International Journal of Scientific & Engineering Research, Volume 3, Issue 8, August-2012 1
ISSN 2229-5518

KDSODTEX: A Novel Technique to Extract Second Order Decision Table Using KDRuleEx

Kamal Kumar Sethi, Durgesh Kumar Mishra, Bharat Mishra

Abstract— Decision tree and decision tables are well known representational model for classification technique. Decision table is widely used due to its comprehensibility and expainability. KDRuleEx algorithm extracts accurate and comprehensible single order decision table. Many times decision tables may be lengthy due to duplication of most of the feature values. This lengthy decision table can be reduced by allowing non atomic values in decision table. A decision table having non atomic feature values is called second order decision table. An equivalent and more comprehensible second order decision table can be extracted by applying certain transformation rules on the single order decision table. In this paper, a novel algorithm KDSODTEX, for extracting second order decision table with the use of KDRuleEx has been proposed. The experiment and result shows that extracted table is more comprehensible and equivalent**.**

Index Terms— Rule Extraction, decision table, second order decision table, comprehensibility, transparency, accuracy, classifier, ANN.

---

1 Introduction

RULE extraction is the task of transforming an opaque model into transparent and hopefully comprehensible model. Comprehensibility is one of the most targeted research area in rule extraction. Comprehensibility is required in many areas and decision table represents the classification of the dataset in comprehensible and explainable form then the other classifier like decision tree, if then rule etc. Rules represented by decision table may have common values for many feature causing redundancy. By removing this redundancy size of decision table can be reduced. These rules with many common values with same decision may be combined to represent a set of rules. In this paper we propose a novel algorithm KDSODTEX which takes first order decision table as input and extracts equivalent second order decision table. Result shows that the new algorithm is capable of deriving equivalent, more comprehensible and consistent second order decision tables. Second order decision table associate the same classification(s) to any condition as given to first order table. The resultant second order decision table increases the comprehensibility by having less number of rules and feature [8][11] which increase the local and global comprehensibility.

2 Literature Survey

Classifier model can be represented in the form of if-then rules [9], association rules, decision trees [10] and decision table [1]. The usefulness of a classifier entirely depends on comprehensibility, accuracy and explain-ability of the model. Decision tree and decision table are widely used representation model for classifying data and rule extraction techniques. The improved and more comprehensible representation model of decision tree is decision table which can be easily interpreted by users[13]. Decision tables are widely used in knowledge based decision support systems [12]. Research shows that decision table wins user confidence over decision tree due to its resemblance with most familiar structures like spread sheets, relational data etc[8]. In literature, extraction of single order decision tables is done: by viewing reverse Naïve bayes structures [6], by combining feature subset selection and computation of probabilities [7], by using the wrapper algorithm that selects features for a hypothesis with the highest future prediction accuracy, known as IDTM algorithm[7], and by using the ANN, known as KDRuleEx [1].

The idea of rough set theory is to extract rules by feature and value reduction in the building the ability of the classification [3]. Based on rough set theory proposed by Pawlak [14] an information system is a useful concept for classification of data. Information system is based on the assumption that to every object of the universal set some information is associated. The information system contains data about objects of interest characterized in terms of some attributes. If we distinguish condition and decision attributes in the information system, then such a system is called a decision table. The concept of information system and decision table was originally proposed by Pawlak. Formally Information System can be represent by $S = (U, A_t)$. If there exist C, D $\subseteq$ At such that C $\cap$ D = Φ and C $\cup$ D = A_t , then S may be treated as decision table, denote by S = (U, C, D) [14]where C is known as condition attributes and D as decision attribute. The decision table contains a set of decision rules. The expression if Φ then ψ is called a decision rule, where Φ and ψ belong to C and D, respectively [16, 17, 18, 19]. Then, certainty factor and coverage factor are defined for every decision rule [4]. The two factors show correctness and consistency of a decision rule [5, 17, 20], based on which the classification of decision rules is done.

The structure of conventional decision table has two main components scheme and body. The scheme represents set of condition attributes and class attribute and the body is essen-

---

- *Kamal Kumar Sethi is currently pursuing PhD program incompuer engineering from MGCGV, Chitrakoot, Satna (MP), India and Professor & Head, Department of Information Technology, Acropolis Institute of Technology and Research, Indore(MP), India. PH-+91-9425720000. E-mail: kamalksethi@gmail.com*
- *Dr. Durgesh Kumar Mishra is Professore and Dean (R&D) in Sri Aurobindo Institute of Technology, Indore (MP), India and senior member of IEEE and chairman of CSI Indore chapter. E-mail: durgeshmishra@ieee.org*
- *Dr. Bharat Mishra is currentlyreader in MGCGV University, Chitrakoot, Satna (MP), India. E-mail: bm_cgv@rediffmail.com*

tially a table of labeled data items where the attributes specified by the scheme form the rows and the decisions (classifications) form the columns [15]. The empty component in condition is referred to as "don't care" values. With every condition attribute $a \in C$, set of values Va is associated, called the domain of that attribute. Each column of decision table induces a decision rule.

