A Fuzzy Beam-Search Rule Induction Algorithm

Cristina S. Fertig[1], Alex A. Freitas[2], Lucia V. R. Arruda[3], Celso Kaestner[4]

[1, 3, 4] CEFET-PR, CPGEI. Av. Sete de Setembro, 3165
Curitiba – PR. 80230-901. Brazil.
csfertig@dainf.cefetpr.br, arruda@cpgei.cefetpr.br, kaestner@dainf.cefetpr.br

[2, 4] PUC-PR, PPGIA-CCET. Rua Imaculada Conceição, 1155
Curitiba – PR. 80215-901. Brazil.
{alex, kaestner}@ppgia.pucpr.br http://www.ppgia.pucpr.br/~alex

Abstract. This paper proposes a fuzzy beam search rule induction algorithm for the classification task. The use of fuzzy logic and fuzzy sets not only provides us with a powerful, flexible approach to cope with uncertainty, but also allows us to express the discovered rules in a representation more intuitive and comprehensible for the user, by using linguistic terms (such as low, medium, high) rather than continuous, numeric values in rule conditions. The proposed algorithm is evaluated in two public domain data sets.

1 Introduction

This paper addresses the classification task. In this task the goal is to discover a relationship between a goal attribute, whose value is to be predicted, and a set of predicting attributes. The system discovers this relationship by using known-class examples, and the discovered relationship is then used to predict the goal-attribute value (or the class) of unknown-class examples.

There are numerous rule induction algorithms for the classification task. However, the vast majority of them work within the framework of classic logic. In contrast, this paper proposes a fuzzy rule induction algorithm for the classification task. The motivation for this work is twofold. First, fuzzy logic is a powerful, flexible method to cope with uncertainty. Second, fuzzy rules are a natural way to express rules involving continuous attributes. Actually, rule induction algorithms implicitly perform a kind 'hard' (rather than soft) discretization when they cope with continuous attributes. For instance, within the framework of classic logic a rule induction algorithm discovers rules containing conditions such as 'age < 25', which has the obvious disadvantage of not coping well with the borderline ages of 24 and 25. In contrast, a fuzzy rule can contain a condition such as 'age = young', where young is a fuzzy attribute describing persons around 25 years old. This concept is more natural and more comprehensible for the user.

In principle any rule induction method can be 'fuzzyfied'. Indeed, some fuzzy decision tree algorithms have been proposed in the past [3], [2]. In this paper we have chosen, as the underlying data mining algorithm to be fuzzyfied, a variant of the beam-search rule induction algorithm described in [8]. Our motivation for this choice is as follows. Despite their popularity, most decision tree algorithms perform a kind of hill-climbing search for rules, by exploring just one alternative at a time, which makes them very sensitive to the problem of local maxima. Beam search algorithms have the advantage of being less sensitive to this problem, since they explore w alternatives at a time, where w is the beam width [13]. To the best of our knowledge, this paper is the first work to propose a fuzzy beam search-based rule induction algorithm.

2 Beam Search-Based Rule Induction

The basic idea of a beam search-based rule induction algorithm [8] is described in Figure 1. The algorithm receives two arguments as input, namely max_depth (the maximum depth of the search tree) and w, the beam width (the number of rules or tree paths being explored by the algorithm at a given time). Both parameters are small integer numbers. In Figure 1, R_i denotes the i-th rule, i=1,…,w, A_j denotes the j-th predicting attribute, v_{jk} denotes the k-th value of the j-th predicting attribute, and R_{ijk} denotes the new rule creating by adding condition $A_j = v_{jk}$ to the rule R_i.

