

Categorization and Construction of Rule Based Systems

Han Liu¹, Alexander Gegov¹ and Frederic Stahl²

¹ School of Computing, University of Portsmouth, Buckingham Building, Lion Terrace,
PO1 3HE, Portsmouth, United Kingdom
{Han.Liu, Alexander.Gegov}@port.ac.uk

² School of Systems Engineering, University of Reading, Po Box 225 Whiteknights,
Reading, RG6 6AY, United Kingdom
F.T.Stahl@reading.ac.uk

Abstract. Expert systems have been increasingly popular for commercial importance. A rule based system is a special type of an expert system, which consists of a set of 'if-then' rules and can be applied as a decision support system in many areas such as healthcare, transportation and security. Rule based systems can be constructed based on both expert knowledge and data. This paper aims to introduce the theory of rule based systems especially on categorization and construction of such systems from a conceptual point of view. This paper also introduces rule based systems for classification tasks in detail.

Keywords: Data Mining, Machine Learning, Rule Based Systems, Rule Based Classification, if-then Rules

1 Introduction

The development of rule based systems began in the 1960's but became popular in the 1970's and 1980's [1]. A rule based system typically consists of a set of if-then rules, which can serve many purposes such as decision support or predictive decision making in real applications. One of the main concerns in this area is the construction of such systems which could be based on both expert knowledge and data. Thus the construction techniques can be divided into two categories: knowledge based construction and data based construction. This paper introduces the theoretical aspects of categorization and construction of rule based systems as well as the use for classification tasks. The purpose is to explore the research direction in context as well as combine the authors' previous work together to make an evolution from specialization to generalization for the theoretical concepts.

The rest of this paper is organized as follows: Section 2 introduces the categorization of rule based systems according to some special characteristics; Section 3 introduces two main categories of construction of rule based systems: knowledge based construction and data based construction. A special type of rule based systems used for classification tasks is introduced in detail in Section 4. The potential of this approach is also specified in a healthcare case study in Section 5 to

demonstrate the value and impact of the approach. The summary of completed work and further directions of research in this area are highlighted further in Section 6.

2 Categorization of Rule Based Systems

Rule based systems can be categorized in the following aspects: *number of inputs and outputs, type of input and output values, type of structure, type of logic, type of rule bases, number of machine learners and type of computing environment.*

For rule based systems, both inputs and outputs could be single or multiple. From this point of view, rule based systems can be divided into four types [2] with respect to number of inputs and outputs: *single-input-single-output, multiple-input-single-output, single-input-multiple-output, and multiple-input-multiple-output.* All the four types above can fit the characteristics of association rules. This is because association rules reflect the relationship between attributes. An association rule may have a single or multiple rule terms in both antecedent (left hand side) and consequent (right hand side) of the rule. Thus the categorization based on number of inputs and outputs is very relevant to fulfill the distinction of association rules.

However, association rules include two special types: classification rules and regression rules, depending on type of output values. Both classification rules and regression rules may have a single or multiple rule terms in antecedent but can only have a single term in the consequent. The difference between classification rules and regression rules is that the output values of classification rules must be discrete while those of regression rules must be continuous. Thus both classification rules and regression rules fit the characteristics of 'single-input-single-output' or 'multiple-input-single-output' and are seen as special type of association rules. As the basis of above description, rule based systems can also be categorized into three types with respects to both number of inputs and outputs and type of input and output values: *rule based classification systems, rule based regression systems and rule based association systems.*

In machine learning, classification rules can be generated in two approaches: divide and conquer [3] and separate and conquer [4]. The former method is generating rules directly in the form of a decision tree, whereas the latter method produces a list of 'if-then' rules. An alternative structure called Rule Based Networks represents rules in the form of networks. With respect to structure, rule based systems can thus be divided into three types: *treed rule based systems, listed rule based systems and networked rule based systems.*

Construction of rule based systems is based on special type of logic such as Boolean logic, fuzzy logic and probabilistic logic. From this point of view, rule based systems can also be divided into the following types: *deterministic rule based systems, probabilistic rule based systems and fuzzy rule based systems.*

As rule based systems can also be in the context of rule bases including single rule bases, chained rule bases and modular rule bases. From this point of view, rule based systems can also be divided into the three types: *standard rule based systems, hierarchical rule based systems and networked rule based systems.*

In machine learning context, a single algorithm could be applied to a single data set for training a single learner. It can also be applied to multiple samples of a data set by ensemble learning techniques for construction of an ensemble learner which consists of a group of single learners. In addition, there could also be a combination of multiple algorithms involved in machine learning tasks. From this point of view, rule based systems can be divided into two types according to the number of machine learners constructed: *single rule based systems and ensemble rule based systems*.

