J-MEASURE BASED PRUNING FOR ADVANCING CLASSIFICATION PERFORMANCE OF INFORMATION ENTROPY BASED RULE GENERATION

HAN LIU[1], MIHAELA COCEA[2], WEILI DING[3]

[1]School of Computer Science and Informatics, Cardiff University
Queen's Buildings, 5 The Parade, Cardiff, CF24 3AA, United Kingdom
[2]School of computing, University of Portsmouth
Buckingham Building, Lion Terrace, Portsmouth, PO1 3HE, United Kingdom
[3]Laboratory of Pattern Recognition and Intelligent Systems
Key Laboratory of Industrial Computer Control Engineering of Heibei Provience, Department of Automation
Institute of Electrical Engineering, Yanshan University, Qinghuangdao, 066004, China
E-MAIL: liuh48@cardiff.ac.uk, mihaela.cocea@port.ac.uk, weiye51@ysu.edu.cn

Abstract:

Learning of classification rules is a popular approach of machine learning, which can be achieved through two strategies, namely divide-and-conquer and separate-and-conquer. The former is aimed at generating rules in the form of a decision tree, whereas the latter generates if-then rules directly from training data. From this point of view, the above two strategies are referred to as decision tree learning and rule learning, respectively. Both learning strategies can lead to production of complex rule based classifiers that overfit training data, which has motivated researchers to develop pruning algorithms towards reduction of overfitting. In this paper, we propose a J-measure based pruning algorithm, which is referred to as Jmean-pruning. The proposed pruning algorithm is used to advance the performance of the information entropy based rule generation method that follows the separate and conquer strategy. An experimental study is reported to show how Jmean-pruning can effectively help the above rule learning method avoid overfitting. The results show that the use of Jmean-pruning achieves to advance the performance of the rule learning method and the improved performance is very comparable or even considerably better than the one of C4.5.

Keywords:

Data mining; Machine learning; Decision tree learning; Rule learning; Rule pruning; J-measure

1. Introduction

The generation of classification rules is a main branch of machine learning, which can be achieved through two inductive strategies, namely, divide-and-conquer and separate-and-conquer. The former is also known as Top-Down Induction of Decision Trees (TDIDT), since it essentially aims to generate classification rules in the form of a decision tree, whereas the latter is also known as covering approach, since it trains a rule based classifier by learning a rule that covers a subset of training instances and then learning the next rule from the remaining instances. From the above point of view, we refer to the former strategy as decision tree learning and to the latter strategy as rule learning, in the rest of this paper.

As described in [1], decision tree learning is likely to produce a complex rule based classifier that overfits training data. Also, as argued in [2], the nature of decision tree learning requires that all rules (extracted from different branches of a decision tree) must have at least one common attribute, which is likely to result in the replicated sub-tree problem. In order to address the above issues of decision tree learning, people have been motivated to develop pruning algorithms for generating simpler decision trees and avoiding the case of overfitting. On the other hand, development of rule learning methods has become a main way to eliminate the replicated sub-tree problem that arises with decision tree learning methods. However, rule learning methods are also likely to encounter the issue of

overfitting [1]. This again motivated people to develop pruning methods for simplifying rules.

In this paper, we focus on investigating the adoption of rule pruning, towards advancing the performance of rule learning methods, which essentially follow the separate and conquer strategy. In particular, we propose a J-measure based pruning algorithm, which is referred to as Jmean-pruning, and investigate both theoretically and empirically the impact of the pruning algorithm on the Information Entropy Based Rule Generation (IEBRG) method [3]. We also compare IEBRG with a popular decision tree learning method (C4.5), in terms of both their own performance and the impacted performance through use of pruning algorithms.

The rest of this paper is organized as follows: Section 2 presents related work on decision tree learning, rule learning and pruning. In Section 3, we illustrate the procedure and key features of the proposed J-measure based pruning algorithm and also justify its theoretical significance on advancing the performance of rule learning. In Section 4, we report an experimental study and show the results on how the Jmean-pruning algorithm impacts the performance of IEBRG. In Section 5, the main contributions of this paper will be summarized, and relevant further directions will be suggested towards advancing this research area in the future.

