# Inducing Non-Orthogonal and Non-Linear Decision Boundaries in Decision Trees via Interactive Basis Functions

Antonio Paez*[,a], Fernando Lopez[b], Manuel Ruiz[b], Maximo Camacho[c]

[a] *School of Geography and Earth Sciences, 1280 Main Street West, Hamilton ON, Canada L8S 4K1*
[b] *Facultad de Ciencias de la Empresa, Dept. de Métodos Cuantitativos e Informáticos, Calle Real, 3, 30201 Cartagena, Murcia, España*
[c] *Facultad de Economía y Empresa, Dept. de Métodos Cuantitativos para la Economía y la Empresa, 30100 Murcia, Murcia, España*

**Abstract**

Decision Trees (DTs) are a machine learning technique widely used for regression and classification purposes. Conventionally, the decision boundaries of Decision Trees are orthogonal to the features under consideration. A well-known limitation of this is that the algorithm may fail to find optimal partitions, or in some cases any partitions at all, depending on the underlying distribution of the data. To remedy this limitation, several modifications have been proposed that allow for oblique decision boundaries. The objective of this paper is to propose a new strategy for generating flexible decision boundaries by means of interactive basis functions (IBFs). We show how oblique decision boundaries can be obtained as a particular case of IBFs, and in addition how non-linear decision boundaries can be induced. One attractive aspect of the strategy proposed in this paper is that training Decision Trees with IBFs does not require custom software, since the functions can be precalculated for use in any existing implementation of the algorithm. Since the underlying mechanisms remain unchanged there is no substantial computational overhead compared to conventional trees. Furthermore, this also means that IBFs can be used in any extensions of the Decision Tree algorithm, such as evolutionary trees, boosting, and bagging. We conduct a benchmarking exercise to understand under which conditions the use of IBFs can improve model the performance. In addition, we present three empirical applications that illustrate the approach in classification and regression. As part of discussing the empirical applications, we introduce a device called *decision charts* to facilitate the interpretation of DTs with IBFs. Finally, we conclude the paper by outlining some directions for future research.

*Corresponding Author
*Email addresses:* paezha@mcmaster.ca (Antonio Paez), fernando.lopez@upct.es (Fernando Lopez), manuel.ruiz@upct.es (Manuel Ruiz), m.camacho@um.es (Maximo Camacho)

## 1 Introduction

Decision Trees (DTs) are a popular machine learning technique used both for regression and classification purposes (Loh 2011; James et al. 2013). A DT is trained by means of a training dataset that provides a set of independent variables (or *features*) used to create recursive partitions of the decision space. This is achieved by locally optimizing at each step a loss function that depends on the type of problem (i.e., regression or classification) and/or the specific implementation of the algorithm (i.e., an entropy function for C4.5 and a gini index for CART; see Loh 2011).

Decision Trees find applications in a variety of domains, including, inter alia, transportation (e.g., Ghasri, Rashidi, and Waller 2017), physical geography (e.g., Praskievicz 2018), engineering (e.g., Bektas, Carriquiry, and Smadi 2013; Suhail, Denton, and Zwiggelaar 2018), and environmental sciences (e.g., Choubin et al. 2018). There are several characteristics that make DTs an appealing modeling approach. Notably, DTs are more intuitive than linear/logistic regression (James et al. 2013, 315) and have much greater interpretability than, for instance, artificial neural networks and support vector machines (Yang et al. 2017, 354). In addition, in some settings DTs provide a reasonable heuristic for human decision making (James et al. 2013, 315). Finally, although no single technique can be expected to be uniformly superior in every case, the performance of DTs has been shown to be competitive with, and in some cases superior to, alternatives such as linear regression, logistic regression, support vector machines, and artificial neural networks (e.g., Kurt, Ture, and Kurum 2008; Choubin et al. 2018; Yang et al. 2017).