In practice, an ensemble learning task could be done in parallel, distributed way or a mobile device according to the specific computing environments. Therefore, rule based systems can also be divided into the following three types: *parallel rule based systems, distributed rule based systems and mobile rule based systems*.

3 Construction of Rule Based Systems

As mentioned in Section 1, the construction of rule based systems can be based on both expert knowledge and data. This section introduces and discusses two special types of construction: *knowledge based construction and data based construction*.

3.1 Knowledge Based Approach

Knowledge based construction follows a traditional engineering approach, which is in general domain dependent. It is necessary to have knowledge or requirements acquired from experts at first and then to identify the relationships between attributes (features). Modelling, which is the most important step, is further to be executed in order to build a set of rules. Once the modelling is complete, then simulation is started to check the model towards fulfillment of systematic complexity such as model accuracy and efficiency. Finally, statistical analysis is undertaken in order to validate whether the model is reliable and efficient in application.

3.2 Data Based Approach

Data based construction follows a machine learning approach, which is in general domain independent. Machine learning techniques can be subdivided into two types: supervised learning and unsupervised learning. Supervised learning means learning with a teacher. This is because all instances from a data set are labelled. The aim of this type of learning is to predict attribute values for unknown instances by using the known data instances [5]. The predicted value of an attribute may be either discrete or continuous. Therefore, supervised learning could be involved in both classification and regression tasks for categorical prediction and numerical prediction respectively. On the other hand, unsupervised learning means learning without a teacher. This is because all instances from a data set are unlabeled. The aim of this type of learning is to find previously unknown patterns from data sets. It includes association, which

aims to find relationships among attributes with regards to their values [5], and clustering, which aims to find a group of objects that are similar from data sets [5].

As mentioned in Section 1, rule based systems can be used for construction of classification, regression and association systems. In general, all the three types of rule based systems can be constructed with the following steps: Data collection->Data pre-processing->Learning from data->Testing. However, there are different requirements in different learning tasks. In other words, in order to build a high quality model by using machine learning techniques, it is important to find algorithms which are suitable to the chosen data sets with respects to the characteristics of data. From this point of view, data preprocessing may be not necessary if the chosen algorithms are good fits. In addition, different type of dimensionality reduction techniques (such as feature selection), a type of data preprocessing, may be required for different tasks. If it is a classification or regression task, supervised feature selection techniques may be required in general. Otherwise unsupervised feature selection techniques may be suitable. The step for learning from data mentioned above may also need to be broken down in some special cases. For example, it may be required to simplify rules in classification tasks or to reduce the number of rules in association tasks. A specific construction for rule based classification systems is further introduced in more detail in Section 4.

3.3 Discussion

In this paper, the authors aim to motivate the use of data based approach instead of knowledge based approach for construction of complex rule based systems. The main reason is that expert knowledge may be incomplete or inaccurate; some of experts' points of view may be biased; engineers may misunderstand requirements or have technical designs with defects. In other words, with regards to solving problems with high complexity, both domain experts and engineers are difficult to have all possible cases considered or to have perfect technical designs. Once a failure arises with such a system, experts or engineers may have to find the problem and fix it by reanalyzing or redesigning. However, the real world has been filled with Big Data. Some previously unknown information or knowledge may be discovered from data. Data may potentially be used as supporting evidence to reflect some useful and important pattern by using modeling techniques. More importantly, the model could be revised automatically as the update of database in real time if data based modeling technique is used. Therefore, data based approach may be more suitable than knowledge based approach for construction of complex systems. The rest of the paper will focus on discussion in the machine learning context.