This is a top-down algorithm, which starts with an 'empty rule' with no condition, and iteratively adds one condition at a time to all the current rules, so specializing the rules. Each condition added to a rule is of the form 'Attribute = Value', where Value is a value belonging to the domain of the attribute. (The operator could be '>' or '<' in the case of continuous attributes. However, for simplicity we consider here only the '=' operator, which is the only one necessary for our proposed fuzzy version of the algorithm.) In each iteration the algorithm tries to add all possible conditions to each rule (as long as the corresponding attribute does not occur yet in the rule), evaluates the just-created rules according to a rule-quality measure, and chooses the best w rules to proceed the search. This process is repeated until either no just-created rule is better than any rule of the previous iteration or the maximum depth has been reached.

Note that the algorithm expands a tree search where the root node is a rule with no conditions, each path from the root until a current leaf node corresponds to a current rule, and the depth of the path corresponds to the number of conditions in the rule. Therefore, the parameter max_depth effectively specifies the maximum number of conditions in a rule.

To compute the quality of a rule we used a variant of the well-known confidence factor (CF) measure. Let A => C denote a rule, where A is the rule antecedent (a conjunction of conditions) and C is the rule consequent (the valued predicted for the goal attribute). The CF measure is simply $|A \& C| / |A|$, where $|x|$ denotes the cardinality of set x. In other words, CF is the ratio of the number of examples that both satisfy the conditions in the rule antecedent and have the goal-attribute value predicted by the rule consequent over the number of examples satisfying the conditions in the rule antecedent. We borrowed from [10] the idea of using a variant of this measure defined as: $(|A \& C| - ½) / |A|$. The motivation for subtracting ½ from the numerator is to favor the discovery of more general rules, by avoiding the overfitting of the rules to the data. For instance, consider two rules R_1 and R_2, where R_1 has $|A \& C| = |A| = 1$ and R_2 has $|A \& C| = |A| = 100$. Without the ½ correction, both rules have a CF = 100%. However, rule R_1 is probably overfitting the data, and rule R_2 is more likely to be accurate on unseen data. With the ½ correction we achieve the more realistic CF measures of 50% and 99.5% for rules R_1 and R_2, respectively.

```
Input: max_depth, w;
depth := 0;
RuleSet = a single rule with no conditions;
REPEAT
  FOR EACH rule R_i in RuleSet
    FOR EACH attribute A_j not used yet in rule R_i
      Specialize R_i by adding a condition A_j = v_jk to the rule,
      and call the new rule R_ijk;
      Compute a rule-quality measure for R_ijk;
    END FOR
END FOR
RuleSet = the best w rules among all current rules;
  depth := depth + 1;
UNTIL (no rule created in this iteration is better than
any rule of previous iteration) OR (depth = max_depth)
```

Fig. 1. Overview of a Beam Search-based Rule Induction Algorithm

Finally, note that the algorithm of Figure 1 is searching for conditions to add to the rule antecedent only. It does not specify how to form the rule consequent. This part of the rule contains the value (class) predicted for the goal attribute. The choice of the class predicted by a rule can be made in several ways. One approach would be to let the algorithm automatically choose the best class to form the rule consequent, by picking the class to which the majority of the examples satisfying the rule antecedent belong. Another approach would be to run the algorithm k times for a k-class problem, where in each run the algorithm searches only for rules predicting the k-th class. Yet another approach, assuming a two-class problem, is to run the algorithm to discover only rules predicting one class. In this case, if an example satisfies any of the discovered rules it is assigned the corresponding class; otherwise it is assigned the other class (the 'default' class).

We have opted for this latter approach. In our work the algorithm searches only for rules predicting the minority class (i.e. the class with smaller relative frequency in the data being mined), whereas the majority class is the default class. We have chosen this approach for two reasons. First, its simplicity. Second, focusing on the discovery of rules predicting the minority class has the advantage that these rules tend to be more interesting to the user than rules predicting the majority class [5]. For instance, rules predicting a rare disease tend to be more interesting than rules predicting that the patient does not have that disease. (Of course, one of the reasons why minority-class rules tend to be more interesting is that it is more difficult to discover them, in comparison with majority class rules.)