One characteristic of DTs as conventionally implemented, is that partitions of the variable space are usually done orthogonally to the features. In this way, the partitions are a set of rectangular $p$-dimensional prisms,or hyperboxes. While this is done to reduce the search space of the algorithm, it has the downside that it may fail to find appropriate partitions, and in some extreme cases, to find any partitions at all. In this case, the performance of the algorithm tends to be mediocre. Accordingly, a number of proposals have aimed at ameliorating this situation by inducing oblique partitions (e.g., Murthy, Kasif, and Salzberg 1994; Wickramarachchi et al. 2016; Cantu-Paz and Kamath 2003) and oblique decision tree ensemble variants (e.g., Menze et al. 2011; Zhang and Suganthan 2014; Zhang and Suganthan 2015; Zhang and Suganthan 2017; Zhang et al. 2017; Qiu et al. 2017).

The objective of this paper is to introduce a novel strategy for non-orthogonal partition of variable space in the training of DTs. The approach is based on the use of interactive basis functions (IBFs). We will show that oblique partitions result as a particular case of an IBF. Moreover, depending on the basis function selected, non-linear partitions are also possible. The modeling strategy proposed in this paper is attractive because the basis functions can be precalculated and then used as an input to any decision tree algorithm. Since only the inputs to the algorithm change, this implies that 1) the underlying algorithm is not changed and therefore any existing DT software can be used; and 2) basis functions can

be used in many existing implementations of DTs, including evolutionary trees, bagging, and boosting.

The structure of the paper is as follows. In the [background section][Background] we first review some technical aspects of decision trees and motivate the problem. This is followed by a discussion of [basis functions][Interactive Basis Functions] and how they can be employed to induce oblique linear and non-linear partitions. Next, we discuss some [practical aspects][Practical Considerations] of implementing IBFs before conducting a [benchmarking experiment][Benchmarking] the assess the performance or DTs with IBFs by means of a set of publicly available empirical datasets. The results indicate that inducing oblique and/or non-linear partitions using basis functions can improve the performance of the technique and/or produce more parsimonious models. We then illustrating the [application][Sample Applications] of IBFs by means of three empirical examples. Finally, we [conclude][Conclusions and Directions for Future Research] the paper by summarizing our findings and suggesting some directions for future research.

Given the simplicity and ease of implementation of the modeling strategy, the development presented in this paper should be of interest for users of DTs who wish to improve the performance of their models at a relatively modest computational cost.

## 2 Background

We begin in this section by formally describing the DT algorithm. This will serve to motivate subsequent sections of the paper.

A decision tree is a regression/classification algorithm which models a response variable $Y$ (which could be quantitative or qualitative) from a vector of $p$ features, $X = (X_1, \ldots, X_p)$, where $X \in \mathbb{R}^p$. The algorithm operates by recursively partitioning the space of the features into a set of $M$ regions $R_m$, $m = 1, ..., M$.

To state the basic notation, let us start by considering a binary tree model with $M$ terminal nodes, each of which $m \in \{1, ..., M\}$ represents a branch of the tree that is characterized by $m^*$ internal splits. The parameter space that characterizes each branch is $\theta_m^{m^*} = \left(v_{m,1}^{d_1}, s_{m,1}, ..., v_{m,m^*}^{d_{m^*}}, s_{m,m^*}\right)$, where $X_{v_{m,i}}$ is the splitting variable at the internal split $i$ of the terminal node $m$, with $v_{mi} \in \{1, ..., P\}$, $d_i = 0$ if $X_{v_{m,i}} < s_{m,i}$ and $d_i = 1$ if $X_{v_{m,i}} > s_{m,i}$, $s_{m,i}$ is the splitting point, and $i \in \{1, ..., m^*\}$. The parameter space of trees with $M$ terminal nodes, $\Theta_M$, is formed by the collection of all the combinations of $\theta_m = \bigcup_{j=1}^{m^*} \theta_m^j$, $\Theta_M = \bigcup_{m=1}^{M} \theta_m$. In this way, a tree is a collection of branches, each of which ends up in a terminal node, that characterizes the partitions or regions through $\theta_m$.

