University of Reading

Towards expressive modular rule induction for numerical attributes

Conference or Workshop Item

Accepted Version

Almutairi, M., Stahl, F. ORCID: https://orcid.org/0000-0002-4860-0203, Jennings, M., Le, T. and Bramer, M. (2016) Towards expressive modular rule induction for numerical attributes. In: Thirty-sixth SGAI International Conference on Artificial Intelligence, 13-15 DECEMBER 2016, Cambridge, UK, pp. 229-235. Available at https://centaur.reading.ac.uk/68358/

It is advisable to refer to the publisher's version if you intend to cite from the work. See Guidance on citing.
Published version at: http://dx.doi.org/10.1007/978-3-319-47175-4_16

Publisher: Springer International Publishing

All outputs in CentAUR are protected by Intellectual Property Rights law, including copyright law. Copyright and IPR is retained by the creators or other copyright holders. Terms and conditions for use of this material are defined in the End User Agreement.

www.reading.ac.uk/centaur

CentAUR

University of Reading

Central Archive at the University of Reading

Reading's research outputs online

Towards Expressive Modular Rule Induction for Numerical Attributes

Manal Almutairi, Frederic Stahl, Mathew Jennings, Thien Le, Max Bramer

Abstract The Prism family is an alternative set of predictive data mining algorithms to the more established decision tree data mining algorithms. Prism classifiers are more expressive and user friendly compared with decision trees and achieve a similar accuracy compared with that of decision trees and even outperform decision trees in some cases. This is especially the case where there is noise and clashes in the training data. However, Prism algorithms still tend to overfit on noisy data; this has led to the development of pruning methods which have allowed the Prism algorithms to generalise better over the dataset. The work presented in this paper aims to address the problem of overfitting at rule induction stage for numerical attributes by proposing a new numerical rule term structure based on the Gauss Probability Density Distribution. This new rule term structure is not only expected to lead to a more robust classifier, but also lowers the computational requirements as it needs to induce fewer rule terms.

1 Introduction

The classification of unseen data instances (Predictive Analysis) is an important data mining task. Most classification techniques are based on the 'divide-and-conquer' approach to generate decision trees as detailed in [6]. Many variations exist despite the inherent weakness in that decision trees often require irrelevant tests to be included in order to perform classification tasks [3]. Cendrowska's Prism algorithm

Manal Almutairi, Frederic Stahl, Mathew Jennings and Thien Le
Department of Computer Science, University of Reading, PO Box 225, Whiteknights, Reading, RG6 6AY, UK, e-mail: manal.almutairi@pgr.reading.ac.uk, F.T.Stahl@reading.ac.uk, m.jennings@student.reading.ac.uk, t.d.le@pgr.reading.ac.uk

Max Bramer
University of Portsmouth, School of Computing, Buckingham Building, Lion Terrace, PO1 3HE, UK, e-mail: Max.Bramer@port.ac.uk

addresses this issue through a 'separate-and-conquer' approach, which generates if-then rules directly from training instances [3]. Cendrowska's Prism algorithm sparked work on a range of different Prism variations that aim to improve upon Cendrowska's original algorithm. The original Prism algorithm was not designed to work with numerical attributes but an extended version of the algorithm with the ability to deal with numerical attributes was described in [1] by incorporating local discretisation. PrismTCS which is a computationally more efficient version of Prism aims to introduce an order in the rules it induces [2]. PMCRI is a parallel implementation of PrismTCS and Prism [7] etc. In addition there are various pruning methods for the Prism family of algorithms that aim to reduce the induced ruleset from overfitting, i.e. J-pruning [2] and Jmax-pruning [8]. Another advantage of the Prism family of algorithms compared with decision trees is that Prism algorithms are more expressive. Prism rules avoid unnecessary rule terms and thus are less confusing and more expressive compared with decision trees. Algorithms of the Prism family may not overfit as much as decision trees on the training data, nevertheless, they still tend to overfit if there is noise in the data, especially for numerical values. A recent development of one of the co-authors of this paper uses Gauss Probability Density Distribution in order to generate more expressive rule terms for numerical attributes. This development was part of a real-time rule based data stream classifier [4] and resulted in more expressive rulesets and faster real-time rule induction. This paper presents the ongoing work on a new member of the Prism family of algorithms that makes use of the rule term structure introduced in [4]. The new version of Prism based on Gauss Probability Density Distribution is expected to produce more expressive rulesets, produce them faster and with less overfitting compared with its predecessor N-Prism, especially on noisy training data.

This paper is organised as follows: Section 2 introduces Prism, Section 3 describes and positions our ongoing development of a new version of Prism based on Gauss Probability Density Distribution. Section 4 provides an initial empirical evaluation of G-Prism in comparison with Prism. Section 5 provides concluding remarks including ongoing developments of this new version of Prism.

2 Modular Rule Induction with Prism

One of the major criticisms of decision trees is the replicated subtree problem. The problem has been first highlighted in [3] and been given the name replicated subtrees in [9].

