2009 Ninth International Conference on Intelligent Systems Design and Applications

The Impact of Pruning BayesFuzzy Rule Set

I-Hsien Yin[1], Estevam R. Hruschka Jr[1,2] and Heloisa de A. Camargo[1]

[1]DC/UFSCar – Federal University of São Carlos, Brazil, Rod. Washington Luís, Km 235

Mailbox 676, CEP 13565-905 São Carlos-SP

[2]Carnegie Mellon University, Machine Learning Department, 5000 Forbes Avenue, Pittsburgh, PA, USA

ishien224@hotmail.com, estevam@dc.ufscar.br, heloisa@dc.ufscar.br

Abstract: **The use of Bayesian Network Classifiers (BCs) combined with the Fuzzy rule model to explain the learned BCs have been previously presented as the BayesFuzzy approach. This paper follows along BayesFuzzy lines of investigation aiming at improving the comprehensibility of a BC model and enhancing BayesFuzzy results by combining new pruning methods. In order to improve BayesFuzzy performance, in addition to the Markov Blanket-based pruning idea used by BayesFuzzy, two other pruning methods are proposed, implemented and empirically evaluated. The first pruning method is based on the conditional probability estimates given by the BC and the second one is the well-known post-rule pruning approach, usually used to prune rules extracted from decision trees. Also, three different Bayesian Networks induction algorithms, namely IC, K2 and Naïve-Bayes, as well as, the C4.5 Decision Tree induction algorithms are employed in the empirical comparative analysis performed in the experiments. The obtained results reveal that BayesFuzzy combined with the new pruning methods can bring comprehensibility enhancements.**

I. INTRODUCTION

Bayesian Networks (BNs) are graphical representations of the joint probability distribution of a set of variables and have been successfully used as classifiers in many application domains. When concerning interpretability, however, the knowledge encoded by a BN is not as comprehensible as some other classification models, such as rule-based and/or Fuzzy classifiers.

Fuzzy Classification Systems (FCS), also known as Fuzzy classifiers, are based on Fuzzy classification rules and are designed to perform a classification task that requires the attribute domains to be granulated by means of fuzzy partitions. Fuzzy classifiers are ideally suited to provide good classification accuracy and comprehensible solutions to users, since they handle imprecise data and perform classification based on a set of rules which are interpretable, i.e. the semantic structure provides insight into the classifier structure and decision making process.

In areas such as Data mining and Decision Support Systems, interpretability can play an important role. In those areas, it is important to have the data characteristics represented as symbolic rules or any other form of knowledge representation that promotes understandability. Following along these lines, some previous works proposed and discussed some methods to translate BNs into sets of crisp [10] [18] and Fuzzy classification rules [1][9]. Doing so, one can achieve a Bayes/Fuzzy hybrid system taking advantage of the capability of Bayesian Networks to identify relevant relationships among variables and its usually high accuracy in classification tasks allied to the high comprehensibility of rule base Fuzzy Systems.

In spite of understanding the importance and being in agreement with the validity of the aforementioned hybrid methods, a deep analysis of them reveals that it is worth to further investigate such previous proposed Bayes/Fuzzy collaborations in order to try to answer some important questions regarding the accuracy and interpretability of such approaches as:

i) can a pruning strategy help improving the comprehensibility of the model while maintaining its classification accuracy?

ii) What is the influence of specific data characteristics (e.g. redundant and irrelevant variables) on the classification accuracy and interpretability?

iii) Which BC learning algorithm is most appropriate to the specific Bayes/Fuzzy hybrid system BayesFuzzy?

Therefore, in this work pruning methodologies (applied to the BayesFuzzy method described in [9]) is proposed, implemented and discussed. In addition, an empirical evaluation of specific data characteristics is conducted and the results are analyzed considering classification accuracy and interpretability of the classification models.

The sequence of this document is organized as follows: Section 2 gives a brief overview of some related works. Section 3 reviews the BayesFuzzy algorithm, describes the pruning strategy designed to optimize the classifier interpretability and shows how to perform the pruning before executing the classification. Section 4 starts describing the particular experimental scenarios we created to the empirical evaluation of the method and then, shows the results of the application of the proposed pruning strategy in different datasets having different characteristics. Finally, Section 5 brings the conclusions and points out some future works.