4 Rule Based Classification Systems

In general, a unified framework for the construction of predictive rule based systems, comprises three basic procedures, the generation of rules, the simplification of rules and the rule representation. This section describes the essence of the three operations

and introduces some methods and techniques which are involved in the operations. The methods and techniques are also discussed comparatively in order to highlight some important aspects in choosing methods or techniques for the fulfillment of each of the three operations.

4.1 Rule Generation

As mentioned in Section 2, the methods for generation of classification rules can be categorized into the 'divide and conquer' and the 'separate and conquer' approaches. Examples for 'divide and conquer' comprise ID3 [3], C4.5 and C5.0. Examples for 'separate and conquer' comprise Prism [7] and PrismTCS [8].

Divide and conquer is a recursive approach as the generation of rules is to select an attribute to split on and then to recursively repeat the process for each branch covering a subset of the training set as illustrated in Fig.1. However, this approach has a principal drawback, the replicated sub-tree problem pointed out in [7] and illustrated in Fig.2. It can be seen from Fig.2 that the four sub-trees which all have node C as root are identical. This is an unnecessary redundancy in the decision tree as illustrated in Fig.2.

<p>IF all cases in the training set belong to the same class THEN return the value of the class ELSE</p> <ul style="list-style-type: none"> (a) Select the attribute A to split on* (b) Sort the instances in the training set into non-empty subsets, one for each value of attribute A (c) Return a tree with one branch for each subset, each branch having a descendant sub-tree or a class value produced by applying the algorithm recursively for each subset in turn. <p>*When selecting attributes at step (a) the same attribute must not be selected more than once in any branch.</p>

Fig.1 TDIDT Tree Generation algorithm [5]

As the problem arises with the rule generation approach, the separate and conquer approach is motivated to generate if-then rules directly and iteratively from training instances. Prism is a method that follows the 'separate and conquer' approach and is illustrated in its original form in Fig.3. Bramer developed a modified version of Prism called PrismTCS. The motivation is to increase computational efficiency because original Prism is computationally expensive. The expensive computation is resulted from frequent deletion of instances during rule generation and restoring the training data to its initial size for rule generation for next class [12]. PrismTCS always chooses the minority class as target class. Thus PrismTCS induces rules in the order of their importance without the restoring the data to its original size (in between the induction of different rules) [8, 9, 10]. PrismTCS has shown to produce classification rules much faster, but also of a similar level of predictive accuracy compared with original Prism [8, 9, 13]. However, the authors have recently pointed out some limitations of Prism algorithm in [6, 14] regarding Prism's way of dealing with clashes, underfitting of the concept in the training data and its computational efficiency.

With respects to clashes, it indicates that Prism may generate a number of rules, each of which covers a clash set. A clash set contains instances that belong to

different classifications but cannot be separated further. According to Bramer's Inducer software implementation for clash handling, Prism prefers to discard a rule instead of assigning it to the majority class. It may result in underfitting of the training set if a large number of rules get discarded. For original Prism, this case may result in a large number of instances remaining unclassified as there is no default rule available and the rules that cover the instances get discarded. For PrismTCS, this case may make a default rule give wrong classifications to the instances covered by discarded rules. This is because the default rule is supposed to cover only the instances that belong to the majority class, but unfortunately some rules that cover the other instances got discarded.

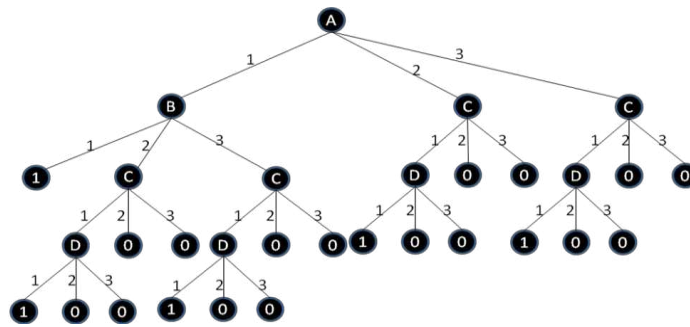


Fig. 2 Cendrowska's replicated subtree example [6, 20]

With respects to computational efficiency, as mentioned above, Prism prefers to discard a rule if a clash occurs. That indicates that the algorithm takes time to generate a rule which is eventually discarded in some cases. It is equivalent to doing nothing and results in unnecessary computational costs.