2. Related work

Decision tree learning has been used as a popular approach of machine learning in real applications, since the trained models are represented in a white box manner so that they can be interpretable to people.

In terms of algorithmic development, decision tree learning has been highly competitive and technically sound, since the ID3 algorithm was developed by [4] with very good performance especially on the chess end games data set [5]. However, the ID3 algorithm can not directly handle continuous attributes, i.e. discretization of continuous attributes is needed prior to training of classifiers. In order to overcome the above limitation of ID3, the C4.5 algorithm was developed as an extension for effectively handling continuous attributes and replacing missing values [6, 7]. Another popular algorithm of decision tree learning is CART [8], which stands for classification and regression tree. CART is essentially aimed at generating a binary tree through binarization of multi-valued attributes. More details can be found in [9].

In addition, decision tree learning methods have been extended in several ways: a) ensemble learning of decision trees, e.g. random forests (RF) [10]; b) incorporating cost functions into heuristics for attribute selection [11], [12] and [13]; c) fuzzification of continuous attributes for generation of fuzzy decision trees [14], [15], and [16].

Due to the case that decision tree learning usually results in very complex models being trained, the use of pruning methods has become a way to simplify decision trees and avoid overfitting of training data. The most popular pruning method used for C4.5 is reduced error pruning (REP) and the one popularly used for CART is referred to as cost complexity pruning (CCP) [17]. A comparison study of pruning methods was reported in [18]. However, although pruning methods can help simplify decision trees in the training stage, it is still difficult to avoid the situation that the trained classifiers are too cumbersome, complex and inscrutable to provide insight for people to use as domain knowledge [6, 1]. According to the argumentation made in [2], the generation of a complex decision tree classifier is partially due to the replicated sub-tree problem as illustrated in Fig. 1. Moreover, complex classifiers are more likely to overfit training data than simpler classifiers [1].

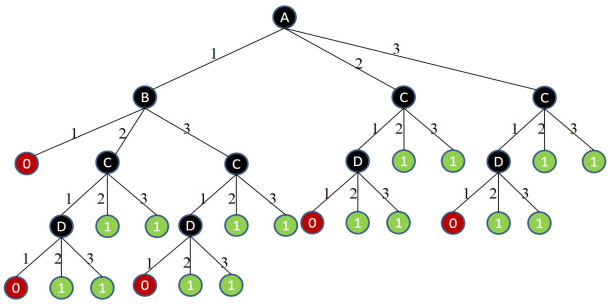

FIGURE 1. Cendrowska's replicated subtree example [19]

In order to address the above issues that arise with decision tree learning, people were motivated to develop algorithms that lead to direct generation of if-then rules, through the separate and conquer strategy, as mentioned in Section 1. In particular, the Prism algorithm was developed in [2] for eliminating the replicated subtree problem. The experimental results reported in [2] indicate that the Prism algorithm led to a rule based classifier much simpler than the one trained by ID3 on the chess game data set, while the two algorithms show the same performance in terms of classification accuracy. Some other rule learning algorithms were developed later on alongside the operation of rule pruning, which include Repeated Incremental Pruning to Produce Error Reduction (RIPPER) [20] and Information Theoretical Rule Induction (ITRULE) [21]. The pruning strategy involved in RIPPER is based on REP, whereas ITRULE involves J-measure based pruning [21]. J-measure

based pruning has also been used for simplifying rules learned by Prism [22]. Our proposed pruning algorithm is also based on J-measure, which will be presented in Section 3 for advancing the performance of IEBRG.

3. J-measure based pruning approach

In this section, we present the theoretical preliminaries on J-measure and the essence of the IEBRG algorithm. Also, we illustrate how the proposed Jmean-pruning works to simplify rules learned by using the IEBRG algorithm.