II. BAYESFUZZY AND ITS PRUNED VERSION

The *Pruned BayesFuzzy* (PBF), is a variation of the *BayesFuzzy* (BF) defined in [9]. The main difference between the two methods is that PBF explores pruning strategies to select attributes and rules from the set of Fuzzy classification rules generated by BF. A brief BF

978-0-7695-3872-3/09 $26.00 © 2009 IEEE
DOI 10.1109/ISDA.2009.89

overview is given in the next subsection (2.1).

A. BayesFuzzy

As showed in [9], *BayesFuzzy* is an algorithm suitable to be used in situations when a BN Classifier (BNC) needs to be built from data (D) to be used as input to a *Rule Based Fuzzy System* (RBFS), and the classification model must be comprehensible to human beings. BF algorithm can be summarized in four basic steps:

1) "Discretize" the dataset D using a fuzzyfication process, generating a qualitative dataset D';

2) Learn a Bayesian Classifier (BC) using D';

3) Extract rules from the BC generated in Step 2;

4) Use the fuzzy rule base generated in Step 3 in a fuzzy classification system.

Considering the above four steps, some very interesting research questions can be drawn: i) Which BC learning algorithm is most appropriate to BF? ii) How to optimize the interpretability and precision of the set of rules extracted form the BC? iii) What is the influence of specific data characteristics (e.g. redundant and irrelevant variables) on the classification accuracy and interpretability? iv) which fuzzyfication function is most suitable to BF?

In this work, however, we are mainly focused in trying to answer the first, the second and the third questions regarding the optimization of the rule set.

B. Pruned BayesFuzzy

As stated in the literature, there are many rule interestingness metrics such as support, confidence, lift, correlation, collective strength, etc. Such metrics are often used to determine the more relevant rules from a rule set in a pruning strategy. Many of these measures, however, provide conflicting information about the interestingness of a pattern. Therefore, the best metric to use for a given application domain is hard to define. See [16] for a more detailed description of properties of some of the most commonly used rule interestingness measures.

In the Pruned BayesFuzzy algorithm described in this paper, two pruning strategies are performed (in addition to the Markov Blanket-based pruning and the "RemoveSuperfluousRules" performed by the original BayesFuzzy).

The first one is based on the rule probability estimate given by the BC. It is a very simple idea and is mainly motivated by the fact that it can be applied without any extra computation effort. Considering the rule set given by the original BayesFuzzy as an ordered list of rules (ordered based on the probability estimates), the pruning can be done by only taking into account the rules having probability estimates higher than a predefined threshold. This measure (the rule probability estimate) can be seen as a confidence measure and has three interesting properties:

1. It monotonically increases with P(A,B) when P(A) and P(B) remain the same.
2. When it is symmetrized by taking max(P(A|B);P(B|A)), the measure is Symmetry under variable permutation;
3. It presents null invariance, which is useful for domains having sparse data sets, where co-presence of items is more important than co-absence.

Even knowing that such properties are important for a good interestingness measure [16], we are not claiming that it is the best measure to be used in a rule set pruning task. As aforementioned, different measures have different intrinsic properties. Thus, it is important to notice that, some of these properties may be desirable for certain applications but not for others. Therefore, in order to find the most suitable measure, one must match the desired properties of an application against the properties of the existing measures. As our PBF is not designed for a specific application domain, we don't want to state which pruning method is the best. We are interested, instead, in find a pruning strategy suitable for helping to consistently reduce the number of Fuzzy rules extracted from a Bayesian Classifier in a broad range of domains.

The second pruning strategy used in PBF is a *Rule Post-Pruning* strategy similar to the one used when learning decision Trees from data [15]. It can be summarized in the following steps:

1. Receive a rule base R;
2. For each rule ri in R;
3. Simplify ri by greedily deleting antecedents in order to minimize the rule's estimated classification error rate.
4. Store the simplified version of ri in the pruned rule set R'.
5. Define a default class based on the MAP (Maximum a Posteriori) [12] criterion.

III. EXPERIMENTS AND RESULTS

This section initially describes the adopted experimental setting and then, the results obtained for the assessed algorithms are presented and analyzed.