Execute the following steps for each classification (class= i) in turn and on the original training data S:

1. $S' = S$.
2. Remove all instances from S' that are covered from the rules induced so far. If S' is empty then stop inducing further rules
3. Calculate the conditional probability from S' for class=i for each attribute-value pair.
4. Select the attribute-value pair that covers class= i with the highest probability and remove all instances from S' that comprise the selected attribute-value pair
5. Repeat 3 and 4 until a subset is reached that only covers instances of class= i in S' . The induced rule is then the conjunction of all the attribute-value pairs selected.

Repeat 1-5 until all instances of class i have been removed

*For each rule, no one attribute can be selected twice during rule generation

Fig.3 Basic Prism algorithm [5]

The authors have recently developed another rule generation method called "Information Entropy Based Rule Generation" (IEBRG) which also follows separate and conquer approach and is illustrated in Fig.4. However, it uses "from cause to effect" approach whereas Prism uses "from effect to cause" approach. The main focus of IEBRG is on minimizing the uncertainty that exists in the subset no matter what the target class is. A popular measure of uncertainty is information entropy introduced by Shannon in [15]. One of the advantages of IEBRG compared with Prism can be seen from an example with reference to the lens 24 dataset reconstructed by Bramer in [5]. The dataset indicates that $p(class=3|tears=1) = 1$. The first rule generated could be "if tears=1 then class=3". This implies that "tears=1" is only relevant for predicting class 3. IEBRG can capture this information by the conditional entropy $E(tears=1)$

=0. However, this is actually unknown prior to the rule induction by Prism algorithm. The PrismTCS would assign *class 1* as target class to the first rule being generated (as *class 1* is the minority class). Original Prism may also select *class 1* as the index of the class is smaller. However, according to [8] the first rule generated by original Prism is “if astig=2 and tears=2 and age=1 then class=1”. It indicates that the computational cost is slightly higher than expected and so the rule has a higher complexity. In some cases, the Prism algorithm may be even generating incomplete rules, covering a clash set, especially if the target class is not a good fit to the attribute-value pairs in the training data. The rule may be discarded resulting in underfitting and unnecessary computational cost.

1. Calculate the conditional entropy of each attribute-value pair in the current subset
 2. Select the attribute-value pair with the smallest entropy to be split on, i.e. remove all other instances that do not comprise the attribute-value pair.
 3. Repeat step 1 and 2 until the current subset contains only instances of one class (the entropy of the resulting subset is zero).
 4. Remove all instances covered by this rule.
- Repeat 1-4 until there are no instances remaining in the training set.
- * For each rule, no one attribute can be selected more than once during generation.

Fig.4 IEBRG algorithm

In comparison with the Prism algorithm family, IEBRG may need significantly less computational effort. In contrast to Prism, the IEBRG algorithm deals with clashes by assigning a majority class to the rule. This may potentially reduce the underfitting of the rule set and thus reduce the number of unclassified instances. However, there is potential that the number of misclassified instances increases. Yet, IEBRG is potentially better in avoiding clashes compared with Prism.

4.2 Rule Simplification

Rule simplification is necessary in some cases. The reason is the principal problem of rule based classifiers to overfit on the training data [17]. When a large data set is used for training, this may lead to the induction of a very large number of complex rules. This will lower both the predictive accuracy and the computational efficiency. This has motivated the development of pruning methods for rule simplification with respect to the reduction of overfitting. Pruning methods can be subdivided into two categories- pre-pruning [5] and post-pruning [5]. The former prunes rules during rule generation and the latter generates a whole rule set and then discards a number of rules and rule terms, by means of using statistical (or other) tests [17]. There is a family of pruning algorithms for Prism algorithms based on the J-measure [18], an information theoretic means to compute the theoretical information content of a rule. This is based on the hypothesis [19] that, if a rule has high information content (value of J-measure, or also called J-value), it is also prone to have a high classification accuracy. Two existing J-measure based pruning algorithms are J-pruning [17] and Jmax-pruning [9, 10]. They have been successfully applied on different versions of Prism algorithms for reducing overfitting. When a rule is being generated, the J-value may go up or go down after specialising the rule by appending an additional term. Both pruning algorithms expect to find the global maximum of J-value for the rule. Each rule is assigned a complexity degree which is the number of terms. The increase