The objective of the recursive partitioning mechanism is to try to find optimal partitions in the search space of the features as follows. The path of the partitions determined by each branch is:

$$\theta_m^j = \theta_m^{j-1} \bigcup \left(v_{m,j}^{d_j}, s_{m,j}\right),$$

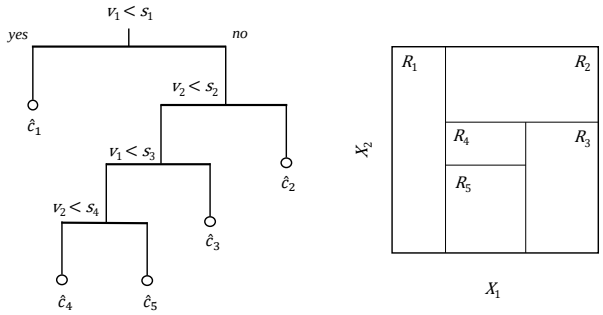


Figure 1: Prototypical Decision Tree

Within this framework, each branch $m$ is a collection of sequential partitions $\theta_m^j$, $j = 1, \ldots, m^*$. For each of these partitions, we define a prediction function $f\left(X, \theta_m^j\right)$, which typically is a central tendency measure of the values of the dependent variable conditional to that region. The goal is to find a decision tree that optimizes some tradeoff between prediction performance, measured by a loss function, $loss\left\{Y, f\left(X, \theta_m^j\right)\right\}$, and a measure of the complexity of the tree, $comp\left(\theta_m^j\right)$, $\hat{\theta}_m^j = \arg\left[loss\left\{Y, f\left(X, \theta_m^j\right)\right\} + comp\left(\theta_m^j\right)\right]$.

In practice, the recursion ends when any additional partitions result in subsamples below a minimum number of cases (e.g., less than five cases). Operating in this way, the recursive partitioning method produces partitions that are orthogonal to the $p$ features, and thus results in a total of $M$ $p$-dimensional rectangular prisms, or hyperboxes (see Figure 1 for a prototypical decision tree with two features $X_1$ and $X_2$).

Despite its advantages, it is not difficult to find examples where orthogonal recursive partitioning does not work. Consider the situation depicted in Figure 2, a simple classification problem with $p = 2$ features. It is straightforward to see that in this case the algorithm fails to find an initial partition, as the loss function on the resulting subsets cannot be reduced by any first split. This situation, on the other hand, is easily avoided if oblique splits are used. In this case an optimal split can be found at the first step, as illustrated by the candidate split shown with a dashed line in the figure.

Motivated by the challenge posed by situations such as the one in Figure 2, and more generally to better capture non-orthogonal decision boundaries, previous research has seen the development of numerous methods to induce oblique decision boundaries for trees (see Cantu-Paz and Kamath 2003; and

