ESANN 2022 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 5-7 October 2022, i6doc.com publ., ISBN 978287587084-1. Available from http://www.i6doc.com/en/.

Bayes Point Rule Set Learning

Fabio Aiolli[1] Luca Bergamin[1] Tommaso Carraro[1,2] Mirko Polato[3]

[1] Department of Mathematics, University of Padova
Padova, Italy

[2] Fondazione Bruno Kessler
Trento, Italy

[3] Department of Computer Science, University of Turin
Turin, Italy

Abstract. This paper proposes an effective bottom-up extension of the popular FIND-S algorithm to learn (monotone) DNF-type rulesets. The algorithm greedily finds a partition of the positive examples. The produced monotone DNF is a set of conjunctive rules, each corresponding to the most specific rule consistent with a part of positive and all negative examples. We also propose two principled extensions of this method, approximating the Bayes Optimal Classifier by aggregating monotone DNF decision rules. Finally, we provide a methodology to improve the explainability of the learned rules while retaining their generalization capabilities. An extensive comparison with state-of-the-art symbolic and statistical methods on several benchmark data sets shows that our proposal provides an excellent balance between explainability and accuracy.

1 Introduction

Thanks to their performance, statistical machine learning methods (e.g., SVM and neural networks) nowadays define the go-to approach in most real-world tasks. However, they are (usually) complex and lack transparency, which leads to poor comprehensibility. Recently, there has been an increasing awareness of the importance of having the ability to explain the decisions of artificial intelligence systems. Within *interpretable machine learning* methods, rule learning represents one of the go-to approaches. The literature offers many algorithms for rule induction [1, 2], but still most of them are far from having performance comparable to the state-of-the-art. Rule sets [3] are by far the most discussed approaches, thanks to their natural interpretation. In binary classification problems, the rules can be combined so that the resulting hypothesis resembles a Boolean formula in Disjunctive Normal Form (DNF): the set contains only rules that describe the positive class (akin to concept learning [4]). Thus, an instance is classified as positive if and only it satisfies the conditions of at least one rule.

Following this direction, we propose FIND-RS (Find Rule Set), an effective bottom-up extension of the popular FIND-S algorithm for DNF-ruleset induction. This method applies a specific-to-general approach by greedily partitioning the set of positive examples while learning for each partition the most specific conjunctive rule consistent with the training set. Conjunctive rules are learned using the same idea of FIND-S [4] and the final hypothesis consists of a DNF. Under mild conditions, FIND-RS guarantees to find a hypothesis that correctly

ESANN 2022 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 5-7 October 2022, i6doc.com publ., ISBN 978287587084-1. Available from http://www.i6doc.com/en/.

classifies the training set. In addition, at the end of the algorithm, an efficient pruning procedure is used to simplify the final DNF and improve interpretability.

As typical in rule learning, on more complex classification tasks the hypothesis of FIND-RS may perform poorly w.r.t. state-of-the-art classification methods. To address this limitation, inspired by the work on Bayes Point Machines [5, 6], we define a principled method to approximate the center of mass of the version space. This method, named FIND-RS-BP, has the advantage to produce an interpretable and very accurate set of rules. We show that this new hypothesis significantly improves the performance. Finally, we provide a heuristic to drastically simplify the rule set generated, thus improving the explainability of FIND-RS-BP while retaining excellent generalization capabilities.