A. Experimental Setting

Our experimental setting is based on the desire to evaluate the relative performance of the algorithms being studied under controlled conditions. With this purpose in mind, we have conceived some particular experimental scenarios. In brief, for each scenario we have built hypothetical domains of interest which were then used to generate synthetic datasets (SDs). From this standpoint, each of these datasets can be viewed as a particular realization of a given specific situation to be analyzed. Doing so, true characteristics are a priori known for each dataset used in the experiments, allowing us to derive interesting analyses regarding the relative performance of the algorithms under investigation.

We have designed seven different specific situations and for each situation we've generated 4 datasets

containing 1000, 10000, 30000 and 50000 instances each, resulting in 28 datasets. Then, each dataset was used to perform a supervised learning task to induce a Bayesian Network. Next, BayesFuzzy and Pruned BayesFuzzy were applied to each induced Bayesian Network and the obtained Rule Sets.

To better understand how the experiments here described were defined is important to remember that our claim, in this paper, is that it is possible to take advantage of the causal knowledge representation (which is possible in a BN or BC) and the usually good accuracy of Bayesian classifiers to have a set of Fuzzy classification rules (extracted from the BC) as a knowledge base. Even knowing, however, that the main idea of the conducted experiments is not to show that the BF and PBF are better predictors than traditional classifiers or rule extraction methods, this section also brings results of the use of the C4.5 algorithm (as implemented in the J48 method in WEKA [17]) to extract classification rules from the datasets generated for each one of the 7 experimental scenarios.

To generate the datasets for each scenario we used a data generator based on rules that describe the desired domain characteristics[1]. The seven experimental scenarios can be described as follows:

1. *Baseline*: In this domain, we simulate a situation in which all variables are relevant to the classification task and there is no redundant rule. In this sense, there is no noisy data. The generated dataset has 3 continuous and 3 discrete variables. In addition, the class is well balanced. The purpose of this database is to become the base of comparison for other databases.
2. *Irrelevant Variables (IV)*: As happens in the Baseline domain, this (IV) domain also has 6 variables. Instead, however, of having all variables being relevant to the classification task, the IV domain has 4 irrelevant variables and 2 relevant ones. The purpose of this domain is to simulate a situation where there is irrelevant information in the dataset.
3. *Repeated Pattern (RP)*: This domain also has 6 variables, there are, however, 2 redundant rules out of the 8 rules used to describe the dataset. The purpose of this domain is to simulate a situation where one of the data pattern (rule) is stronger than the other ones.
4. *More Generalized Pattern (MGP)*: Equal to the Baseline, except that part of a data pattern is stronger then another. In other words, part of the antecedent of more than one rule is the same. The purpose of this domain is to simulate the situation where there only a small part of a pattern (rule antecedent) can define the class value through BC.
5. *Noisy Data (ND):* Similar to the Baseline, except that in this domain up to 4 variables can be considered noisy data, i.e., don't have a pattern associated with the class variable. The purpose of this domain is to simulate a situation where noisy data is present.
6. *Uncertainty:* Similar to Baseline, except that there are uncertain information, i.e., the same data pattern doesn't necessary have the same class value. The purpose of this domain is to simulate a situation where uncertain information is present.
7. *Unbalanced:* Similar to Baseline, except that the dataset is unbalanced. The obvious purpose of this domain is to simulate a situation where the database is unbalanced;

The first interesting discussion about the performed experimental analysis is related to the *Probability Estimate-based* pruning. In all 7 domains, the use of that pruning strategy brought no relevant results and was not able to generated differences in the classification accuracy, neither in the number of rules generated by BF and PBF. Taking it into account, and also considering the lack of space, no results related to that pruning approach is reported here. Considering, however, that the *Probability Estimate-based* pruning technique was successfully employed in the specific domain of "risk of weed infestation" [1], more investigation is necessary before stating that the technique is not valid to BayesFuzzy. And such results reinforce the idea (presented in section 2) that some pruning techniques might be suitable to specific domains and not for other ones.

Nevertheless, one interesting aspect of these results is that in [1], the BCs' structures were generated by a human expert and only the numerical parameters of the probabilistic models were induced from data. In contrast, all the BCs' structures used in the experiments described in this section were induced from data. Thus, one plausible hypothesis is the following:

Hypothesis 1: Based on the three properties described in section 2.2, the *Probability Estimate* can be seen as a rule interestingness metric very similar to confidence. Thus, when a BC structure is induced from data, the co-presence of items (in the training dataset) will be already used in the BC structure definition and will have less impact in the pruning.