3 Proposed Technique: KDSODTEX

Second order decision table can be generated by applying certain transformational rules on first order decision table. Two similar rules having only one attribute value changed, can be joined, resulting non atomic entries in decision table. Non atomic entry is a set of values and will be treated as disjunction or choices. In second order decision tables, attribute values appearing in the form of set inside the table populated. If non atomic entry for a attribute $a \in C$ is equal to complete domain value Va, then that attribute can be replaced by $\varnothing$. Second order decision table is capable of representing complex rules more easily. Second order table can be viewed as a type of one-level nested relation [13]. Logical view of structure of conventional table for the light bulb example [21] has been shown in table 1. An equivalent second order and simplified structure of second order has been shown in table 2 and table 3. The ϕ in table 3 represents don't care condition. Don't care condition represents that all possible feature values.

Table 1: Conventional Decision Table

Decision Attribute	Off	Off	Off	On	On	On	On	Off
Rule Length	3	3	3	3	3	3	3	3
Att1	T	T	T	T	F	F	F	F
Att2	T	T	F	F	T	T	F	F
Att3	T	F	T	F	T	F	T	F

Table 2: Second Order Decision Table

Decision Features	On	Off	Off	Off	On	On
Rule Length	3	3	2	2	2	2
Att1	T	F	T	T	F	F
Att2	F	F	T	{T,F}	T	{T,F}
Att3	F	F	{T,F}	T	{T,F}	T

Table 3: Simplified Second Order Decision Table

Decision Features	On	Off	Off	Off	On	On
Rule Length	3	3	2	2	2	2
Att1	T	F	T	T	F	F
Att2	F	F	T	ϕ	T	ϕ
Att3	F	F	ϕ	T	ϕ	T

Decision table represented in table 2 and table 3 are an equivalent second order decision table having less number of rule and rule length in comparison to decision table specified by Table 1 which result increased comprehensibility. In the process of extraction of decision table two types of comprehensibility, global and local can be visualized where global comprehensibility deals with entire size of table and local deals with size of individual rule length [2]. Second order decision table would be useful for users as well as machines for faster decision making.

1. **Join:** Any two rules having all the attribute values same except one must be joined and can be replaced by joined rule. This reduces the size of decision table and increases the comprehensibility. Joining of two rules generate non atomic entry for the changed attribute in the joined rule. The non atomic entry is set of atomic values. If the changed attribute has all possible value of that attribute equivalent to domain then that attribute value can be replaced by $\varnothing$ in the joined rule, equivalent to saying that this attribute valued do not have any effect. Let R1 = {x1, x2, x3, x4} and R2 = {x1, y1, x3, x4}. R1 Join R2 extracts new rule R12 as {x1, {x2,y1}, x3, x4} and is added in decision table. If {x2, y1} is set of all possible attribute values for the second attribute then the joined rule R12 can be changed from {x1, {x2,y1}, x3, x4} to {x1, $\varnothing$, x3, x4} resulting reduce rule length.
2. **Elimination:** Joining of R1 and R2 affects the decision table by adding new rule R12 and eliminating R1 and R2.
3. **Rule Avoidance:** Rules having minimal role in classifying the data set can be merged with the highly matched rule.
4. **Reduction of Rows:** In our decision table, rows represent the contribution of feature values amongst various decision rules specified by columns of the decision table. The size of decision table may be reduced by deleting the unnecessary features which has no value in body [11].

KDRuleEx algorithm extracts first order decision table from the given data set. The output of KDRuleEx is used as input for the proposed algorithm (KDSODTEX) which extracts second order decision table. Detailed KDSODTEX algorithm has been presented in Figure 2 and Figure 3. KDSODTEX starts by initializing maxRow and maxCol by size of the rows and columns of first order decision table. Increase maxRow by one to add one row at bottom to denote the processing status of the associated rule. Algorithm then compares unprocessed rules for checking the availability of two rules having one feature value changed against the same class. Such rules can be joined according to the transformational rule no 1, 2 and 3. After joining the rules if changed feature value includes all the feature values from domain Va, then this feature value can be set $\varnothing$ (NULL) in new joined rule. Whenever a join condition sets true new rule is added in the decision table. If ith rule is tested against jth rule then denote the status of ith and jth column "processed" in the last row of second order decision table.