2 Background and notation

We consider binary classification problems with training sets $\mathcal{S} \equiv \{(\mathbf{x}_i, y_i)\}_{i=1}^n$, where $\mathbf{x}_i \in \mathcal{X}$ are categorical feature vectors, with $\mathcal{X} \equiv \times_{j=1}^m \mathcal{X}_j$ for some finite (symbolic) attribute/variable domains $\mathcal{X}_j$, and $y_i \in \{-1, +1\}$. We call $\mathcal{P} \equiv \{\mathbf{x} \mid (\mathbf{x}, y) \in \mathcal{S} \wedge y = +1\}$ the set of positive instances, and conversely, $\mathcal{N} \equiv \{\mathbf{x} \mid (\mathbf{x}, y) \in \mathcal{S} \wedge y = -1\}$ the set of negative instances. We denote with $\mathbf{r} \in \times_{i=1}^m (\mathcal{X}_i \cup \{?\})$ a rule, and we say $\mathbf{r}$ covers an instance $\mathbf{x} \in \mathcal{X}$ (or $\mathbf{x}$ satisfies $\mathbf{r}$), $\mathbf{r} \succeq \mathbf{x}$, iff $\forall i \in [m]$, $r_i = ?$ or $x_i = r_i$. In logical terms, $\mathbf{r} \succeq \mathbf{x}$ means that the conjunction $\bigwedge_{r_i \in \mathbf{r} | r_i \neq ?} (r_i = x_i)$ is true. It is noteworthy that an instance $\mathbf{x}$ can be seen as a rule in which every variable is constrained. A set $D \equiv \{\mathbf{r}_1, \ldots, \mathbf{r}_k\}$ of (conjunctive) rules covers an instance $\mathbf{x} \in \mathcal{X}$, denoted $D \succeq \mathbf{x}$, iff $\exists \mathbf{r} \in D \mid \mathbf{r} \succeq \mathbf{x}$. From a logical stand point, a set of conjunctive rules corresponds to a monotone DNF (MDNF). We say that a DNF formula is consistent with a training set iff it covers the positive training instances and it does not cover any negative training instance. We say that a rule set (or DNF) D_1 generalizes a rule set D_2, $D_1 \geq D_2$, iff $\forall \mathbf{x} \in \mathcal{X}$, $(D_2 \succeq \mathbf{x}) \Rightarrow (D_1 \succeq \mathbf{x})$. With a slight abuse of notation, an MDNF rule h is often used as a classification function in such a way that given an example $\mathbf{x}$, $h(\mathbf{x}) = +1$ if $h \succeq \mathbf{x}$, -1 otherwise. An ordered set of rules is denoted by $\langle \mathbf{r}_1, \ldots, \mathbf{r}_k \rangle$. Finally, $[\![b]\!] \in \{0, 1\}$ denotes the indicator function which is 1 iff the condition b is true.

3 FIND-RS

FIND-RS considers a hypothesis space consisting of MDNF formulas of arbitrary size. To find the solution that better explains the dataset, it uses a bottom-up approach, starting from a very specific hypothesis that is greedily generalized to allow new positive instances to be covered, while being consistent with the negative instances.

FIND-RS computes the classification MDNF hypothesis by building a conjunctive rule at a time. Each conjunction is initially defined as the rule corresponding to a randomly picked positive training instance $\mathbf{s} \in \mathcal{P}$ (not already covered by the running MDNF). Then, FIND-RS tries to greedily generalize

ESANN 2022 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 5-7 October 2022, i6doc.com publ., ISBN 978287587084-1. Available from http://www.i6doc.com/en/.

such conjunctive rule considering the remaining uncovered positive instances in a fashion similar to the FIND-S algorithm. Akin FIND-S, FIND-RS generalizes a rule by removing one or more attribute-value constraints (i.e., setting it equals to ?) while keeping the overall hypothesis consistent with the negative examples.

It is worth to notice that, by design, FIND-RS will always find a (MDNF) hypothesis that correctly classifies all the training set iff there are no contradictory examples, that is $\nexists(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2) \in \mathcal{S} \mid (\mathbf{x}_1 = \mathbf{x}_2) \wedge (y_1 \neq y_2)$. In the worst case, FIND-RS will return a hypothesis of the form $\bigvee_{\mathbf{p} \in \mathcal{P}} \mathbf{p}$ that clearly overfits the training set. Algorithm 1 provides a detailed description of FIND-RS.

Rule pruning. Being FIND-RS a greedy algorithm, at the end of the training process, the produced hypothesis may contain superfluous rules, i.e., rules that can be removed without losing the consistency with the training set. Thus, the idea of the *rule pruning* is to discard such redundant rules. To speed up this pruning process, we rely on the observation that at every iteration t holds that $\forall j < i \in [k],\ \nexists \mathbf{x} \in B_i \mid \mathbf{r}_j \succeq \mathbf{x}$, where the current hypothesis is $D^t = \langle \mathbf{r}_1, \ldots, \mathbf{r}_k \rangle$. This provides a "backward incompatibility" between rules, however, it does not say anything in the other direction. In particular, it may happen that every instance in B_i, for some i, can be covered by other conjunctive rules $\mathbf{r}_j$ for $j > i$. In practice, this post-processing step (`prune` in Algorithm 1) checks if some B_i can be emptied. In such a case, the corresponding conjunctive term can be safely removed from the current hypothesis, thus creating a more specialized one that is still consistent with the training set.