In spite of arguing in favor of the plausibility of Hypothesis 1, we don't have yet enough empirical (nor theoretical) evidence that it is true. Thus, we leave the investigation of the validity of Hypothesis 1 as a future work.

The sequel of this section presents the results obtained using the *Post-Pruning* rule pruning approach. In all the classification tasks performed, the *Average Correct Classification Rate* (ACCR), the average number of rules (#Rules), the average number of antecedents in the rules (#Ant.), as well as, the proximity (P) to the original rules were used as features to our comparative analysis. It is worth to mention that the original rules (OR) are the rules used to generate each dataset and, in this sense, it is considered our *golden set of rules*. In other words, the OR represent the real patterns present in the data, thus, the closer the results gotten using algorithm *A* are to the results obtained using OR, the better algorithm *A* performance is.

The graphics depicted in Fig. 1, Fig. 2 and Fig. 3

[1] To see the set of rules used to generate each dataset for all domains go to http://www.cs.cmu.edu/~estevam/isda2009.html

show (see the bars) the ACCR, the #Rules and #Ant. respectively. The circles in the graphics represent the proximity of the obtained rule sets when compared with the OR, in this sense, the higher the proximity value, the better the results. The proximity circles are exactly the same in all three graphics. In spite of representing the same information in all graphics they are replicated in order to help cross-results analysis.

Considering that we are also interested in starting to investigate which BC learning algorithm is most appropriate to BF, for each dataset, 3 different BC induction algorithms were applied (IC [13], K2 [4] and NaiveBayes (NB) [6]) thus 3 different BC as well as 3 different set of rules were built. In addition, in the three graphics below, for each domain, the results for J48 *Decision Tree* induction algorithm and for the use of the original rules (OR) are showed.

All the results reported in Fig. 1, Fig. 2 and Fig.3 were obtained through a modified 10 *Fold Cross-Validation* strategy. The modified *Cross-Validation* was adopted in order to allow us to perform the *Post-Pruning* approach and it works as follows: we use 80% of the data as training set (used to induce BC), 10 % of the data as the pruning set (used by *Post-Pruning)* and 10% of the data as the test set.

During the results analysis we've verified that the number of instances in a dataset did not have relevant influence in the classification rates neither in the interpretability of the obtained rule sets. Thus, we are reporting only the results obtained when using the 7 datasets (one for each scenario) containing 10000 instances.

The analysis of the 3 graphics can help us to derive some discussions regarding the 3 questions we want to answer (see section 2.1).

Considering the most appropriate BC induction algorithm to BF, the IC algorithm presented better results than K2 and NB. IC provided ACCRs very close the OR ACCRs in 4 out of the 7 datasets (see Fig. 1) and also produced number of rules very close to OR (see Fig. 2). The IC drawback is related to the number of antecedents (see Fig. 3) that tends to be lower than the antecedents in OR. Depending, however, on the main goal of the rule extraction, this low number of antecedents can be seen as a good characteristic because it can help to improve comprehensibility.

Another interesting result we got regarding IC is that in addition to generating similar number of rules, when compared to OR, it was the algorithm which generated the rule sets most similar (based a basic similarity function results) to OR (with or without pruning) . In the *IV* domain, for instance, IC generate exactly the same rules present in OR. The J48 presented the worst results regarding similarity to OR.

Regarding interpretability, the results in the performed experiments revealed that the *Post-Pruning* strategy can be considered a good approach to be applied to BF. The use of the pruning helped reducing the number of rules and antecedents in most of the performed experiments. Obviously, the pruning tended to reduce the ACCRs, this is, however, not a surprise. The main motivation to incorporate a new pruning approach to BF was to help enhancing the comprehensibility of the generated rule set and not to improve the classification accuracy. In this sense, having the *Post-Pruning* as part of BF can allow the user to use the not pruned model to classification purposes and the pruned one to help understanding the problem and having insights about the classification process.

When concerning on the influence of specific data characteristics (e.g. redundant and irrelevant variables and patterns) on the classification accuracy and interpretability, the results did not present any significant tendency to be considered.