3 Fuzzyfying a Beam Search-based Rule Induction Algorithm

In this section we describe how we have fuzzyfied the algorithm described in the previous section. First of all, as a pre-processing step, for each continuous attribute in the data being mined we must associate: (1) a set of fuzzy linguistic terms (such as high, medium, low), which are essentially labels for fuzzy sets; and (2) a fuzzy membership function, which specifies the degree to which each attribute's original value belongs to the attribute's linguistic terms. Loosely speaking, this can be regarded as a kind of 'discretization', since the originally-continuous attribute can now take on just a few linguistic terms as its value. However, this is a very flexible discretization, since each of the now 'discrete' values of the attribute is actually a flexible linguistic term corresponding to a fuzzy set. The continuous attributes were fuzzyfied by using trapezoidal membership functions [1], since this kind of function often leads to a data modeling closer to reality. Note that it is necessary to fuzzyfy only continuous attributes, and not categorical ones.

In addition to the above-described fuzzyfication of continuous attributes, we also need to fuzzyfy the computation of the rule-quality measure. In our case this corresponds to fuzzyfying the formula $(|A \& C| - ½) / |A|$, defined in the previous section. In our first attempt to define a fuzzy version of $|A|$, this term was the summation, over all training examples, of the degree to which the example satisfies the rule antecedent. For each example, this degree is computed by a fuzzy AND of the degree to which the example satisfies each condition. We have used the conventional definition of the fuzzy AND as the minimum operator. For instance, suppose a rule antecedent contains the following three conditions 'age = young and salary = low and sex = male', where age and salary are fuzzyfied (originally continuous) attributes and sex is a categorical attribute. Suppose that a given training example satisfies the condition 'age = young' to a degree of 0.8, the condition 'salary = low' to a degree of 0.6 and the condition 'sex = male' to a degree of 1. In this case the example satisfies the rule antecedent to a degree of 0.6 (minimum value among 0.8, 0.6 and 1). Note that if the example had 'sex = female' it would satisfy the above rule antecedent to a degree of 0. (Conditions involving categorical attributes are satisfied to a degree of either 0 or 1.)

However, the above fuzzyfication of $|A|$, although theoretically sound, has a disadvantage. In practice, many training examples can satisfy a rule antecedent to a small degree, and the summation of all these small degrees of membership can undermine the reliability of the CF measure. To avoid this, our summation of the degree to which an example satisfies the rule antecedent includes only the examples with a degree of membership greater than or equal to a predefined threshold (set to 0.5 in our experiments), rather than all training examples. This operation is based on the alpha-cut technique used in fuzzy arithmetic [9].

In our fuzzy version, $|A \& C|$ is the summation, over all examples that have the class predicted by the rule, of the degree to which the example satisfies the rule antecedent. Analogously to the computation of $|A|$, this summation considers only examples which satisfy the rule antecedent to a degree greater than or equal to the above-mentioned membership threshold.

4 Computational Results

The above described fuzzy rule induction algorithm was evaluated on two public domain data sets, available from http://www.ics.uci.edu/~mlearn/MLRepository.html. The first data set used in our experiments was the Pima Indians Diabetes Database [11], which contains 9 continuous attributes and 768 examples. The second data set was the Boston Housing Data [7], which contains 13 continuous attributes and 1 binary attribute. The latter was removed from the data set for the purposes of our experiments, since only continuous attributes are fuzzyfied by our algorithm. This data set contains 506 examples. The goal attribute for this data set (median value of owner-occupied homes in $1000's) was originally continuous. To adapt this data set to the classification task, the goal attribute was discretized, so that it can take on only two classes (cheap and expensive).

Each data set was divided into 5 partitions, and a cross-validation procedure [6] was then performed. For each data set, this corresponds to run the algorithm 5 times, where in each time a different partition is used as the test set and all the remaining four partitions are used as the training set.