Algorithm 1: FIND-RS

Input: $\mathcal{P}$: set of positive examples; $\mathcal{N}$: set of negative examples
Output: Disjunctive Normal Form rule

$B, D, k \leftarrow [\,], [\,], 0$ ▷ Create an empty bucket list and an empty rule list
while $\mathcal{P}$ *is not empty* **do**
  $\mathbf{s} \leftarrow \text{pop}(\mathcal{P})$ ▷ Pick a starting example
  $B, D \leftarrow B + \{\mathbf{s}\}, D + \mathbf{s}$ ▷ Create a new bucket/rule
  $Q \leftarrow [\,]$ ▷ Track examples not covered by a chosen rule
  for $\mathbf{p} \in \mathcal{P}$ **do**
    $\mathbf{r} \leftarrow D_k$ ▷ Get the latest rule found
    $\mathbf{r}' \leftarrow \text{generalize}(\mathbf{r}, \mathbf{p})$ ▷ Attempt to further generalize it
    if $\nexists \mathbf{n} \in \mathcal{N} \mid \mathbf{r}' \succeq \mathbf{n}$ **then**
      $B_k \leftarrow B_k \cup \{\mathbf{p}\}$ ▷ Update the last bucket
      $D_k \leftarrow \mathbf{r}'$ ▷ Update the last rule
    else
      $Q \leftarrow Q + \mathbf{p}$ ▷ Update examples not covered by current rule
  $\mathcal{P} \leftarrow Q$ ▷ Keep finding new rules for examples not covered
  $k \leftarrow k + 1$ ▷ Index the next bucket
return $\text{prune}(D, B)$

ESANN 2022 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 5-7 October 2022, i6doc.com publ., ISBN 978287587084-1. Available from http://www.i6doc.com/en/.

3.1 Bayes optimal approximation

Under the condition that there are no contradictory instances, FIND-RS guarantees to find one hypothesis of the version space. Moreover, assuming there exists a target MDNF that has generated data, such hypothesis lies in the version space. In large version spaces such hypothesis is hard to find and with a uniform prior over the hypothesis it is known that the optimal choice is to return the expected value of the decision of the hypotheses in the version space (aka Bayes Optimal Classifier). A surrogate of the optimal classifier is the so-called Bayes Point Classifier (BPC) in which the center of mass of the version space is selected as classification hypothesis. Unfortunately, to get such hypotheses, one needs to sample uniformly from the version space and this is a very difficult task. With the aim to approximate the BPC [5], we propose to combine different MDNFs obtained from multiple runs of FIND-RS.

Consider an R-dimensional vector space where R is the number of possible rules. Then, a ruleset can be seen as a vector $\hat{\mathbf{w}} \in \mathbb{R}^{R+1}$ where $\hat{w}_r = 1, r \leq R$ iff the rule indexed by r is present in the ruleset, and $\hat{w}_{R+1} = -\frac{1}{2}$. An instance is represented in the same space as the vector $\hat{\mathbf{x}} \in \mathbb{R}^{R+1}$ where $\hat{x}_r = [\![\mathbf{r} \succeq \mathbf{x}]\!], r \leq R$, and $\hat{x}_{R+1} = 1$. It can be easily verified that the ruleset decision can be given as $h(\mathbf{x}) = \text{sign}(\langle \hat{\mathbf{w}}, \hat{\mathbf{x}} \rangle)$. FIND-RS-BP is obtained by taking the approximate Bayes point as the average of the hypotheses $\hat{\mathbf{w}}^{(t)}$ found by running FIND-RS T times:

$$H_{bp}(\mathbf{x}) = \text{sign}\left(\frac{1}{T}\sum_{t=1}^{T}(\langle \hat{\mathbf{w}}^{(t)}_{1:R}, \hat{\mathbf{x}}_{1:R} \rangle - \frac{1}{2})\right) > 0 \Leftrightarrow \sum_{t=1}^{T}\langle \hat{\mathbf{w}}^{(t)}_{1:R}, \hat{\mathbf{x}}_{1:R} \rangle > \sum_{t=1}^{T}\frac{1}{2} = \frac{T}{2}.$$

Noticing that $\sum_t \langle \hat{\mathbf{w}}^{(t)}_{1:R}, \hat{\mathbf{x}}_{1:R} \rangle$ equals the number of rules in the T rulesets that $\mathbf{x}$ satisfies, then H_{bp} classifies a new instance as positive whenever the number of rules $\mathbf{x}$ satisfies exceeds $T/2$. Moreover, let G be the set of unique discovered rules, then the decision can be compacted as $\sum_{\mathbf{r}\in G} \alpha_{\mathbf{r}} [\![\mathbf{r} \succeq \mathbf{x}]\!] > T/2$ where $\alpha_{\mathbf{r}}$ is the number of times $\mathbf{r}$ has been discovered.