3.1 J-measure

J-measure is an information theoretic measure that quantifies the average information content of a single rule, which is essentially the product of two terms as defined in Eq. 1.

$$J(Y, X = x) = P(x) \cdot j(Y, X = x) \tag{1}$$

The first term is the probability that the antecedent $X = x$ (left hand side) of a rule occurs, and it is considered as a measure of rule simplicity [21], since a simpler rule generally has a higher probability. The second term is read as j-measure, which is a measure of goodness-of-fit of a rule and is also known as cross entropy [21]. The j-measure is defined in Eq. 2.

$$\begin{aligned} j(Y, X = x) = P(y|x) \cdot \log_2 \frac{P(y|x)}{P(y)} + \\ (1 - P(y|x)) \cdot \log_2 \frac{1 - P(y|x)}{1 - P(y)} \end{aligned} \tag{2}$$

where $P(y|x)$ represents the posterior probability of the class y given that the rule antecedent $X = x$ occurs, and $P(y)$ represents the prior probability of the class y.

According to [21], J-measure has an upper bound, which is referred to as Jmax and is defined in Eq. 3.

$$\begin{aligned} Jmax = P(x) \cdot \max(P(y|x) \cdot \log_2 \frac{P(y|x)}{P(y)}, \\ (1 - P(y|x)) \cdot \log_2 \frac{1 - P(y|x)}{1 - P(y)}) \end{aligned} \tag{3}$$

As illustrated in [22], J-measure and Jmax were used jointly for design of the Jmid-pruning algorithm towards simplifying rules learned by Prism. In particular, when a rule is specialized by adding a term to its left hand side, the value of J-measure may increase or decrease. However, the value of Jmax would monotonically decrease when a rule is being specialized. In this context, the values of J-measure and Jmax obtained at each iteration would be observed, and the value of J-measure obtained at the current iteration is compared with the maximum value of J-measure observed during the previous iterations, which means to update the maximum value if the value of J-measure at the current iteration is higher. Jmid-pruning would stop the learning of a rule at an iteration when the value of Jmax obtained at the iteration has been lower than the maximum value of J-measure observed previously. The procedure of Jmid-pruning is illustrated by using the following example:

To generate a rule: if a=2 and b=1 and c=3 and d=1 then class=3; there would be four iterations involved for adding the rule terms subsequently. In this process, the J-measure and Jmax values at the four iterations are changed as follows:

- Iteration 1: if a=2 then class=3; (J-measure=0.230, Jmax=0.538)
- Iteration 2: if a=2 and b=1 then class=3; (J-measure=0.165, Jmax=0.297)
- Iteration 3: if a=2 and b=1 and c=3 then class=3; (J-measure=0.006, Jmax=0.079)
- Iteration 4: if a=2 and b=1 and c=3 and d=1 then class=3; (J-measure=0.029, Jmax=0.029)

For the above example, Jmid-purning would stop the learning of this rule at iteration 3, since the value (0.079) of Jmax has been lower than the maximum value (0.230) of J-measure observed previously.

3.2 Information Entropy Based Rule Generation

The procedure of IEBRG is illustrated in Algorithm 1, which is essentially to select an attribute-value pair that obtains the minimum value of conditional entropy (see Eq. 4) and can best discriminate between different classes. A step-by-step illustration of the algorithm can be found in [23].

$$E = -\sum_{i=0}^{c} p(class_i|A_x = v_j) \log_2 p(class_i|A_x = v_j) \tag{4}$$

where A_x represents an attribute; x is the index of the attribute, v_j is a value of the attribute A_x and j is the index of the attribute value. Also, $p(class_i|A_x = v_j)$ is read as the conditional probability of classifying an instance to $class_i$ given that $A_x = v_j$.