In spite of not being part of our initial research questions, the behavior of the J48 Decision Tree learning algorithm deserves some comments. This algorithm presented the closest ACCRs to OR in the *uncertainty* and *unbalanced* domains. In addition, J48 got nice ACCRs on *RP.* The main drawback of this algorithm is the number of rules it tend to generate, thus, mainly when comprehensibility (number of rules) is crucial, using BF should be preferred.

IV. Related Work

The idea of explaining a BN has been explored in some previous works using crisp rules [10][17], fuzzy rules [9] and graphic representations [7]. Our work is focused in describing a Bayesian Classifier in terms of Fuzzy rules mainly because of two critical issues:

- Fuzzy Set Theory can be used to transform quantitative data into qualitative data; in this sense, the discretization method (required by the traditional Bayesian Classifiers) can be performed by a Fuzzyfication Process (FP). The discretization of the data using a FP generates linguistic-valued attributes and, therefore, the model interpretability and comprehensibility may be enhanced.
- The use of linguistic variables and their linguistic values, that are defined by context-dependent fuzzy sets, can enhance the interpretability of the knowledge represented in rule sets [5].

There are other pieces of research in the literature focusing on BN rules explanation as we do. In [9] and [10] a set of Fuzzy rules and a set of Crisp rules are extracted from a Bayesian Classifier respectively. In none of those methods, however, the pruning idea is explored. Another relevant related work is presented in the method by Yap et al. [17] called EBI. In a nutshell, EBI can be described as follows: first, the complexity of a previously given BN is reduced considering only the Markov Blanket of the class node, then an Arc Reversal procedure on the Markov Blanket is applied in order to prepare it for the generation of a Decision Tree (DT) for each possible class value based on Context-Specific Independencies. Afterward, rules are extracted from the DTs. A treatment for missing value and erroneous input data is applied. Different from the BF and PBF (see sSection 3), EBI offers an explanation of a given input and not of the whole BN Classifier. So EBI is classified as predictive explanation as

defined in [3].

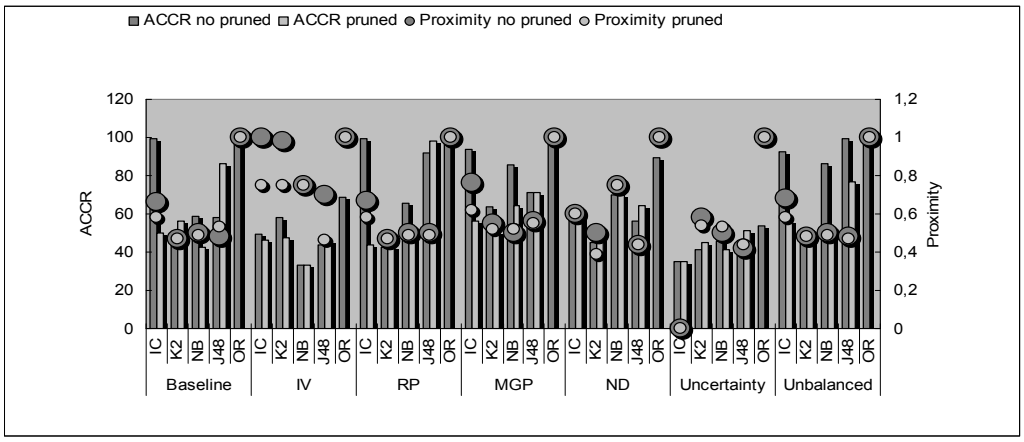

Fig. 1. ACCR and Proximity measures in all 7 domains for the 3 BC induction algorithms, for J48 Decision Tree induction algorithm and for the Original Rules set.

