Table 9. Tree complexity analysis II - Feature Selection

Dataset	C4.5		C4.5 with CFS	
	Tree height	Count(leafs)	Tree height	Count(leafs)
kr-vs-kp	14	43	**7**	**9**
ionosphere	12	18	**10**	**16**
sonar	8	18	**7**	**15**
mushroom	6	25	**5**	**17**
anneal	13	53	**12**	**45**
waveform	20	339	**17**	**311**
spambase	31	190	**19**	**115**
splice	**9**	3597	11	**440**
sponge	4	14	**2**	**5**
cylinder-bands	**3**	**430**	**3**	**430**
audiology	14	**37**	**10**	38
lung-cancer	4	8	**3**	**5**
spectf	**7**	**9**	8	10
credit-g	11	359	**5**	**21**
breast-cancer	7	152	**6**	**80**

Table 10. Tree size III – Feature Extraction

Dataset	C4.5		C4.5 with PCA	
	Attribute#	Tree size	Attribute#	Tree size
vehicle	19	207	8	**165**
waveform	41	677	35	**369**
spambase	58	379	49	**345**
trains	33	11	9	**3**
hepatitis	20	31	17	**9**
lung-cancer	57	12	26	**7**
vowel	14	277	20	**241**
sonar	61	**35**	31	**35**
sponge	46	18	66	**7**
autos	26	88	37	**61**
car	7	186	16	**123**
cmc	10	665	16	**197**
heart-statlog	14	61	13	**21**
dermatology	35	44	72	**17**
tic-tac-toe	10	208	17	**43**

In terms of feature extraction, it can be seen from Table 10 that the tree generated by using the pre-processed data set is simpler than the one generated by using the original data set in 14 out of 15 cases. The similar phenomenon can also be seen from Table 11. For the case of the *sonar* data set that the same tree is generated after the data is transformed by PCA, the reason is typically that C4.5 can very effectively learn from the data set without the need to transform the data and thus the data transformation by PCA does not provide any help.

Table 11. Tree complexity analysis III - Feature Extraction

Dataset	C4.5		C4.5 with PCA	
	Tree height	Count(leafs)	Tree height	Count(leafs)
vehicle	17	104	**15**	**83**
waveform	**20**	339	**20**	**185**
spambase	31	190	**16**	**173**
trains	3	9	**2**	**2**
hepatitis	10	16	**4**	**5**
lung-cancer	**4**	8	**4**	**4**
vowel	**11**	178	23	**121**
sonar	**8**	**18**	**8**	**18**
sponge	4	14	**3**	**4**
autos	**8**	65	16	**31**
car	**6**	134	18	**62**
cmc	15	437	**14**	**99**
heart-statlog	10	31	**6**	**11**
dermatology	**7**	33	8	**9**
tic-tac-toe	**7**	139	14	**22**

Table 12. Tree size IV - Attribute Discretization

Dataset	C4.5 with original attributes	C4.5 II with discretised attributes
anneal	72	**64**
balance-scale	119	**13**
heart-c	77	**71**
heart-h	47	**45**
heart-statlog	61	**43**
labor	22	**13**
sick	72	**58**
tae	69	**5**
liver-disorders	53	**3**
cmc	665	**462**
colic	129	**107**
haberman	47	**15**
glass	59	**50**
weather	**8**	**8**
hypothyroid	**36**	76

In terms of attribute discretization, it can be seen from Table 12 that the tree generated by using the discretized set of attributes is simpler than the one generated by using the original data set. The similar phenomenon can also be seen from Table

13. For the case on the *hypothyroid* data set, the reason is typically that much information gets lost after inappropriate discretization of continuous attributes. In particular, if the discretization is not appropriate, it is very likely to result in the case that important patterns cannot be learned from important continuous attributes and thus more attributes need to be selected for learning of a tree. A similar argumentation is also made in [21].

Table 13. Tree complexity analysis IV- Attribute Discretization

Dataset	C4.5 with original attributes		C4.5 with discretised attributes	
	Tree height	Count(leafs)	Tree height	Count(leafs)
anneal	13	53	**10**	**50**
balance-scale	11	60	**5**	**7**
heart-c	**8**	46	9	**42**
heart-h	7	29	**6**	**26**
heart-statlog	10	31	**8**	**22**
labor	5	13	**4**	**8**
sick	**11**	41	**11**	**35**
tae	12	35	**3**	**3**
liver-disorders	9	27	**2**	**2**
cmc	15	417	**9**	**325**
colic	**7**	95	**7**	**82**
haberman	4	34	**3**	**13**
glass	11	**30**	**6**	35
weather	**3**	**5**	**3**	**5**
hypothyroid	10	**20**	**8**	57