Algorithm 1: IEBRG Algorithm

Input : a training set T, a subset $T' \subseteq T$, an attribute set AS, an instance $t \in T$, dimensionality d, an attribute A_x, an attribute value v_{xn}, conditional entropy E, class C_i

Output: a rule set RS, a result set of instances T" covered by a rule $R \in RS$

```
Initialize: T' = T, T" = T, E = 1;
while T' ≠ φ do
    while E ≠ 0 do
        x = 0; j = 0; E = 1; while x < d do
            k = 0;
            for each value v_xn of A_x do
                Calculate E(A_x = v_xn)
                if E(A_x = v_xn) < E) then
                    E = E(A_x = v_xn); j = x; k = n;
                end
            end
            x + +;
        end
        assign A_j = v_jk to R as a rule term, when
          E(A_j = v_jk) is minimal; AS= AS - {A_j};
        d= d − 1; ∀t : T" = T" − {t}, if t ∈ T" and t
          comprise A_j = v_jk;
    end
    RS= RS ∪ {R}; T'= T' - T";
end
```

3.3. Key features of Jmean-pruning

As illustrated in Section 3.1, J-measure and Jmax are used jointly for design of the Jmid-pruning algorithm, which is essentially aimed at pruning rules that are learned for a target class, e.g. the Prism algorithm needs to select a class as the target class, and then a set of rules are learned for the target class. In this context, it is known in advance which class is assigned as the consequent of a rule being learned, so the way illustrated in Eqs. 2 and 3 respectively for computing the values of j-measure and Jmax would work out.

However, as shown in Algorithm 1, the essence of IEBRG makes it impossible to know in advance which class will be eventually assigned as the consequent of a rule being learned, i.e. the class assigned as the rule consequent would not be known until the completion of learning this rule. From this point of view, the ways of computing the values of j-measure and Jmax need to be modified according to [21], such that the Jmean-pruning algorithm can be designed in a way that works effectively for IEBRG. In particular, the modified ways of computing the values of J-measure and Jmax are shown in Eqs. 5 and 6, respectively.

$$j(Y, X = x) = \sum_{i=0}^{n} (P(y_i|x) \cdot \log_2 \frac{P(y_i|x)}{P(y_i)}) \quad (5)$$

where i is the index of y and n is the number of classes.

$$Jmax = P(x) \cdot \max_{i=0}^{n} (P(y_i|x) \cdot \log_2 \frac{P(y_i|x)}{P(y_i)}) \quad (6)$$

In fact, Eqs 2 and 3 essentially mean to calculate the values of j-measure and Jmax in the case of binary classification, whereas Eqs 5 and 6 represent the case of multi-class classification. For example, there are three classes A, B and C involved in a classification task. The former way of calculation is achieved by considering the target class (say A) as the positive class and the combination of the other two classes (say B + C) as the negative class (say $\neg A$). In this way, J-measure is used to measure the average information content on discriminating the positive class from the negative class. However, in the latter way of calculation, J-measure is used to measure the average information content on discriminating between different classes, without defining a class as the positive class.

Since the essence of Prism is to learn each rule in a way that discriminates the target class (positive class) from the other classes (negative class), towards obtaining a full probability for the positive class, the ways of computing J-measure and Jmax values need to be in a binary manner. However, the essence of IEBRG is to learn each rule in a way that discriminates between different classes, towards reducing the entropy value (uncertainty) to 0, so it is more reasonable to compute the J-measure and Jmax values in a multi-class manner, but the pruning strategy remains the same as Jmid-pruning, i.e. the learning of a rule is stopped at iteration t when the Jmax values obtained at the iteration has been higher than the maximum value of J-measure observed during the previous $t - 1$ iterations.

4. Experimental setup, results and discussion

In this section, we report an experimental study conducted by using 15 UCI data sets [24]. The characteristics of the data sets are shown in Table 1.