In all our experiments the maximum depth of the tree search was set to 2 and the beam width w was set to 10. Almost all continuous attribute were fuzzyfied into 3 linguistic values, each represented by a trapezoidal membership function. The only exceptions were the attributes *Rad* and *Tax* of the housing data set, which were fuzzyfied into two linguistic values.

The results of our experiments are reported in Table 1. Each cell of the table refers to the average results over the 5 runs of the algorithm associated with the cross-validation procedure. The first row indicates the baseline accuracy of each data set, that is the relative frequency of the majority class. A basic requirement for the predictive accuracy of a classifier is that it be greater than the baseline accuracy, since it would be trivial to achieve such accuracy by always predicting the majority class.

The second row indicates the overall predictive accuracy achieved by our fuzzy algorithm. This was computed as the ratio of the number of correctly classified test examples over the total number of test examples. Note that for both data sets the algorithm achieved an accuracy rate significantly better than the baseline accuracy.

Note that the above overall accuracy rate only takes into account whether the classification of an example was correct or wrong, regardless of which kind of rule (a discovered rule or the default-class rule) was used to classify the example. To analyze in more detail the quality of the discovered rules, we also measured separately the accuracy rate of the discovered rules and the accuracy rate of the default-class rule. Recall that all discovered rules predict the same class, namely the minority class, whereas the default-class rule, which is applied whenever the test example does not satisfy any discovered rule, simply predicts the majority class. Recall also that an example is considered to satisfy a discovered rule only if the rule antecedent belongs to the alpha-cut membership function describing the linguistic terms (with alpha = 0.5 in our experiments), as explained in section 3.

The third row of Table 1 indicates the accuracy rate of the discovered rules, computed as the ratio of the number of examples correctly classified by the discovered rules over the number of examples satisfying any of the discovered rules. (Since all discovered rules predict the same class, it is irrelevant which rule is actually classifying the example.)

The fourth row indicates the accuracy rate of the default-class rule, computed as the ratio of the number of examples correctly classified by this rule over the number of examples classified by this rule (i.e. the number of examples that do not satisfy any of the discovered rules).

As can be seen in the table, in both data sets the default rule has a predictive accuracy significantly better than the discovered rules. This was somewhat expected, given the difficulty of predicting the minority class in both data sets. Actually, if we were to compute the 'baseline accuracy of the minority class', we would find the values 0.349 (1 – 0.651) and 0.245 (1 – 0.755) for the Diabetes and Housing data sets, respectively. From this point of view, the discovered rules can still be considered as good-quality ones with respect to predictive accuracy, since their accuracy is 0.711 and 0.838 for the Diabetes and Housing data sets, respectively.

Table 1. Computational Results

	Diabetes data set	Housing data set
Baseline accuracy	0.651	0.755
Overall accuracy	0.711	0.838
Accuracy of discovered rules	0.641	0.725
Accuracy of default rule	0.730	0.863

Another issue to be considered is the comprehensibility of the discovered rules. Although this is a subjective criterion, it is common in the literature to evaluate comprehensibility in terms of the syntactical simplicity of the discovered rules. In this case, the smaller the number of rules and the smaller the number of conditions per rule, the more comprehensible the discovered rule set is.

With this definition of comprehensibility we can say that the rules discovered by our algorithm are comprehensible, almost by definition of the algorithm and a suitable choice of its parameters. Firstly, the algorithm discovers only a small set of rules, whose number is the user-specified beam width. Secondly, the algorithm discovers only relatively short rules, where the number of conditions of the discovered rules is at most the user-specified maximum depth of the search tree. In addition, and more important, the use of linguistic terms associated with our fuzzy algorithm can be considered as a form of improving rule comprehensibility, in comparison with continuous, numeric values, as argued in the introduction.

It should be noted, however, that high accuracy and comprehensibility do not necessarily imply interestingness. To consider a classical example, in a hospital database we can easily mine a rule such as 'if (patient is pregnant) then (patient sex is female)'. Although this rule is highly accurate and comprehensible, it is obviously uninteresting, since it states the obvious.

