A SUPPORT VECTOR MACHINE APPROACH TO CREATE OBLIQUE DECISION TREES FOR REGRESSION

Andrea Carta [1]

[1] Department of Business and Economics, University of Cagliari, Cagliari, Italy, (e-mail: andrea.carta88@unica.it)

ABSTRACT: Decision trees are a popular statistical learning algorithm for classification and regression that recursively split the data based on the most informative characteristics. Unfortunately, they do not have a high predictive power with respect to other statistical learning methods. To enhances their performances, this paper proposes a support vector machine approach to create oblique decision trees for regression problems. In this novel model, the split at each node is made through a weighted support vector machine classifier with a linear Kernel that minimizes the deviance of the split. We test the model with respect to the usual CART on four public datasets with numerical predictors on three global metrics: Root Mean Squared Error, Mean Absolute Deviation, and R^2. The results of repeated cross-validation show that the novel model can overperform the usual Decision trees.

KEYWORDS: Trees, Oblique Split, SVM, Regression, Oblique Trees

1 Introduction

Decision trees (DTs) are a popular statistical learning algorithm for classification and regression. They can be easily viewed and interpreted by humans, making them valuable assets in data. A DT is a tree structure in which each internal node represents a decision based on a specific characteristic of the data, and where each leaf node represents a prediction or result. The algorithm works by recursively splitting the data based on the most informative characteristics until a stopping criterion is met. Unfortunately, DTs are prone to overfitting and do not have a high predictive power with respect to other statistical learning methods. To improve their performances oblique DTs were introduced (Breiman, 2017), and lately, they are gaining interest in the research community. Unlike traditional DTs, in which each node corresponds to a single variable split and the separation between the branches is orthogonal to the axes, oblique DTs allow the definition of separation hyper-planes that can be inclined with respect to the Cartesian axes. In other words, oblique

DTs use linear combinations of multiple variables to define decision boundaries. However, to find the linear combination of variables to construct the best-suited hyperplane is an NP-hard problem, in fact, to split a node with n observations using an axis-aligned CART, an exhaustive search would require no more than $n \cdot p$ evaluations. On the other hand, oblique CART would require a significantly larger number of evaluations, specifically $2^p \binom{n}{p}$. Nevertheless, oblique DTs have the advantage of generally building smaller trees with better accuracy compared with axis parallel trees (Wickramarachchi *et al.*, 2016). In contrast to the Breiman's approach, we introduce Support Vector Machine Regression Oblique Tree (SVM-ROT). In the Breiman method, the algorithm optimizes the coefficients of oblique splits based on a coordinate descent method. This is an iterative approach where each coefficient is optimized individually while keeping the others fixed. On the other hand, in SVM-ROT the split at each node is determined through a weighted support vector machine (SVM) classifier with a linear Kernel that minimizes the deviance of the split. SVM is a supervised statistical learning method introduced by Vapnik, 1999 to solve pattern classification and regression problems, moreover, it can be linear or nonlinear but is most commonly the former. Essentially, SVM identifies a reproducible hyperplane that maximizes the margin between the support vectors of both class labels. To improve the performance of the SVM classifiers, Yang *et al.*, 2007 suggests adding different weights to observations to different data points such that the weighted SVM algorithm estimates the best hyperplane according to the relative importance of the observation in the training data set. This short paper is organized as follows. Section 2 introduces the model in detail, in Section 3 the model is tested on 4 datasets and some concluding comments are reported.

2 Model

SVM-ROT at each node separates the observations given the results of a SVM classifier. Let us consider N observations characterized by a continuous response Y and p continuous features. First, Y is transformed K times into a dichotomous variable, each time using a different quantile as the threshold for its partitioning. Then, for each of these dichotomized variables, a weighted SVM classifier with linear kernel is applied, and the algorithm saves the deviance reduction resulting from the two partitions. The algorithm then chooses the split that has the highest reduction in deviance. The weighting of the SVM is very important because when the algorithm dichotomizes the target variable much information is lost. To overcome this problem the absolute values of

the scaled elements of the target variable Y are used as weights in the classifiers. This process assures that the hyperplane takes into account the values of the original Y. The result of this process will be a set of coefficients $\mathbf{w}$ of length p, and an intercept b, which describe the separating hyperplane. The hyperplane will be then expressed in a decision rule similar to that one of the usual DT, creating the pair of half-spaces: $R_1(w,b) = \{\mathbf{X} \mid \mathbf{w} \cdot \mathbf{x} + b \leq 0\}$ and $R_2(w,b) = \{\mathbf{X} \mid \mathbf{w} \cdot \mathbf{x} + b > 0\}$, where $\mathbf{X}$ is the matrix of the p predictors.