of complexity degree may make the J-value of this rule go up or down. The pruning algorithms are aimed at finding the complexity degree corresponding to the global maximum of J-value as illustrated in Fig. 5 using a fictitious example. Both pruning methods above employ different strategies to search for the global maximum of the J-value. J-pruning monitors the change of the J-value when appending rule terms and stops once the J-value goes down. In contrast, Jmax-pruning induces the rule fully until complexity degree X_3 (regarding Fig.4), but monitors and records the so far highest J-value when appending rule terms. In the example in Fig.5, J-pruning would stop inducing rule terms when reaching complexity degree X_1 but Jmax-pruning would stop when reaching complexity degree X_3 and then reduce the complexity degree to X_2 by removing rule terms between X_3 and X_2 .

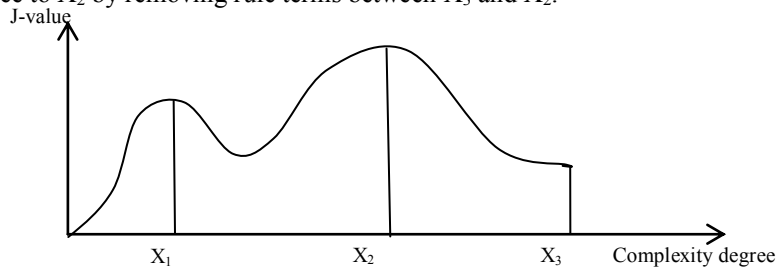


Fig.5 Relationship between complexity degree and J-value

The authors have recently developed an alternative J-measure based pruning algorithm called Jmid-pruning [20] in order to overcome the limitations mentioned above. This algorithm not only monitors and records the highest J-value observed but also measures the Jmax value that may be achieved by adding further rule terms.

In comparison with Jmax-pruning, Jmid-pruning also always finds the global maximum but it is in theory computationally more efficient in some cases. An example [11] is considered that a rule could be generated using the lense24 dataset:

If tears=2 and astig=1 and age=3 and specRx =1 then class= 3;

As the rule is being specialized by appending the four terms subsequently, the corresponding values of J and Jmax change in the pattern as follows:

If tears=2 then class=3; (J=0.210, Jmax=0.531)

If tears=2 and astig=1 then class=3; (J=0.161, Jmax=0.295)

If tears=2 and astig=1 and age=3 then class=3; (J=0.004, Jmax=0.059)

If tears=2 and astig=1 and age=3 and specRx =1 then class= 3; (J=0.028, Jmax=0.028)

In the example above all three pruning algorithms would generate the same rule: if tears=2 then class=3. The reason is that the highest J-value is computed right after the first rule term was added (tears=2). However, with regard to computational efficiency, J-pruning is the fastest and stops right after the second term (astig=1) is generated. Jmid-pruning is faster than Jmax-pruning. This is because Jmax-pruning stops when the rule is complete and cuts it back to 'if tears=2 then class=3' but Jmid-pruning stops the generation after the third term is generated as the Jmax-value is below the so far highest J-value.

4.3 Rule Representation

Rule representation aims to represent a rule set in a suitable structure to achieve more efficient prediction. As mentioned in Section 2, the existing rule representations include decision tree and linear list. The former is a representation that automatically represents classification rules induced using the 'divide and conquer' method. The latter automatically represents rules generated by the 'separate and conquer' method. However, the decision tree representation has been criticised by Cendrowska and identified as a major reason for overfitting [7]. It is also pointed out in [16] that in the worst case it needs to go through the entire tree for extracting a classification. It undoubtedly increases the computational costs and thus is a major drawback, hence the motivation for using 'if-then' that can be represented in a linear list structure. However, prediction on test instances by the list representation is done in linear time while the number of rule terms in the rule set is the input size n . It indicates it may have to go through the whole rule set in the worst case in order to find the first rule firing. This may result in huge computational costs in prediction stage when a rule set is very complex. Therefore, the authors have recently developed a new representation called Rule Based Classification Networks [6] which performs logarithmic time.