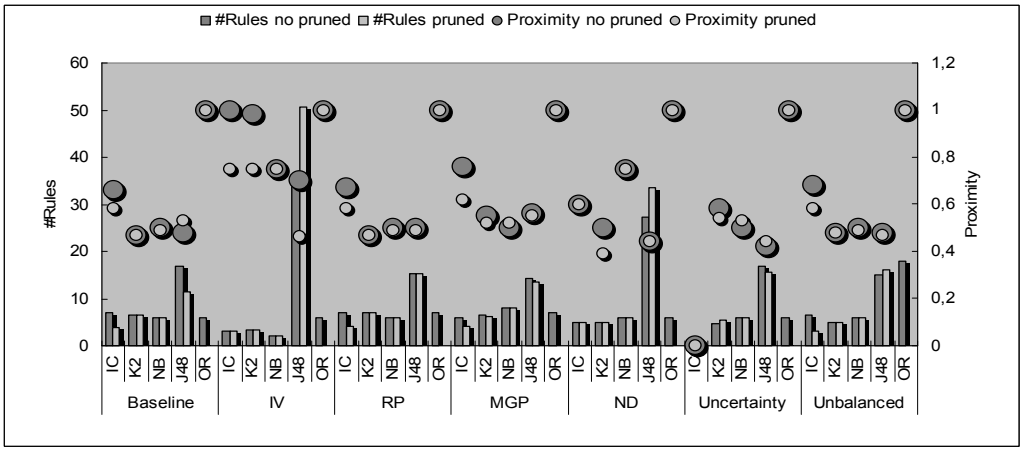

Fig. 2. #Rules and Proximity measures in all 7 domains for the 3 BC induction algorithms, for J48 Decision Tree induction algorithm and for the Original Rules set.

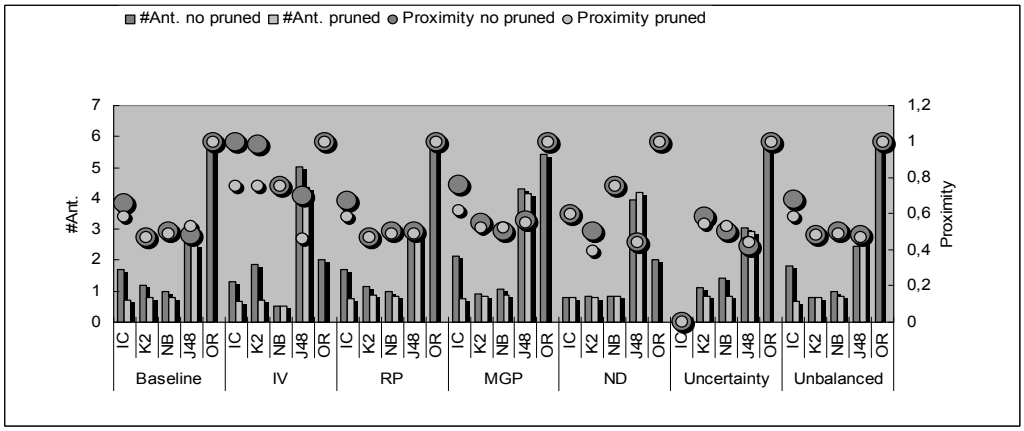

Fig. 3. #Ant. and Proximity measurements in all 7 domains for the 3 BC induction algorithms, for J48 Decision Tree induction algorithm and for the Original Rules set.

Some classic methods deal with the rule extraction problem, like the *C4.5* algorithm [15] which can be used to induce a DT from data. The DT can be easily translated into a set of classification rules. In addition, the *Reduced*

Error Pruning [15] can be used to improve the classification rate and reduce the complexity and amount of the rules. A practical version of the *Reduced Error Pruning*, *Rule Post-Pruning* can be applied as well. As showed in our experiments, however, this methodology can generate a high number of rules.

V. Conclusion and Future Work

In this paper we investigated BayesFuzzy algorithm focusing in three main research questions: i) can a pruning strategy help improving the comprehensibility of the model while maintaining its classification accuracy? ii) What is the influence of specific data characteristics (e.g. redundant and irrelevant variables and patterns) on the classification accuracy and interpretability? iii) Which BC learning algorithm is most appropriate to the specific Bayes/Fuzzy hybrid system BayesFuzzy?

The obtained results allowed us to draw solid conclusions regarding the specific domains (and datasets) used in our experimental analysis, as well as, to draw some very relevant and interesting general interpretations which helped to clearly identify important future work directions.

In a nutshell, for all the experiments using the 7 domains, *Post-Pruning* approach was able to help improving the comprehensibility of the model in a satisfactory way. The influence of specific data characteristics on the classification accuracy and interpretability could not be very well established. And the IC Bayesian Network learning algorithm presented a very nice balance of accuracy and comprehensibility.

Some interesting future work are the further investigation of: i) BayesFuzzy and *Post-Pruning* in bigger and real domains. ii) other conditional independency Bayesian Networks learning algorithms. And iii) automatic fuzzyfication techniques to be used as discretization function to BayesFuzzy.

