Fusion Engineering and Design 130 (2018) xxx–xxx

Contents lists available at ScienceDirect

Fusion Engineering and Design

journal homepage: www.elsevier.com/locate/fusengdes

On the potential of ruled-based machine learning for disruption prediction on JET☆

M. Lungaroni[a], A. Murari[b], E. Peluso[a,*], J. Vega[c], G. Farias[d], M. Gelfusa[a], JET Contributors[1]

[a] Department of Industrial Engineering, University of Rome "Tor Vergata", via del Politecnico 1, Roma, Italy
[b] Consorzio RFX (CNR, ENEA, INFN, Universita' di Padova, Acciaierie Venete SpA), Corso Stati Uniti 4, 35127 Padova, Italy
[c] Laboratorio Nacional de Fusión, CIEMAT, Av. Complutense 40, 28040 Madrid, Spain, Spain
[d] Pontificia Universidad Catolica de Valparaiso, Av. Brasil 2147, Valparaiso, Chile

ARTICLE INFO

Keywords:
Disruptions
Machine learning predictors
Classification and regression trees
Boosting
Bagging
Random forests
Noise-based ensembles

ABSTRACT

In the last years, it has become apparent that detecting disruptions with sufficient anticipation time is an essential but not exclusive task of predictors. It is also important that the prediction is accompanied by appropriate qualifications of its reliability and it is formulated in mathematical terms appropriate for the task at hand (mitigation, avoidance, classification etc.). In this paper, a wide series of rule-based predictors, of the Classification and Regression Trees (CART) family, have been compared to assess their relative merits. An original refinement of the training, called noise-based ensembles, has allowed not only to obtain significantly better performance but also to increase the interpretability of the results. The final predictors can indeed be represented by a tree or a series of specific and clear rules. Such performance has been proved by analysing large databases of shots on JET with both the carbon wall and the ITER Like Wall. In terms of performance, the developed tools are therefore very competitive with other machine learning techniques, with the specificity of formulating the final models in terms of trees and simple rules.

1. Rule-based machine learning for disruption prediction in tokamaks

Since they can compromise the integrity of large tokamaks, particularly in the parameter range of the next generation of devices, disruptions have been intensively studied in the last decades [1,2]. These invesigations range from mitigation techniques, such as massive gas injection, to prediction and avoidance strategies. Of course, reliable forecasting tools are an essential ingredient in the implementation of any mitigation or avoidance intervention. Unfortunately, the theoretical understanding of disruption causes is not sufficient to programme reliable simulation models for forecasting. Consequently, in the last decades, many efforts have been devoted to deriving empirical models from experiments, to identify the boundary between the safe and disruptive regions of the operational space. Among these empirical models, the most performing are based on machine learning tools. On JET two generations of machine learning predictors, APODIS and SPAD [3–7], have been implemented in the real time network. These classifiers, and the others tested offline, are based on various machine learning techniques, ranging from the distance based ones (SVM and Neural Networks), to clustering and fuzzy logic [8–10]. A family of techniques not significantly explored are the rule based ones, which are the subject of this paper.

In the field of computer science, the term rule-based machine learning (RBML) indicates the machine learning methods that extract "rules" to solve a problem directly from the data available. These rules are typically in the form of *if…then* clauses and an example in the case of disruption prediction could be: *if* the locked mode amplitude is higher than threshold_1 *and* the internal inductance is lower than threshold_2 *then* the discharge is going to disrupt. Therefore, the defining aspect of rule-based machine learners, in their application to data mining, is their capability to identify a set of relational rules that best represent the knowledge in the data, relevant to solving the problem at hand. This is in contrast to traditional rule-based systems, which are hand-crafted and therefore simply encode already available, prior human knowledge. Expressing the data driven knowledge as rules is a significant advantage for both the interpretation and the implementation of the results, as will become clear in the next sections. The methods implemented and refined to perform the studies described in the rest of the paper are based on the Classification And Regression Tree

☆ EUROfusion Consortium, JET, Culham Science Centre, Abingdon, OX14 3DB, UK.
* Corresponding author.
E-mail address: emmanuele.peluso@uniroma2.it (E. Peluso).
[1] See the author list of "X. Litaudon et al. 2017, Nucl. Fusion, 57, 102001".

https://doi.org/10.1016/j.fusengdes.2018.02.087
Received 14 October 2017; Received in revised form 11 January 2018; Accepted 22 February 2018

(CART) technology. This technique allows producing a tree summarising the rules as the final output.

Rule-based classifiers of the CART family are very powerful and easy to interpret. On the other hand, one of their main problems is the sensitivity to the details of the training set. Their final trees are indeed not very stable; small changes in the training set can result in major differences in the final trees. To alleviate this problem, the approach of ensemble rule-based classifiers has proved to be very successful. It consists of training many even not very performing classifiers and then somehow average their results in order to obtain the final classification of the new examples. An original refinement of the weak learners training, called noise-based ensembles, has allowed not only to obtain both better performance and increased the interpretability of the results.