6 Conclusion

This paper argued the significance of complexity control for rule based models for the purpose of knowledge discovery and predictive modelling. In particular, rule based models need to be more efficient and interpretable. This paper also identified two main impact factors for model complexity namely learning algorithms and data characteristics, and also analyzed in what way the two factors impact on the model complexity. The main contributions of this paper include theoretical analysis of the proposed techniques for control of model complexity and empirical validation of these techniques to show the extent to which these techniques are effective towards generation of more efficient and interpretable models in practice. The results have been discussed critically and comparatively and indicated that the proposed techniques can effectively manage to reduce the model complexity. On the basis of the results obtained, the further directions identified for this research area are to investigate in depth how to employ existing methods to achieve scaling up algorithms and scaling down data, respectively, in more effective ways.

References

[1] H. Liu, A. Gegov and F. Stahl, "Categorization and Construction of Rule Based Systems," in *15th Inter-national Conference on Engineering Applications of Neural Networks*, Sofia, Bulgaria, 2014.

[2] J. Furnkranz, "Separate-and-Conquer rule learning," *Artificial Intelligence Review,* vol. 13, pp. 3-54, 1999.

[3] H. Liu, A. Gegov and M. Cocea, "Network Based Rule Representation for Knowledge Discovery and Predictive Modelling," in *IEEE International Conference on Fuzzy Systems*, Istanbul, 2015.

[4] C. Schaffer, "Overfitting Avoidance as Bias," *Machine Learning,* vol. 10, p. 153–178, 1993.

[5] D. H. Wolpert, "On Overfitting Avoidance as Bias," SFI TR, 1993.

[6] H. Liu, M. Cocea and A. Gegov, "Interpretability of Computational Models for Sentiment Analysis," in *Sentiment Analysis and Ontology Engineering: An Environment of Computational Intelligence*, vol. 639, W. Pedrycz and S. M. Chen, Eds., Switzerland, Springer, 2016, pp. 199-220.

[7] H. Liu, A. Gegov and F. Stahl, "Unified Framework for Construction of Rule Based Classification Systems," in *Inforamtion Granularity, Big Data and Computational Intelligence*, vol. 8, W. Pedrycz and S. M. Chen, Eds., Springer, 2015, pp. 209-230.

[8] H. Liu, A. Gegov and F. Stahl, "J-measure Based Hybrid Pruning for Complexity Reduction in Classification Rules," *WSEAS Transactions on Systems,* vol. 12, no. 9, pp. 433-446, 2013.

[9] R. Quinlan, C4.5: Programs for Machine Learning, Morgan Kaufman, 1993.

[10] J. Cendrowska, "PRISM: an algorithm for inducing modular rules," *International Journal of Man-Machine Studies,* vol. 27, p. 349–370, 1987.

[11] X. Deng, "A covering-based algorithm for classification: PRISM," SK, 2012.

[12] Q. Ross, "Induction of Decision Trees," *Machine Learning,* vol. 1, pp. 81-106, 1986.

[13] M. Lichman, "UCI Machine Learning Repository," University of California, Irvine, School of Information and Computer Sciences, 2013. [Online]. Available: http://archive.ics.uci.edu/ml. [Accessed 25 June 2015].

[14] H. Liu, A. Gegov and M. Cocea, Rule Based Systems for Big Data: A Machine Learning Approach, 1 ed., vol. 13, Switzerland: Springer, 2016.

[15] T. Elomaa and M. Kaariainen , "An Analysis of Reduced Error Pruning," *Journal of Artificial Intelligence Research,* vol. 15, no. 1, pp. 163-187, 2001.

[16] M. A. Hall, "Correlation-based Feature Selection for," Hamilton, NewZealand, 1999.

[17] I. T. Jolliffe, Principal Component Analysis, New York: Springer, 2002.

[18] R. Kerber, "ChiMerge: discretization of numeric attributes," in *Proceedings of the 10th National Conference on Artificial Intelligence*, California, 1992.

[19] T. J. Ross, Fuzzy Logic with Engineering Applications, West Sussex: John Wiley & Sons Ltd, 2004.

[20] H. Liu and A. Gegov, "Induction of Modular Classification Rules by Information Entropy Based Rule Generation," in *Innovative Issues in Intelligent Systems*, vol. 623, V. Sgurev, R. Yager, J. Kacprzyk and V. Jotsov, Eds., Switzerland, Springer, 2016, pp. 217-230.

[21] D. Brain, "Learning from Large Data: Bias, Variance, and Learning Curves," 2003.