In this study, we compare IEBRG with C4.5, in terms of their own performance and the affected performance through pruning. In particular, the REP algorithm is used for pruning of

TABLE 1. Data sets

Dataset	Attribute Types	#Attributes	#Instances	#Classes
anneal	mixed	38	798	6
breast-cancer	discrete	9	286	2
breast-w	continuous	10	699	2
credit-g	mixed	20	1000	2
cylinder-bands	mixed	40	540	2
hepatitis	mixed	20	155	2
ionosphere	continuous	34	351	2
iris	continuous	4	150	3
labor	mixed	17	57	2
lymph	mixed	19	148	4
mushroom	discrete	22	8124	2
tae	mixed	6	151	3
vote	discrete	16	435	2
wine	continuous	13	178	3
zoo	mixed	18	101	7

decision trees learned by C4.5, and the Jmean-pruning is used for pruning of rules learned by IEBRG. The experiments are conducted by partitioning a data set into a training set and a test set in the ratio of 70:30. On each data set, the experiment is repeated 100 times (in terms of data partitioning) and the average accuracy is taken for comparative validation. The results are shown in Table 2.

TABLE 2. Pruning Results

Dataset	C4.5		IEBRG	
	unpruned	pruned	unpruned	pruned
anneal	98%	98%	99%	99%
breast-cancer	67%	71%	66%	69%
breast-w	94%	94%	95%	95%
credit-g	68%	72%	64%	68%
cylinder-bands	58%	58%	71%	71%
hepatitis	76%	80%	81%	82%
ionosphere	89%	88%	89%	89%
iris	94%	94%	94%	94%
labor	80%	77%	83%	84%
lymph	76%	76%	77%	79%
mushroom	100%	100%	100%	100%
tae	53%	48%	60%	59%
vote	95%	95%	93%	94%
wine	91%	88%	92%	94%
zoo	92%	90%	89%	91%

The results show that Jmean-pruning achieves to improve the performance of IBERG in 8 out of 15 cases, which indicates the effectiveness of reducing overfitting. In the other 7 cases, the use of Jmean-pruning does not result in changes of the IEBRG performance, with an exception on the tae data set that the performance drops marginally. The above phenomenon would indicate that the use of Jmean-pruning does not result in underfitting, while classifiers trained by IEBRG do not overfit, since 5 out of 6 such cases show that the performance of IEBRG is considerably high (89% or above) without use of pruning. The performance of IEBRG is also very comparable to or even higher than the one of C4.5, especially when Jmean-pruning is used to avoid overfitting. In addition, the C4.5 performance drops in 3 cases when using REP, whereas Jmean-pruning shows its positive impact on the IEBRG performance in 2 out of the 3 cases, which again shows the effectiveness of Jmean-pruning.

5. Conclusions

In this paper, we proposed the Jmean-pruning algorithm for simplifying rules learned by IEBRG towards avoiding the case of overfitting. In particular, we analyzed the essence of J-measure and IEBRG, and designed Jmean-pruning by using J-measure in a modified way that works well for IEBRG. We investigated empirically the impact of Jmean-pruning on the performance of IEBRG, and the results show that Jmean-pruning achieves to improve the IEBRG performance, which shows to be very comparable to or even better than the performance of C4.5, even when the REP algorithm is used for pruning.

In future, we will investigate theoretically and empirically when pruning is necessary or not, based on the characteristics of data. In other words, use of pruning methods could reduce overfitting, but may also result in underfitting, especially when overfitting does not really occur. Therefore, it is important to identify the likelihood that overfitting may occur on specific data sets, such that both underfitting and overfitting would be avoided effectively. We will also investigate combination of different ways of rule quality measure for pruning purpose, towards generation of an optimal set of rules.

Acknowledgements

The authors acknowledge support for research reported in this paper through Research Development Fund at the University of Portsmouth and support from the China Scholarship Council and the Natural Science Foundation of Hebei Province, China (No. F2016203211).

References

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

[2] J. Cendrowska, "Prism: An algorithm for inducing modular rules," *International Journal of Man-Machine Studies*, vol. 27, pp. 349–370, May 1987.

[3] H. Liu and A. Gegov, *Induction of Modular Classification Rules by Information Entropy Based Rule Generation*. Switzerland: Springer, 2016, pp. 217–230.