This approach gives a principled method to order the discovered rules according to their weights α. This weight represents the importance of a given rule in the decision thus providing an indicator that could be used to perform rule pruning. Note that, when pruning (*cut-off pruning*) is performed, let say by retaining the first K rules G_K, then the discriminant function needs to be changed accordingly. Namely, $H_K(\mathbf{x}) = +1$ iif

$$\sum_{\mathbf{r}\in G_K} \alpha_{\mathbf{r}} [\![\mathbf{r} \succeq \mathbf{x}]\!] > \frac{\gamma_K T}{2}, \text{ where } \gamma_K = \frac{\sum_{\mathbf{r}\in G_K} \alpha_{\mathbf{r}}}{\sum_{\mathbf{r}\in G} \alpha_{\mathbf{r}}}.$$

4 Experiments

This section presents the experiments performed with FIND-RS. They have been executed on an Apple MacBook Pro (2019) with a 2,6 GHz 6-Core Intel Core i7. The model has been implemented in Python using `scikit-learn`. Our source

ESANN 2022 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 5-7 October 2022, i6doc.com publ., ISBN 978287587084-1. Available from http://www.i6doc.com/en/.

code is publicly available[1]. We selected 9 discrete real-world datasets from the UCI Machine Learning Repository. Multi-class datasets have been converted into binary classification datasets by selecting the most frequent class as the positive one. We applied this procedure [7] for all the datasets expcept for `monk-1`, `monk-2`, `monk-3`, where we used the value `1` as the positive class. We compared FIND-RS with different baselines (i.e., `RIPPER`, `CART`, Logistic Regression, Random Forests, and `SVM`). `RIPPER` [1] is a state-of-the-art rule learning model that scales nearly linearly with the number of training examples and can efficiently deal with noisy datasets. `CART` [8] is a widespread implementation of decision trees that can have good performance and high interpretability.

For the experiments we used the following procedure: (1) the dataset is randomly split into training and test set, using a 50/50 proportion; (2) a grid search is performed on the training set using 5-fold cross-validation, with hyper-parameters reported in [9]; (3) the best model found is trained on the entire training set; (4) the performance is computed on the test set. The number of iterations T of FIND-RS-BP has been set to 100 (FIND-RS-BP$_{100}$). We repeated this process for ten runs and the `F1-score` has been averaged among these runs. Additionally, we tried both the attribute-value (AV) and the one-hot (OH) encoding representations for `CART`, `RIPPER`, and FIND-RS and we reported the results obtained with the best performing one. For the other baselines, we used the OH encoding representation only, as AV is not compatible.

Results Table 1 summarizes the results. In our experiments, SVM outperformed Random Forests. Hence, for space constraints, we reported only the results for SVM.

	FIND-RS	FIND-RS-BP$_{100}$	RIPPER	CART	LR	SVM
`car`	$0.988_{(0.00)}$	$\mathbf{0.990}_{(0.00)}$	$0.988_{(0.01)}$	$0.983_{(0.00)}$	$0.963_{(0.00)}$	$0.996_{(0.00)}$
`kr-vs-kp`	$0.988_{(0.00)}$	$\mathbf{0.991}_{(0.00)}$	$0.981_{(0.00)}$	$0.989_{(0.01)}$	$0.972_{(0.00)}$	$0.991_{(0.00)}$
`monk-1`	$\mathbf{1.000}_{(0.00)}$	$\mathbf{1.000}_{(0.00)}$	$0.939_{(0.06)}$	$0.909_{(0.05)}$	$0.669_{(0.02)}$	$1.000_{(0.00)}$
`monk-2`	$0.864_{(0.06)}$	$\mathbf{0.894}_{(0.04)}$	$0.175_{(0.12)}$	$0.811_{(0.08)}$	$0.038_{(0.05)}$	$0.966_{(0.02)}$
`monk-3`	$0.962_{(0.01)}$	$0.978_{(0.01)}$	$0.935_{(0.01)}$	$\mathbf{0.986}_{(0.01)}$	$0.972_{(0.01)}$	$0.985_{(0.01)}$
`mushrooms`	$\mathbf{1.000}_{(0.00)}$	$\mathbf{1.000}_{(0.00)}$	$\mathbf{1.000}_{(0.00)}$	$\mathbf{1.000}_{(0.00)}$	$\mathbf{1.000}_{(0.00)}$	$1.000_{(0.00)}$
`ttt`	$\mathbf{1.000}_{(0.00)}$	$\mathbf{1.000}_{(0.00)}$	$0.986_{(0.02)}$	$0.932_{(0.02)}$	$0.986_{(0.00)}$	$0.986_{(0.00)}$
`vote`	$0.888_{(0.03)}$	$0.908_{(0.02)}$	$0.815_{(0.03)}$	$0.933_{(0.02)}$	$\mathbf{0.942}_{(0.01)}$	$0.939_{(0.02)}$
`connect-4`*	0.851	**0.896**	0.730	0.849	0.849	0.917
AvgRank	2.61	**1.67**	3.89	3.06	3.78	-