Acknowledgements

Authors acknowledge the Brazilian company E-BIZ SOLUTION SA Soluções Tecnológicas as well as the Brazilian research agencies CNPq and CAPES for their support.

References

[1] Bressan, G. M. ; Oliveira, V. A. ; Hruschka Jr., E. R. ; Nicoletti, M. C., Using Bayesian networks with rule extraction to infer the risk of weed infestation in a corn-crop. Engineering Applications of Artificial Intelligence, p. 579-592, 2009.

[2] Chajewska U., Draper DL., "Explaining Predictions in Bayesian Networks and Influence Diagrams." In: AAAI Spring Symposium, Stanford Univ., Palo Alto, CA (1998);

[3] Cintra, M E ; Camargo, H. A.; Hruschka JR., ER ; Nicoletti, M. C.. Automatic construction of fuzzy rule bases: a further investigation into two alternative inductive approaches. Journal of Universal Computer Science, 14(15), 2456—2470, 2008.

[4] Cooper, G. F. and Herskovits, E., A Bayesian Method for the Induction of Probabilistic Networks from Data. Mach. Learn. 9(4), 309-347, 1992.

[5] Cordón, O.; Herera, F.; Hoffmann, F. and Magdalena, L. (2001) Genetic Fuzzy Systems – Evolutionary Tuning and Learning of Fuzzy Knowledge Bases. World Scientific, Singapore.

[6] Duda, R. O. & Hart, P. E., Pattern Classification and Scene Analysis. New York, Wiley, 1973.

[7] Heckerman, D.; Chickering, D. M.; Meek, C.; Rounthwaite, R.; Kadie, C. (2000) Dependency networks for inference, collaborative filtering, and data visualization, Journal of Machine Learning Research, vol. 1(1), pp. 49-75.

[8] Hisao Ishibuchi, Tomoharu Nakashima, "*Effect of Rule Weights in Fuzzy Rule-Based Classification Systems*", IEEE Transactions on Fuzzy Systems (Special Issue on FUZZ-IEEE 2000);

[9] Hruschka, E.R.; de Camargo, H.; Cintra, M.E.; do Nicoletti, M., "BayesFuzzy : using a Bayesian Classifier to Induce a Fuzzy Rule Base", Fuzzy Systems Conference, 2007. FUZZ-IEEE 2007. IEEE International, 23-26 July 2007 Page(s):1 – 6;

[10] Hruschka JR., E. R.; Nicoletti, M. C.; Oliveira, V. A.; Bressan GM . BayesRule: a Markov-Blanket based procedure for extracting a set of probabilistic rules from Bayesian classifiers. International Journal of Hybrid Intelligent Systems, v. 5, p. 83-96, 2008.

[11] Jan van den Berg, Uzay Kaymak, Willem-Max van den Bergh, "*Fuzzy Classification Using Probability-Based Rule Weighting*", Fuzzy Systems, IEEE International Conference, vol. 2, pp. 991-996, 2002;

[12] Mitchell, T. M., Machine Learning, McGraw-Hill Higher Education, 1997

[13] Neapolitan, R. E., Learning Bayesian Networks. Prentice-Hall, Inc., 2003.

[14] Pang-Ning Tan, Vipin Kumar, Jaideep Srivastava, "Selecting the Right Interestingness Measure for Association Patterns", Technical Report 2002-112, Army High Performance Computing Reserch Center, 2002;

[15] Quinlan, J. R., C4.5: Programs for Machine Learning. San Mateo, Calif.: Morgan Kaufmann, 1993.

[16] Tan, P., Kumar, V. and Srivastava, J. (2002). Selecting the right interestingness measure for association patterns, 8th ACM SIGKDD int. Conference on Knowledge Discovery and Data Mining, Edmonton, Alberta, Canada, pp. 32 – 41.

[17] Witten, I. H. and Frank, E. 2005 Data Mining: Practical Machine Learning Tools and Techniques, Second Edition (Morgan Kaufmann Series in Data Management Systems). Morgan Kaufmann Publishers Inc.

[18] Yap Ghim-Eng, Tan Ah-Hwee, Pang Hwee-Hwa, *"Explaining inferences in Bayesian Networks*", Applied Intelligence, Volume 29, Issue 3, (December 2008), pages 263-278;