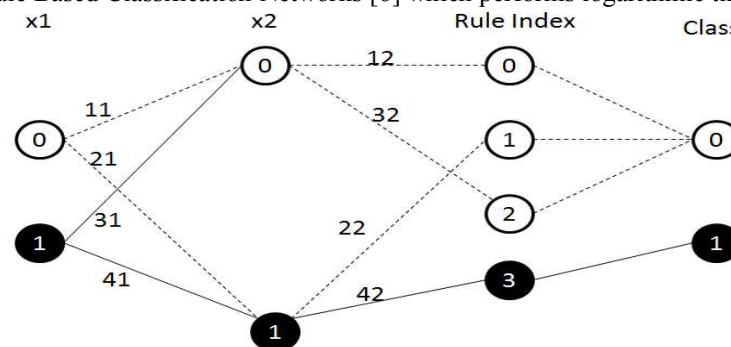


Fig.6 Rule Based Classification Networks [6]

The networked representation is illustrated in Fig.6 to represent a rule set based on Boolean logic. The rule set has two input attributes (x_1 and x_2) and the class value is 1 if and only if both variables get input value of 1. In this representation, the terms: $x_1=1$ and $x_2=1$, are the two inputs for testing. Thus, in both 'x1' and 'x2' layers, node labeled 1 becomes black and node labeled 0 becomes white. This is because each node in layer x1 represents a value of attribute x_1 and equivalent in layer x_2 . The two digits with which the connections between layer x1 and x2 are labeled represent the index of the rule and rule term respectively, i.e. the two digits '11' state that this is the first term of the first rule. It can also be easily seen that this particular term is ' $x_1=0$ '. However, as the value of x_1 is 1, the connection is not satisfied and thus becomes dot. The connections '31' and '41' are both solid because condition ' $x_1=1$ ' is met. The same principle applies to the connections between layers 'x2' and 'Rule Index'. The connections '31', '41' and '42' become solid as the inputs are $x_1=1$ and $x_2=1$; and this thus results in that node 3 becomes black in the 'Rule Index' layer and the output is 1 in the layer 'Class'.

5 Applications in Healthcare

As mentioned in Section 3, rule based classification systems constructed by machine learning approach are domain independent and thus can be applied in many areas. For example, as mentioned in [21], Inductive learning algorithms are domain independent and can be involved in any classification or pattern recognition tasks. Some successful applications listed in [21] include lymphography, prognosis of breast cancer recurrence, location of primary tumour and thyroid problem diagnosis in medicine [22, 23, 24].

The authors have recently developed a healthcare process modeling approach which could be implemented in the following procedures using classification rules:

- diagnosis of illness
- patient classification
- treatment recommendation

In detail, patients need to have diagnosis with regards to the illness. This could be achieved by using classification rules to identify the possible illness of a patient. Once the diagnosis is complete, the patient will be categorized into a special group for this illness based on the risk level by using classification rules checking patient's features such as blood pressure and heart rate. Finally, a list of treatments can be recommended by using classification rules checking patient's features and finding all fired rules.

6 Conclusion

This paper introduces the categorization of rule based systems for both academic and commercial purpose. This brings new insights to researchers and practitioners and positions a new type of rule based systems for applications. It also motivates the use of data based approach in the context of machine learning for construction of complex systems instead of knowledge based approach. The importance of the data based approach is also highlighted. A special type of rule based system used for classification tasks is introduced in detail explaining the construction framework and reflecting some important aspects of choosing methods or techniques for rule generation, simplification and representation operations. This is in order to explore the significance of data based approach in depth. In addition, a healthcare case study is also specified to demonstrate the value and impact of the approach. The construction framework could be extended to include multiple rule based classification systems as a whole by means of a system of systems by adopting an ensemble learning approach. Such an extended framework for constructing ensemble rule based systems will be validated. The studies will also be extended towards fulfilment for construction of rule based systems for regression and association tasks.