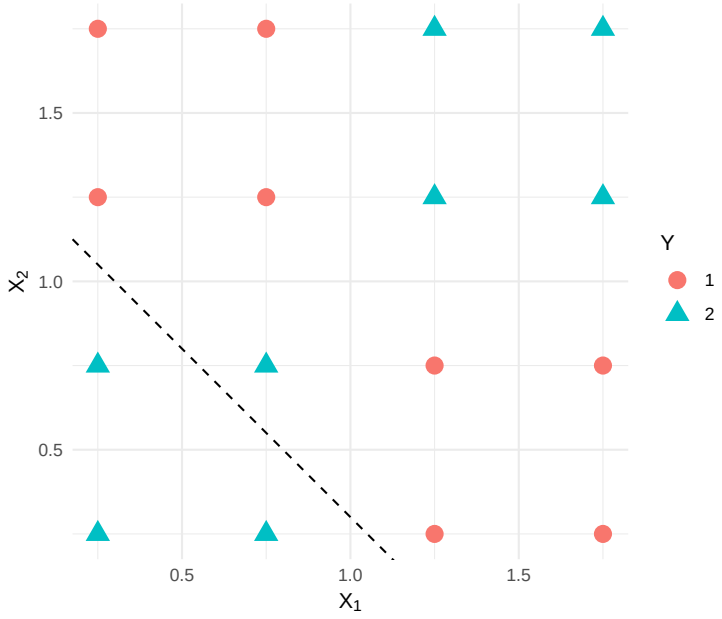


Figure 2: Chessboard example with variable Y with two classes

Wickramarachchi et al. 2016 for an overview). More concretely Menze et al. (2011) propose the use of oblique random forests (oRF) which explicitly learn optimal split directions at internal nodes using linear discriminative models, rather than using random coefficients as the original oRF; Zhang and Suganthan (2014) propose a new method to increase the diversity of each tree in the forests and thereby improve the overall accuracy by building the Random Forests with two projection methods Principal Component Analysis and Linear Discriminate Analysis; Zhang and Suganthan (2015) propose a new method for oblique trees based on multisurface proximal support vector machine; Zhang et al. (2017) propose to use a more powerful proximal SVM to obtain oblique hyperplanes to capture the geometric structure in the context of visual tracking; Zhang and Suganthan (2017) propose an efficient co-trained kernel ridge regression method and a random vector functional link network ensemble that outperform the behavior of classical classifiers; Qiu et al. (2017) study the oblique random forest in the context of time series forecasting by using a least square classifier to perform partitions. Some others excellent reviews on ensemble learning are Dietterich (2000), Rokach (2010), and Ren et al. (2016) where the authors explain why ensembles can often perform better than any single classifier and reviews traditional as well as state-of-the-art ensemble methods.

As discussed by Wickramarachchi et al. (2016), some algorithms used to induce oblique partitions identify candidate partition boundaries based on the

statistical properties of the data. For instance, a number of algorithms use the orientation of the classes as given by their first principal component (Henrichon and King-Sun 1969), the Household projection matrix (Wickramarachchi et al. 2016), or Fisher's linear discriminant (Friedman, Bentley, and Finkel 1977). New features are added to the problem, and the conventional orthogonal partition mechanism is retained. Other algorithms are based on optimization approaches, which could be deterministic or stochastic. CART-LC (Breiman et al. 1984) is an example of the former, whereas Cantu-Paz and Kamath (2003) with their use of genetic algorithms is an example of the latter. The increase in complexity of these algorithms, and consequently their computational burden, comes from the additional steps needed to conduct additional statistical or optimization procedures. Finally, heuristic approaches have also been used, for example by Manwani and Sastry (2012) and Robertson, Price, and Reale (2013).

In the following section, we describe a novel strategy to induce oblique (and possibly non-linear) decision boundaries via the application of *interactive basis functions* (IBFs). This modelling strategy adds judiciously constructed features to the dataset while retaining the conventional orthogonal partition mechanism.

## 3 Interactive Basis Functions (IBFs)

Let us begin our discussion of basis functions by stating that the search domain of a decision tree is contained by the features under consideration. Put in other words, the features used for training constitute the basis vectors for the problem. For instance, when the number of features is $p = 2$, the search space is the plane formed by the orthogonal axes of the features, and each feature is a basis vector. Three features form a 3D basis, and so on. If we consider a feature as a basis vector, a basis function is simply a transformation thereby. In the simplest case, the basis function could be the identity:

$$b(X_1) = X_1$$

which is a particular case of a polynomial function, with $a = 1$:

$$b(X_1) = X_1^a$$

Other basis functions can be defined as well, such as the exponential:

$$b(X_1) = e^{X_1}$$

Basis functions are commonly used in regression analysis, where they have the effect of changing the properties of a regression plane. For instance, a transformation from identity to the square of a variable has the effect of changing the regression line to a parabola. Alas, the use of basis functions in DTs does not have the same effect. To see why this is so, consider the general case with $K$ real functions $b_i : \mathbb{R} \to \mathbb{R}$ with $i = 1, \ldots, K$ candidate basis functions. We will call $\{b_1, b_2, \ldots, b_K\}$ a set of basis functions. Next proceed to enlarge the set of $p$ features with $T$ new features obtained by means of basis functions:

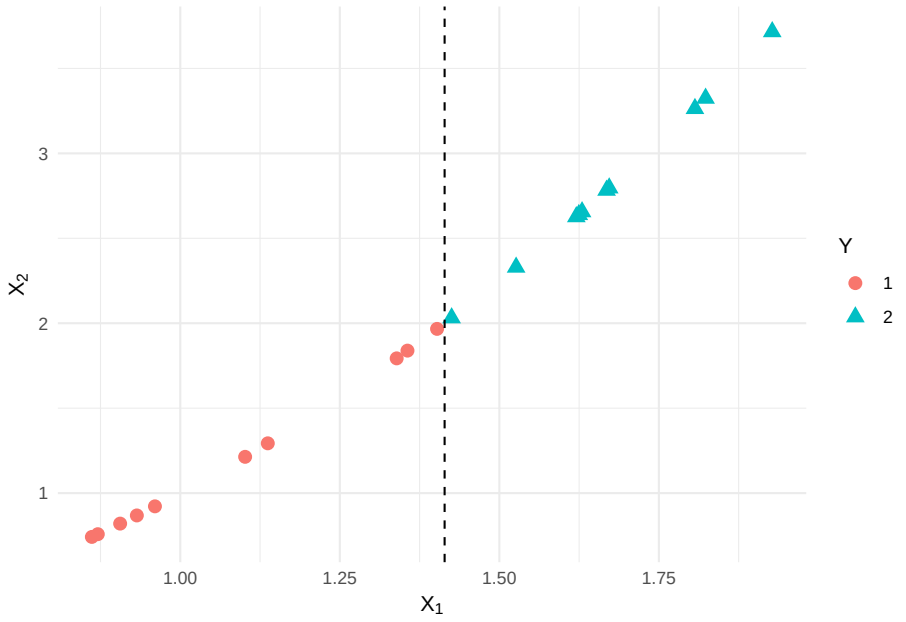


Figure 3: Example of a basis function used as a feature to train a Decision Tree to classify variable Y with two classes

$$X^* = (X_1, \ldots, X_p, X_{p+1}, \ldots, X_{p+T})$$

where $X^* \in \mathbb{R}^{p+T}$, and $X_{p+i} = b_{s_i}(X_{j_i})$, for $i = 1, \ldots, T$, $s_i \in \{1, \ldots, K\}$ and $j_i \in \{1, \ldots, p\}$

Notably, the standard recursive partitioning mechanism of Decision Trees applied to any $X_p$ in the augmented set $T$ leads to partitions of $(p+T)$-dimensional prisms. Furthermore, the projections of these prisms in the subspace of $\mathbb{R}^p$ defined by $X$ are a consequence of a linear orthogothal partition on the original space. Consider an example with $p = 2$, that is $X = \{X_1, X_2\}$. Moreover, $T = 1$ and $K = 1$ with $b_1(x) = x^2$, so that $X^* = \{X_1, X_2, X_3 = X_1^2\}$. Whenever the partitioning mechanism selects a split $s$ in $X_3$ (say, $s = 2$, the solid line in Figure 3), this split is projected in $X$ as $X_1 = \sqrt{X_3}$, which is a constant. As a consequence, any splits on the dimension of the basis function is equivalent to finding an orthogonal decision boundary on the original basis, in the present example at $\sqrt{2}$ in $X_1$ (dashed line).

Since basis functions still produce orthogonal partitions in the original basis, our proposal instead is to use interactive basis functions in the construction of $X^*$ for the tree under consideration. These interactions can be identified by a set of $D$ functions that reproduce functional interactions of the transformations of the features by the basis functions. Thus, these interaction functions are defined

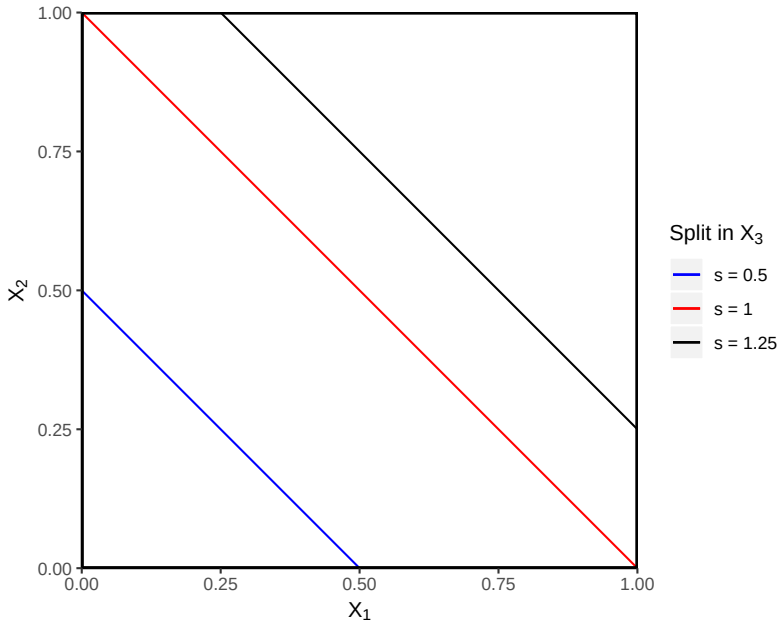


Figure 4: Oblique basis function and projections with different splitting points (s)

as:

$$\begin{array}{rcl} h_i : \mathbb{R}^{pK} & \longrightarrow & \mathbb{R} \\ (b_1(X_1), b_1(X_2) \ldots, b_K(X_p)) & \rightsquigarrow & a \end{array}$$