Our algorithm discovers only rules predicting the minority class, which, as argued in section 3, tend to be more interesting rules for the user. However, a more detailed analysis of the degree of interestingness of the discovered rules is beyond the scope of this paper. Although the literature proposes several methods to evaluate the degree of interestingness of the discovered rules – see e.g. [4], [12] – it does not seem trivial to adapt these methods to the context of fuzzy rules.

5 Conclusion

We have proposed a fuzzy version of a beam search-based rule induction algorithm. We have also evaluated the algorithm on two data sets. Overall, the results are good, not only with respect to predictive accuracy, but also (and more importantly, in data mining) with respect to the comprehensibility of the discovered rules, which is intuitively improved by the use of linguistic terms associated with fuzzy rules.

However, the computational results reported in this paper are still somewhat preliminary, since the algorithm has been applied to only two data sets. Future research will include a more extensive evaluation of the algorithm on other data sets.

A disadvantage of our fuzzy approach is that it requires a preprocessing phase to specify the fuzzy membership for each continuous attribute. To solve this problem two approaches can be used: (1) this specification is done with the help of the user (an expert in the meaning of the data), which requires valuable human time; (2) a fuzzy clustering algorithm can be used to get the membership function [9].

Hopefully, in the future fuzzy databases will be more commonplace, so that the fuzzy values and fuzzy membership functions that our algorithm requires might be already available in the underlying fuzzy database, so avoiding the need for the above preprocessing phase.

References

1. Bojadziev, G., Bojadziev, M.: Fuzzy sets, Fuzzy logic, Applications. World Scientific (1995)
2. Chi, Z., Yan, H.: ID3-derived fuzzy rules and optimized defuzzification for handwritten numeral recognition. *IEEE Trans. On Fuzzy Systems¸4(1),* Feb. (1996) 24-31
3. Cios, K.J., Sztandera, L.M.: Continuous ID3 algorithm with fuzzy entropy measures. *Proc. IEEE Int. Conf. Fuzzy Systems*,. San Diego (1992) 469-476
4. Freitas, A.A.: On objective measures of rule surprisingness. Principles of Data Mining and Knowledge Discovery (Proc. 2nd European Symp., PKDD-98). Lecture Notes in Artificial Intelligence 1510. Springer-Verlag (1998) 1-9
5. Freitas, A.A.: On rule interestingness measures. *To appear in Knowledge-Based Systems journal* (1999)
6. Hand, D.: Construction and Assessment of Classification Rules. John Wiley & Sons (1997)
7. Harrison, D., Rubinfeld, D.L.: UCI Repository of machine learning databases. [http://www.ics.uci.edu/~mlearn/MLRepository.html]. Irvine, CA: University of California, Dept. of Information and Computer Science (1993)
8. Holsheimer, M., Kersten, M., Siebes, A.: Data surveyor: searching the nuggets in parallel. In: U.M. Fayyad et al. (Eds.) *Advances in Knowledge Discovery and Data Mining*. AAAI Press (1996) 447-467
9. Predrycz, W., Gomide, F.: An Introduction to Fuzzy Sets Analysis and Design. MIT (1998)
10. Quinlan, J.R.: Generating production rules from decision trees. *Proc. Int. Joint Conf. AI (IJCAI-87)* (1987) 304-307
11. Sigillito, V.: UCI Repository of machine learning databases. [http://www.ics.uci.edu/~mlearn/MLRepository.html]. Irvine, CA: Univ. of California, Dept. of Information and Computer Science (1990)
12. Suzuki, E., Kodratoff, Y.: Discovery of surprising exception rules based on intensity of implication. Principles of Data Mining and Knowledge Discovery (Proc. 2nd European Symp., PKDD-98). Lecture Notes in Artif. Intelligence 1510. Springer-Verlag (1998) 10-18
13. Winston, P.H.: *Artificial Intelligence*. 3rd Ed. Addison-Wesley (1992)