References

1. Partridge, D., Hussain, K.M.: Knowledge Based Information Systems. Mc-Graw Hill (1994)
2. Gegov, A.: Fuzzy Networks for Complex Systems: A Modular Rule Base Approach. Springer, Berlin (2010)
3. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufman (1993)
4. Michalski, R.S.: On the Quasi-Minimal solution of the general covering problem. In: Proceedings of the Fifth International Symposium on Information Processing. Bled, Yugoslavia, pp. 125–128 (1969)
5. Bramer, M.A.: Principles of Data Mining. Springer, London (2007)
6. Liu, H., Gegov, A., Stahl, F.: Unified Framework for Construction of Rule Based Classification Systems. In: Pedrycz, W., Chen, S.M. (eds.) Springer, Berlin (In press)
7. Cendrowska, J.: PRISM: An Algorithm for Inducing Modular Rules. International Journal of Man-Machine Studies, vol. 27, pp. 349–370 (1987)
8. Bramer, M.A.: Automatic Induction of Classification Rules from Examples using N-Prism. Research and Development in Intelligent Systems, vol. XVI, pp. 99–121. Springer-Verlag, Cambridge (2000)
9. Stahl, F., Bramer, M.A.: Jmax-pruning: A Facility for the Information Theoretic Pruning of Modular Classification Rules. Knowledge-Based Systems, vol.29, pp.12-19 (2012)
10. Stahl, F., Bramer, M.A.: Induction of Modular Classification Rules: using Jmax-pruning. In: In Thirtieth SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence, pp.14-16. Springer, Cambridge (2011)
11. Bramer, M.A.: Inducer: a Public Domain Workbench for Data Mining. International Journal of Systems Science, vol. 36, no. 14, pp.909-919 (2005)
12. Stahl, F., Bramer, M.A.: Computationally Efficient Induction of Classification Rules with the PMCRI and J-PMCRI Frameworks. Knowledge-Based Systems, vol.35, pp.49-63 (2012)
13. Bramer, M.A.: An Information-theoretic Approach to the Pre-pruning of Classification Rules. In: Musen, B.N., Studer, R. (eds.) Intelligent Information Processing, Kluwer, pp. 201-212 (2002)
14. Liu, H., Gegov, A.: Induction of Modular Classification Rules by Information Entropy Based Rule Generation. In: Sgurev, V., Yager, R., Kacprzyk, J. (eds.) Innovative Issues in Intelligent Systems. Springer (In press)
15. Shannon, C.: A Mathematical Theory of Communication. Bell System Technical Journal, vol.27, no.3, pp. 379-423. Fonn (1948)
16. Deng, X.: A Covering-based Algorithm for Classification: PRISM. CS831: Knowledge Discover in Databases (2012)
17. Bramer, M.A.: Using J-Pruning to Reduce Overfitting of Classification Rules in Noisy Domains. In: Proceedings of 13th International Conference on Database and Expert Systems Applications—DEXA. Aix-en-Provence, France (2002)
18. Smyth, P., Goodman, R.M.: Rule Induction Using Information Theory. In: Pietatsky-Shapiro, G., Frawley, W.J. (eds.) Knowledge Discovery in Databases, pp. 159-176. AAAI Press (1991)
19. Bramer, M.A.: Using J-Pruning to Reduce Overfitting in Classification Trees. In: Research and Development in Intelligent Systems XVIII, pp. 25-38. Springer-Verlag (2002)
20. Liu, H., Gegov, A., Stahl, F.: J-measure Based Hybrid Pruning for Complexity Reduction in Classification Rules. WSEAS Transaction on Systems, vol.12, no. 9, pp. 433-446 (2013)
21. Aksoy, M.S.: A Review of Rules Families of Algorithms. Mathematical and Computational Applications, vol. 13, no. 1, pp. 51-60 (2008)
22. Quinlan, J.R.: Induction, Knowledge and Expert Systems. In: Artificial Intelligence Developments and Applications, Gero, J.S. and Stanton, R. (eds.) Amsterdam, North Holland, pp. 253-271 (1988)

23. Michalski, R.S., et al.: The Multi-purpose Incremental Learning System AQ15 and Its Testing Application to Three Medical Domains. In: Proc. National Conf. on AI, Philadelphia, PA., August, pp. 1041-1044 (1996)
24. Quinlan, J.R.: Inductive Knowledge Acquisition: a Case Study. In: Applications of Expert Systems, Quinlan, J.R. (ed.) Turing Institute Press, pp. 157-173 (1987)