Algorithm: KDSODTEX (DT)
/* Extract second order decision table from conventional decision table */
Input: First Order Decision Table,
Output: Second Order Decision Table
Method:

1. maxRow = Number of Row in DTable+1;
2. maxCol= Number of column in DTable;
3. DTable(maxRow,1)={'Processed'};
4. **for** i=2: maxCol-1
5. **for** j=2: maxCol
6. Compare unprocessed i^{th} rule to j^{th} rule with same decision and where i<>j
7. **if** only one value is changed in above selected two rule column then
8. maxCol= maxCol+1
9. Add a new rule into decision table by union of these two rules.
10. Check the value of changed attribute in joined rule for domain completion,
11. **if** it's domain complete then replace it by ∅ (NULL)
12. Mark i^{th} and j^{th} rule processed
13. end if
14. end if
15. end for
16. end for

Figure 2: Algorithm for second order decision table

4 Experiment And Result

The proposed algorithm has been tested for the zoo data set downloaded from UCI data repository. This data set is popularly used for classification. This data set contains 101 instances characterized by 16 categorical attributes and classified by 7 different classes. The details of distribution of instance in different class have been shown in Table 4. Our experiment shows that proposed algorithm is capable to reduce number of rules and rule length.

We have used Info Gain technique for identifying the splitting feature in this study. We have observed that using this technique new kind of relationships amongst features inside the rule has been found, which were not reported earlier. Table 5 represents the first order decision table of Zoo dataset extracted using KDRuleEx. Total no rules represented by decision table are 13. Table 6 represents the equivalent second order decision table extracted from KDSODTEX by summarizing 13 rules into 11 rules. Table 7 represents the confusion matrix showing the accuracy of the extracted model.

5 Conclusion

In this paper we have presented an algorithm that extracts accurate yet comprehensible second order decision table from decision table generated using KDRuleEx. KDRuleEx algorithm uses ANN for accurately inducing first order decision table from the dataset. Experiment and results conclude that explain-ability of second order decision table is better that the first order decision table. The results show that our extracted model is better because obtained rules are optimized on several criteria and results show a high level of accuracy. Extracted second order decision table is an equivalent classification model of first order decision table having less number of rules and rule length.

Table-4: Class wise Distribution of Instances

Value of Target Class	# of Instance in dataset	# of Instance correctly classify in each Class	% of Accurate classify Instances in each Class
Mammal	41	41	100 %
Bird	20	20	100 %
Reptile	5	5	100 %
Fish	13	13	100 %
Amphibian	4	4	100 %
Fly	8	8	100 %
Shellfish	10	9	90 %

Experiment result shows that the accuracy and fidelity is not compromised, number of don't care (∅) conditions in table gets increased, resulting more comprehensible representation. Extracted model and rules can easily be used with other for machine learning systems and average complexity to make prediction using extracted model gets reduced resulting better system performance. Column level (rule) simplification in decision table has been proposed in current algorithm by joining the rule. Simplification against rows (feature) can be done for those features which are not sufficiently contributing in decisions. The proposed algorithm has been suggested for the categorical feature values. Algorithm can be used for discrete or ordinal by making some changes in algorithm.

References

[1] Kamal Kumar Sethi, Durgesh Kumar Mishra, Bharat Mishra, "KDRuleEx: A Novel Approach for Enhancing User Comprehensibility Using Rule Extraction," isms, pp.55-60, 2012 Third International Conference on Intelligent Systems Modelling and Simulation, 2012.

[2] Augasta, M. Gethsiyal; Kathirvalavakumar, T.; , "Rule extraction from neural networks — A comparative study," Pattern Recognition, Informatics and Medical Engineering (PRIME), 2012 International Conference on , vol., no., pp.404-408, 21-23 March 2012

[3] XianWen Luo; , "Study of algorithm for rule extraction based on rough set theory," *Fuzzy Systems and Knowledge Discovery (FSKD), 2011 Eighth International Conference on* , vol.1, no., pp.343-346, 26-28 July 2011.

[4] Yang, H., Xu, Z., Zhang, J., & Cai, J., "A Constructing Method of Decision Tree and Classification Rule Extraction for Incomplete Information System," *2010 International Conference on Computational Aspects of Social Networks*, pp. 49-52., 2010.