Assume a fictitious example training data set comprises four attributes *a, b, c* and *d* and two possible classifications *go* and *stop*. Each attribute can take 2 possible values *True* (T) and *False* (F). And the pattern encoded are the two following rules leading to classification *go*, whereas the remaining cases would lead to classification *stop*:

IF (a) AND (b) → go
IF (c) AND (d) → go

Using a tree induction approach to classify cases *go* and *stop* would lead to the replicated subtree problem illustrated in Figure 1, whereas Prism can induce these two rules directly without adding any unnecessary rule terms.

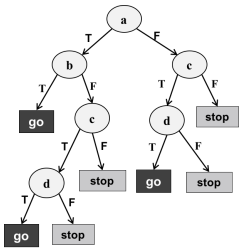

Fig. 1: Replicated subtree problem.

Cendrowska's original Prism algorithm is described in Algorithm 1. Essentially Prism tries to maximise the conditional probability with which a rule covers a particular target class in order to specialise a classification rule. The specialisation stops once the rule only covers instances of the target class. Once a rule is complete Prism deletes all instances from the data that are covered by the rules induced so far and starts inducing the next rule. This process is repeated for every possible target class. Note that Cendrowska's original Prism algorithm [3] does not deal with numerical attributes, however, implementations of algorithms of the Prism family handle numerical attributes as described in lines 7-12 in Algorithm 1. This is very inefficient because there is a large number of probability calculations, in particular $N \cdot m \cdot 2$, where N is the number of training instances and m the number of numerical attributes. Section 3 introduces G-Prism which handles the induction of rule terms from numerical attributes in a more efficient way.

Algorithm 1: Learning classification rules from labelled data using Prism.

```
1  for i = 1 → C do
2      D ← Dataset;
3      while D does not contain only instances of class ω_i do
4          forall the attributes α_j ∈ D do
5              if attribute α_j is categorical then
6                  Calculate the conditional probability, ℙ(ω_i|α_j = v) for all possible attribute-value (α_j = v) from attribute α_j;
7              else if attribute α_j is numerical then
8                  sort D according to v;
9                  forall the values v of α_j do
10                     for each v of α_j calculate ℙ(ω_i|α_j ≤ v) and ℙ(ω_i|α_j > v);
                   end
12             end
           end
15         Select the (α_j = v), (α_j > v), or (α_j ≤ v) with the maximum conditional probability as a rule term ;
16         D ← S, create a subset S from D containing all the instances covered by selected rule term at line 15;
       end
18     The induced rule R is a conjunction of all selected (α_j = v), (α_j > v), or (α_j ≤ v) at line 15;
19     Remove all instances covered by rule R from original Dataset;
20     repeat
21         lines 3 to 9;
       until all instances of class ω_i have been removed;
   end
```

3 Inducing Expressive Module Classification Rules

As described in Section 2, the way Prism deals with numerical attributes requires many cut-point calculations to work out the conditional probabilities for each attribute value, which is computationally expensive. The same critique has been made in [4] for a real-time rule based data stream classifier and a new kind of rule term for numerical streaming data has been proposed. This new rule term is computationally less demanding and also more expressive. This paper proposes to incorporate the in [4] presented rule term structure into the Prism family of algorithms in order to create a more expressive, computationally efficient and robust (to noisy data) Prism classifier.

Instead of creating two rule terms for every possible value in an attribute, only one rule term of the form $(v_i < \alpha_j \leq v_k)$ is created where v_i and v_k are two attribute values. Thus frequent cut-point calculations can be avoided. Additionally, this form of rule term is more expressive than the representation of a rule term for a numerical attribute of $(\alpha_j \leq v)$ or $(\alpha_j > v)$. A rule term in the form of $(v_i < \alpha_j \leq v_k)$, can describe an interval of data whereas the previous approach described in Algorithm 1 would need two rule terms and thus result in longer less readable rules and rulesets.

For each normally distributed numerical attribute in the training data, a Gaussian distribution is created to represent all values of a numerical attribute for a given target class. The most relevant value of a numerical attribute for a given target class ω_i can be extracted from the generated Gaussian distribution of that class. The Gaussian distribution is calculated for the numerical attribute α_j with mean μ and variance σ^2 from all the values associated with classification, ω_i. The conditional probability for class ω_i is calculated as in Equation 1.

$$\mathbb{P}(\alpha_j = v|\omega_i) = \mathbb{P}(\alpha_j = v|\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} exp(-\frac{((\alpha_j = v) - \mu)^2}{2\sigma^2}) \quad (1)$$

Hence, a value based on $\mathbb{P}(\omega_i|\alpha_j = v)$, or equivalently $log(\mathbb{P}(\omega_i|\alpha_j = v))$ can be calculated as shown in the Equation 2. This value can be used to determine the probability of a given class label, ω_i for a valid value, v of a numerical attribute, α_j.

$$log(\mathbb{P}(\omega_i|\alpha_j = v)) = log(\mathbb{P}(\alpha_j = v|\omega_i)) + log(\mathbb{P}(\omega_i)) - log(\mathbb{P}(\alpha_j = v)) \quad (2)$$

Based on the created Gaussian distribution for each class label, the probability between a lower bound and an upper bound, Ω_i can be calculated for such that if $v \in \Omega_i$, then v belongs to class ω_i. Practically, from a generated Gaussian distribution for class ω_i, a range of values μ in the centre is expected to represent the most common values of the numerical attribute for class ω_i. However, if an algorithm is based on 'separate-and-conquer' strategy then the training data examples are different after each iteration and the bounds should be selected from the available values of the numerical attribute from a current iteration.