In order to follow the evolution of the operational space during the campaigns, an adaptive form of training has been adopted. Such a training has also the advantage of optimising the computational efforts by minimising the training set. This procedure implements a "*learning from scratch*" approach so that all the proposed predictors can start working with just one disruptive and one non disruptive example [11,12]. The last model is updated as the campaign progresses, by refining the training with additional cases.

To test the aforementioned technologies and training strategies, we have use the amplitude of the locked mode and the internal inductance as input signals. As discussed later, these are the two most relevant macroscopic quantities for the campaigns investigated. Regarding the structure of the paper, next section gives an overview of the rule-based classifiers of the CART family. Section 3 introduces the methodology of the ensemble rule-based classifiers, Section 4 discusses in detail the adaptive method adopted to train the various versions of the predictors and describes the main characteristics of JET database investigated. The results obtained for the ILW and a Carbon wall are reviewed in Sections 5 and 6. The conclusions and lines of future work are the subject of the last Section 7 of the paper.

2. The basics of classification tree analysis

Nowadays the reference, basic rule-based machine learning tools are the so called Classification and Regression Trees (CART). They have been widely implemented for constructing prediction models from data [13]. Such models are derived directly from the available databases by recursively partitioning the data space and fitting a simple prediction rule at each partition. The final partitioning, once properly optimised, consists therefore of a series of rules that can be represented graphically by a decision tree. Classification trees, the subject of this paper, have been conceived to classify response variables that take a finite number of unordered values. Their performance are therefore typically quantified in terms of misclassification costs. Regression trees are an extension used to handle response variables that take continuous or ordered discrete values, with prediction error typically measured by the squared difference between the observed and predicted values.

Decision trees are supervised techniques and therefore require the a priori definition of the number of classes and a sufficient number of examples. In the applications described in this paper, decision trees are used to solve classification problems, which mathematically can be formalised as follows. Given a training sample of n observations, the class variable is indicated by Y and can in general take a finite set of discrete values 1, 2, …, k. In our application, the number of classes is typically 2. The set of p features used as predictor variables are indicated by $X_1,…, X_p$. The objective of the analysis consists of finding a model, which can predict the class Y from new X values. The method to identify the best model consists of partitioning the database one node at the time starting from the root. The algorithm exhaustively searches the whole database to determine which variable and which value maximise the total purity of its two child nodes. A cartoon describing visually this process for a simple case is provided in Fig. 1. The goal consists of identifying a simple set of rules capable of discriminating between the squares and the circles. Feature B provides the best first rule to separate the two symbols. The corresponding rules can be written in the form: "*if* B is higher than B_1 *then* the class is crosses". Unfortunately, with this simple rule the classification is not perfect since the purity of the two leaf nodes defined by the rule is not perfect. Intuitively the purity of a node can defined as the percentage of properly classified examples by a leaf node. It is called purity because ideally each leaf node should contain only elements of a single target class. To improve the purity of the leaf nodes in the simple example of Fig. 1, it is possible to define two thresholds, A_1 and A_2, in the second feature A. Now it is possible to combine the thresholds in the two features to obtain rules, which separate perfectly the two classes. For example, to classify properly the circles on the top left hand part of the plot, the following rule can be adopted: " *if* B is less than B_1 and *if* A is higher than A2 *then* the class is circles".

To quantify the purity of a node, the version of CART implemented for the studies of this paper uses a generalization of the binomial variance called the Gini index [13]. As a metric to split the nodes, the Gini impurity calculates how often a randomly chosen element from the training set would be incorrectly labelled, under the assumption that the labels are allocated as the distribution of labels in the subset. The Gini factor is typically computed by summing the probability p_i of the item being correctly classified by the probability (1-p_i) of the item being wrongly classified

$$\mathrm{GINI} = \Sigma\, p_i(1\text{-}p_i) \tag{1}$$

where the sum is extended over the number of classes. The GINI impurity reaches its minimum (zero) when all cases in the node fall into a single target category.

3. Ensemble rule-based classifiers

Ensemble rule-based classifiers implement the concept of weak learners. A 'weak' learner (either classifier or predictor) is just a machine learning tool, which produces a model that performs relatively poorly but is often, but not always, computationally simple. The relatively limited computational resources required allow training various versions of such weak learners which can then be pooled (via Bagging, Random Forests etc) together to create a "strong" ensemble classifier. The basic elements of the ensembles used in this paper are decision trees of the type described in the previous section. The next subsections provide some details about the various weak learners trained and pooled to obtain the results reported in the rest of the paper. These techniques are nowadays quite standard; the original methodological development introduced in this treatment is the category of so called *noise-based ensemble classifiers*, which take into account the effects of the noise on the measurements.

3.1. Bagging

One of the main weaknesses of decision trees is the sensitivity of their results to the specific data used for their training. A small change in the inputs (for example even using a subset of the training data) can imply a major variation in the resulting decision tree and in turn quite different predictions. Bagging is an application of ensemble weak learners to reduce this high-variance of decision trees. Bagging of the CART algorithm would consist of the following steps:

1. Generation of many random sub-samples of the original dataset with replacement.
2. Training of a CART model for each subset of samples.
3. Given a new example, calculate the average prediction from each model and select the class with a form of majority vote.

When Bagging, individual tree overfitting the training data is less of a concern. For this reason, the individual decision trees can be grown