The result will be the division of the feature space into two subsets. This operation is then applied in a recursive binary partition manner until a certain criterion is met. These stopping criteria can be the number of elements in a leaf, the number of elements in a node, or the complexity parameter given by the ratio between the resulting deviance after the split and the deviance in the parent node.

3 Application to real datasets

SVM-ROT has been applied to several real datasets using the software `R` (R Core Team, 2022). The first is "Body Fat" dataset from Penrose *et al.*, 1985. In this dataset, the response variable is the percentage of body fat and the eleven predictors represent several physiologic measurements related to 252 men. The second dataset, called BCF, comes from Grisoni *et al.*, 2016, here the target variable is the Bioconcentration Factor in log units of 779 chemicals, while the independent variables are nine molecular numerical descriptors. The third data set is Auto MPG dataset from Dua & Graff, 2017 consisting of 398 observations, but in which only the seven numerical predictors have been used. Finally, the last dataset is from Ancell, 2021, it is made up of 413 instances and contains the 50 year ground snow load at a variety of measurement stations together with four numerical predictors. The performance of the SVM-ROT is compared to the one of a CART. Both models were tuned for the complexity parameter with 10-fold cross-validation, and the most parsimonious model with the one standard error rule was chosen. Then we performed 10 times repeated 10-fold cross-validation. The overall performance is computed by Root Mean Squared Error (RMSE), Mean Absolute Deviation (MAD), and R^2. For the first two metrics, lower values result in better predictive models. However, RMSE is more sensitive to high errors. R^2 is the proportion of variance explained by the model, this means that a value close to one indicates that the model explains most of the variance. Table 1 shows the results of the experiments.

In BCF and MPG SVM-ROT shows a better performance with respect to

	Body Fat		BCF		MPG		Snow	
	SVM-ROT	CART	SVM-ROT	CART	SVM-ROT	CART	SVM-ROT	CART
RMSE	**5.385**$_{(0.151)}$	5.396$_{(0.198)}$	**0.776**$_{(0.008)}$	0.795$_{(0.008)}$	**3.245**$_{(0.071)}$	3.367$_{(0.073)}$	1.506$_{(0.0521)}$	**1.445**$_{(0.058)}$
MAD	**4.422**$_{(0.109)}$	4.430$_{(0.170)}$	**0.597**$_{(0.008)}$	0.613$_{(0.005)}$	**2.404**$_{(0.061)}$	2.460$_{(0.068)}$	**0.898**$_{(0.027)}$	0.940$_{(0.027)}$
R^2	**0.604**$_{(0.022)}$	0.602$_{(0.031)}$	**0.674**$_{(0.008)}$	0.656$_{(0.005)}$	**0.833**$_{(0.008)}$	0.819$_{(0.008)}$	0.861$_{(0.007)}$	**0.871**$_{(0.011)}$

Table 1. *Results of SVM-ROT and CART for all four dataset. The means (standard errors) of the* 10*-times* 10*-fold cross-validation of the three metrics are reported. In bold the best model for each metric and dataset.*

CART for all three global metrics. Instead, in "Snow" the improvement is only for MAD, whilst for "Body Fat" the results are almost identical. Nevertheless, as at each node, the SVM-ROT splits the predictor space using all the covariates at once, so SVM-ROT is prone to overfit the data. In the future, it will be then interesting to use this novel model with an ensemble learning approach such as random forests or gradient boosting, or to apply a kind of feature selection at each split.

References

ANCELL, ETHAN. 2021. *autocart: Autocorrelation Regression Trees*. R package version 1.4.5.

BREIMAN, LEO. 2017. *Classification and regression trees*. Routledge.

DUA, DHEERU, & GRAFF, CASEY. 2017. *UCI Machine Learning Repository*.

GRISONI, FRANCESCA, CONSONNI, VIVIANA, VIGHI, MARCO, VILLA, SARA, & TODESCHINI, ROBERTO. 2016. Investigating the mechanisms of bioconcentration through QSAR classification trees. *Environment international*, **88**, 198–205.

PENROSE, KEITH W, NELSON, AG, & FISHER, AG. 1985. Generalized body composition prediction equation for men using simple measurement techniques. *Medicine & Science in Sports & Exercise*, **17**(2), 189.

R CORE TEAM. 2022. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

VAPNIK, VLADIMIR. 1999. *The nature of statistical learning theory*. Springer science & business media.

WICKRAMARACHCHI, D.C., ROBERTSON, B.L., REALE, M., PRICE, C.J., & BROWN, J. 2016. HHCART: An oblique decision tree. *Computational Statistics Data Analysis*, **96**, 12–23.

YANG, XULEI, SONG, QING, & WANG, YUE. 2007. A weighted support vector machine for data classification. *International Journal of Pattern Recognition and Artificial Intelligence*, **21**(05), 961–976.