Table 1: Test `F1-score` averaged across ten runs. (*) For `connect-4` the hyper-parameter T of FIND-RS-BP is set to 20, and the evaluation procedure has been computed once. SVM is shown just for reference since it is not interpretable.

We can observe that FIND-RS outperforms most of the baselines, even on the most challenging dataset, namely `connect-4`. As expected, FIND-RS-BP$_{100}$ improves upon FIND-RS reaching the best average rank of 1.83.

The effect of cut-off pruning. Figure 1 shows how the training and test accuracies vary w.r.t. the degree of cut-off pruning performed in 3 different datasets.

[1] https://tinyurl.com/bdcwh35d

ESANN 2022 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 5-7 October 2022, i6doc.com publ., ISBN 978287587084-1. Available from http://www.i6doc.com/en/.

We can set a threshold and select the minimum number of rules such that the training accuracy is above that threshold. We show that setting a threshold of $0.99 \times$ tr_acc, where tr_acc is the training accuracy of the complete rule set without pruning, can drastically reduce the number of rules, thus improving the interpretability of the ruleset with no reduction in performance.

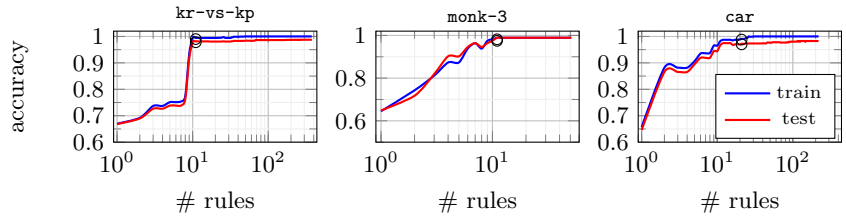

Fig. 1: The plots (log scale) show the accuracy varying the number of kept rules in the FIND-RS-BP pruning phase. Rules are sorted by importance. The black circle highlights the accuracy corresponding to the 99% threshold.

5 Conclusions

We proposed a novel methodology that tries to approximate the Bayes optimal classifier in the hypothesis space of MDNF. The method has demonstrated to discover rules that are very accurate and still interpretable. We have also provided a methodology to reduce the number of rules of the final hypothesis, thus improving interpretability with no performance loss. We are currently working on a principled extension to multi-class classification and continuous variables.

References

[1] William W Cohen. Fast effective rule induction. In *Machine learning proceedings 1995*, pages 115–123. Elsevier, 1995.

[2] P. Clark and T. Niblett. The cn2 induction algorithm. *Machine learning*, 3(4):261–283, 1989.

[3] Johannes Fürnkranz. *Rule Learning*, pages 875–879. Springer US, Boston, MA, 2010.

[4] Thomas M. Mitchell. *Machine Learning*. McGraw-Hill, Inc., USA, 1 edition, 1997.

[5] Ralf Herbrich and Thore Graepel. Large scale bayes point machines. In *Advances in Neural Information Processing Systems*, volume 13, pages 528–534. MIT Press, 2000.

[6] Ralf Herbrich, Thore Graepel, and Colin Campbell. Bayes point machines. *J. Mach. Learn. Res.*, 1:245–279, 2001.

[7] Florian Beck and Johannes Fürnkranz. An empirical investigation into deep and shallow rule learning. *arXiv preprint arXiv:2106.10254*, 2021.

[8] Leo Breiman, Jerome H Friedman, Richard A Olshen, and Charles J Stone. *Classification and regression trees*. Routledge, 2017.

[9] Zhuo Wang, Wei Zhang, Ning Liu, and Jianyong Wang. Scalable rule-based representation learning for interpretable classification. *Advances in Neural Information Processing Systems*, 34, 2021.