[4] R. J. Quinlan, "Induction of decision trees," *Machine Learning*, vol. 1, no. 1, pp. 81–106, March 1986.

[5] J. R. Quinlan, "Learning efficient classification procedures and their application to chess end games," in *Machine Learning: An Artificial Intelligence Approach*, R. S. Michalski, J. G. Carbonell, and T. M. Mitchell, Eds., 1983, pp. 463–482.

[6] R. J. Quinlan, *C4.5: Programs for Machine Learning*. San Francisco: Morgan Kaufmann Publishers, 1993.

[7] J. R. Quinlan, "Improved use of continuous attributes in c4.5," *Journal of Artificial Intelligence Research*, vol. 4, pp. 77–90, March 1996.

[8] L. Breiman, J. Friedman, C. J. Stone, and R. Olshen, *Classification and Regression Trees*. Monterey, CA: Chapman and Hall/CRC, 1984.

[9] H. Liu and M. Cocea, "Induction of classification rules by gini-index based rule generation," *Information Sciences*, vol. 436-437, pp. 227–246, 2018.

[10] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[11] H. Zhao and X. Li, "A cost sensitive decision tree algorithm based on weighted class distribution with batch deleting attribute mechanism," *Information Sciences*, vol. 378, pp. 303–316, 2017.

[12] F. Min and W. Zhu, "A competition strategy to cost-sensitive decision trees," in *Rough Sets and Knowledge Technology: 7th International Conference, RSKT 2012, Chengdu, China, August 17-20, 2012. Proceedings*, T. Li, H. S. Nguyen, G. Wang, J. Grzymala-Busse, R. Janicki, A. E. Hassanien, and H. Yu, Eds., vol. 7414, 2012, pp. 359–368.

[13] X. Li, H. Zhao, and W. Zhu, "A cost sensitive decision tree algorithm with two adaptive mechanisms," *Knowledge-Based Systems*, vol. 88, pp. 24–33, November 2015.

[14] Y. Lertworaprachaya, Y. Yang, and R. John, "Interval-valued fuzzy decision trees with optimal neighbourhood perimeter," *Applied Soft Computing*, vol. 24, pp. 851–866, November 2014.

[15] A. Altay and D. Cinar, "Fuzzy decision trees," in *Fuzzy Statistical Decision-Making: Theory and Applications*, C. Kahraman and zgr Kabak, Eds., vol. 343, 2016, pp. 221–261.

[16] Y. Lertworaprachaya, Y. Yang, and R. John, "Interval-valued fuzzy decision trees," in *IEEE International Conference on Fuzzy Systems*, Barcelona, Spain, 18-23 July 2010, pp. 1–7.

[17] J. R. Quinlan, "Simplifying decision trees," *International Journal of Man-Machine Studies*, vol. 27, pp. 221–234, 1987.

[18] F. Esposito, D. Malerba, and G. Semeraro, "Simplifying decision trees by pruning and grafting: New results," in *8th European Conference on Machine Learning*, vol. 912, Crete, Greece, 25-27 April 1995, pp. 287–290.

[19] H. Liu and M. Cocea, "Fuzzy information granulation towards interpretable sentiment analysis," *Granular Computing*, vol. 2, no. 4, pp. 289–302, 2017.

[20] W. W. Cohen, "Fast effective rule induction," in *Proceedings of the 12th International Conference on Machine Learning*, 1995, pp. 115–123.

[21] P. Symth and R. M. Goodman, "An information theoretic approach to rule induction from databases," *IEEE Transactions on Knowledge and Data Engineering*, vol. 4, no. 4, pp. 301–316, 1992.

[22] 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.

[23] H. Liu and M. Cocea, *Granular Computing Based Machine Learning: A Big Data Processing Approach*. Berlin: Springer, 2018.

[24] M. Lichman, "UCI machine learning repository, http://archive.ics.uci.edu/ml," 2013.