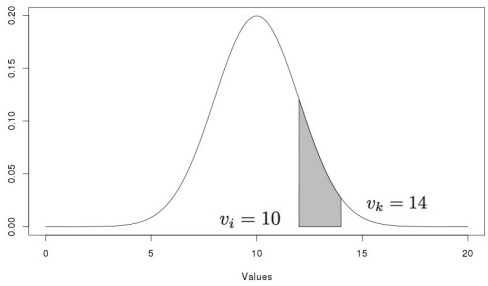

Fig. 2: The shaded area represents a range of values, Ω_i, of continuous attribute α_j for class ω_i.

As shown in Figure 2, the shaded area between lower bound, v_i and upper bound, v_k represents the most common values of the numerical attribute α_j for class ω_i from a given subset of training data examples. The selection of an appropriate range of the numerical attribute is simply to identify a possible rule term in the form of $(v_i < \alpha_j \leq v_k)$, which is highly relevant to a range of values from the numerical attribute α_j for a target class ω_i from the training data examples. Once $\mathbb{P}(\omega_i | v_i < \alpha_j \leq v_k)$ is calculated for each numerical attribute then the Prism learning algorithm selects the rule term with highest conditional probability from both categorical and numerical attributes. The resulting algorithm is termed G-Prism where G stands using Gaussian distribution. Please note that data stream classifiers deal with an infinite amount of numerical values per attribute and hence typically assume a normal distribution, however, this is not the case for batch algorithms like Prism. A test of normal distribution will be applied prior to applying this method. Attributes that are not normal distributed would be dealt with as described Section 2.

As has been shown in [4], the new rule term structure is fitting the target class less exact and covers more training data, whilst still achieving high accuracy. This resulted in less overfitting [4] compared rule terms induced through binary splits. Thus this new rule term structure incorporated in Prism is expected to be also less prone to overfitting in comparison with binary splits on numerical attributes.

4 Preliminary Results

The first prototype version of the G-Prism algorithm has been compared in terms of classification accuracy against its Prism predecessor as described in Algorithm 1. For the comparison two datasets have been used, the Glass and the Iris dataset from the UCI repository [5]. The datasets have been randomly split into a train and test datasets, whereas the testset comprises 30% of the data.

Table 1: Comparison of G-Prism's accuracy with Prism's accuracy

	Datasets	
	Glass	Iris
Accuracy G-Prism	56	93
Accuracy Prism	48	76

Both algorithms seem to struggle on the Glass dataset, however, G-Prism shows a higher accuracy on both datasets. Please note that both algorithms are expected to perform better when pruning is incorporated, which is subject to ongoing work.

5 Conclusions and Future Work

The paper positioned the ongoing work on a new Prism classifier using a different more expressive way of inducing rule terms. The new rule term induction method is based on Gauss Probability Density Distribution and is expected to reduce the number of probability calculations and thus lower computational cost. The new method is also expected to be more robust to overfitting as the new form of rule term should cover a larger amount of examples and thus generalise better. An initial evaluation shows an improvement in classification accuracy. Ongoing work is implementing a pruning facility for both Prism and G-Prism. Thus a more exhaustive comparative evaluation in terms of accuracy, robustness to noise and computational scalability is planned on more datasets.

References

1. M. Bramer. *Principles of Data Mining*. Undergraduate Topics in Computer Science. Springer International Publishing, 2013.
2. M A Bramer. An information-theoretic approach to the pre-pruning of classification rules. In B Neumann M Musen and R Studer, editors, *Intelligent Information Processing*, pages 201–212. Kluwer, 2002.
3. Jadzia Cendrowska. PRISM: An algorithm for inducing modular rules, 1987.
4. Thien Le, Frederic Stahl, João Bártolo Gomes, Mohamed Medhat Gaber, and Giuseppe Di Fatta. Computationally efficient rule-based classification for continuous streaming data. In *Research and Development in Intelligent Systems XXXI*, pages 21–34. Springer, 2014.
5. M. Lichman. UCI machine learning repository, 2013.
6. Ross J Quinlan. Induction of decision trees. *Machine Learning*, 1(1):81–106, 1986.
7. Frederic Stahl and Max Bramer. Computationally efficient induction of classification rules with the {PMCRI} and j-pmcri frameworks. *Knowledge-Based Systems*, 35:49 – 63, 2012.
8. Frederic Stahl and Max Bramer. Jmax-pruning: A facility for the information theoretic pruning of modular classification rules. *Knowledge-Based Systems*, 29(0):12 – 19, 2012.
9. I.H. Witten, E. Frank, and M.A. Hall. *Data Mining: Practical Machine Learning Tools and Techniques: Practical Machine Learning Tools and Techniques*. The Morgan Kaufmann Series in Data Management Systems. Elsevier Science, 2011.