[5] Intan, R., M. Mukaidono, and Y.Y. Yao, Generalization of rough sets with alpha-coverings of the universe induced by conditional probability relations, *Proceedings of International Workshop on Rough Sets and Granular Computing*, pp.311-315, 2001.

[6] Langley, P., Induction of condensed determinations, in Proceeding of

Second International Conference of Knowledge Discovery and Data Mining, pp. 327-330, AAAI Press, 1996.

[7] Kohavi, R., The power of decision tables, in proceedings of the European Conference on Machine Learning (ECML), 1995.

[8] Rattikorn Hewett, John H. Leuchner: The Power of Second-Order Decision Tables. SDM 2002.

[9] Apte, C. and S. Hong, Predicting equity returns from securities data. In Fayad, U., Piatetsky-Shapiro, G., Smyth, P., Uthurusamy, R. (eds.): Advances in Knowledge Discovery and Data Mining. AAAI Press/ The MIT Press, Menlo Park, CA, 307-328, 1996.

[10] Quinlan, J., C4.5: Programs for Machine Learning, Morgan Kaufmann, San Mateo, California, 1993.

[11] Min Dong, Hui Yu Jiang, Minimum Attribute number in Decision Table Based on Maximum Entropy Principle, in Proceeding of the International conference in Environmental Science and Computer Engineering, pp. 446-448.

[12] Rattikorn Hewett, John Leuchner, Restructuring decision tables for elucidation of knowledge, Data & Knowledge Engineering, v.46 n.3, p.271-290, September 2003.

[13] Mues C., Huysmans J., Baesens B., Vanthienen J., An empirical investigation into the interpretability of data mining models based on decision trees, tables and rules, 22nd European Conference on Operational Research, Prague, July 8-11, 2007.

[14] Zdzislaw Pawlak, "Rough set theory and its applications", Journal of Telecommunications and information Technology, March, 2002.

[15] Victoria J. Hodge, Simon O'Keefe, Jim Austin, "A Binary Neural Decision Table Classifier", Advanced Computer Architecture Group 2004.

[16] *J.H. Nasiri, and M. Mashinchi*, "Rough Set and Data Analysis in Decision Tables", Proceeding in Journal of Uncertain Systems Vol.3, No.3, pp.232-240, 2009

[17] Pawlak, Z, Rough sets and intelligent data analysis, *Information Sciences*, vol.147, pp.1-12, 2002

[18] Pawlak, Z., and R. Slowinski, Decision analysis using rough sets, *International Transactions in Operational Research*, vol.1, pp.107-114, 1994.

[19] Pawlak, Z., and R. Slowinski, Rough set approach to multi attribute decision analysis, *European Journal of Operational Research*, vol.72, pp.443-449, 1994.

[20] Intan, R., and M. Mukaidono, Generalization of rough membership function based on alpha-covering of the universe, *Proceeding of AFSS*, pp.129-135, 2002.

[21] Humphrey, Watts S. A Discipline for Software Engineering Reading, MA: Addison-Wesley.

TABLE 5: FIRST ORDER DECISION TABLE FOR ZOO DATASET

Mammal	Mammal	Reptile	Mammal	Fly	Shellfish	Shellfish	Reptile	Mammal	Bird	Shellfish	Shellfish	Amphibian	Fish
Rule Length	2	3	2	2	1	1	3	3	2	2	3	3	3
hair	T		T				F		F			F	
eggs								F					T
aquatic				F			F			T		T	
toothed		T									F		
fins		F						T			F		T
legs	4	0	2	6	8	5	4	0	2	6	0	4	0

TABLE 6: SECOND ORDER DECISION TABLE FOR ZOO DATASET

Mammal	Reptile	Fly	Reptile	Mammal	Bird	Shellfish	Shellfish	Amphibian	Fish	Mammal	Shellfish
Rule Length	3	2	3	3	2	2	3	3	3	2	1
hair			F		F			F		T	
eggs				F					T		
aquatic		F	F			T		T			
toothed	T						F				
fins	F			T			F		T		
legs	0	6	4	0	2	6	0	4	0	{2,4}	{5,8}

ISSN 2229-5518

TABLE 7: CONFUSION MATRIX

	Mammal	Fish	Bird	Shellfish	Fly	Amphibian	Reptile	
Mammal	41	0	0	0	0	0	0	41
Fish	0	13	0	0	0	0	0	13
Bird	0	0	20	0	0	0	0	20
Shellfish	0	0	0	9	0	1	0	10
Fly	0	0	0	0	8	0	0	8
Amphibian	0	0	0	0	0	4	0	4
Reptile	0	0	0	0	0	0	5	5
	41	13	20	9	8	5	5	101