Under this setting define $X^* = (X_1, \ldots, X_p, X_{p+1}, \ldots, X_{p+D})$, with $X_{p+i} = h_i(b_1(X_1), b_1(X_2) \ldots, b_K(X_p))$, for $i = 1, \ldots, D$. Therefore by applying the standard recursive partitioning method to $X^*$, its projection on $X$ will provide an oblique partition (also eventually non-linear as desired), which will take into account the interactions among the features.

For example, in the case of $p = 2$, that is $X = \{X_1, X_2\}$, $T = 1$, $K = 1$ with $b_1(x) = x$, $D = 1$ with $h_1(b_1(X_1), b_1(X_2)) = b_1(X_1) + b_2(X_2) = X_1 + X_2$ (i.e., an additivie function) and $X^* = (X_1, X_2, X_3)$ we obtain that $X_3 = s$ is projected on the plane of the original basis as $X_2 = s - X_1$, thus providing an oblique partition on that plane as shown in Figure 4.

It is interesting to note that the framework provided by IBFs allows us to induce, in addition to oblique partitions, non-linear decision boundaries as well. This is done by projecting the equation $h_i(b_1(X_1), b_1(X_2) \ldots, b_K(X_p)) = a$ in the subspace $X = \{X_1, X_2, \ldots, X_p\}$ generated by the features in the dataset. For example, the IBF $h_1(b_1(X_1), b_1(X_2)) = b_1(X_1)b_2(X_2)$ with $b_1(x) = x$ leads to $X_1X_2$. A split $s$ in the dimension of this multiplicative IBF fixes $X_1X_2 = s$, and therefore $X_2 = s/X_1$, thus creating a hyperbolic partition, as shown in

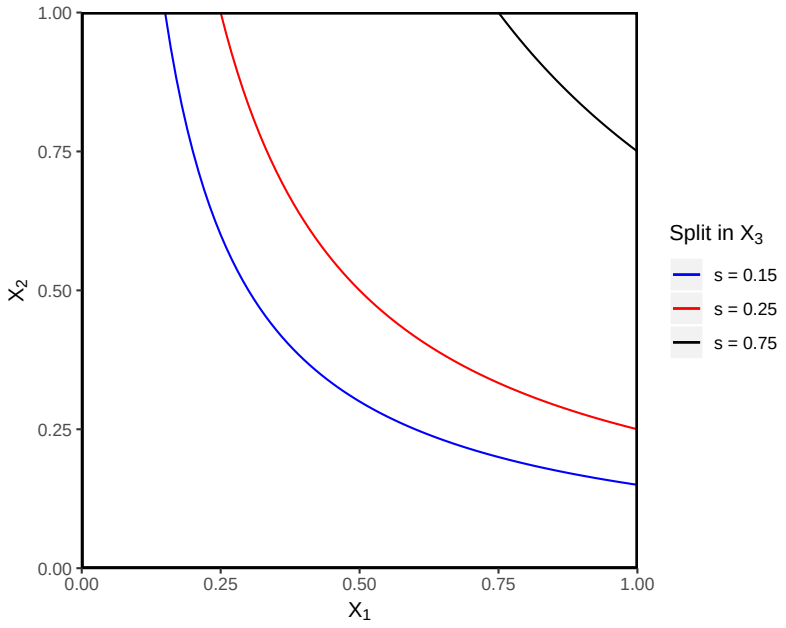


Figure 5: Hyperbolic basis function and projections with different splitting points (s)

Figure 5.

As one final example (see Figure 6), the IBF $h_1(b_1(X_1), b_1(X_2)) = b_1(X_1) + b_2(X_2)$ with $b_1(x) = x^2$ leads to $X_1^2 + X_2^2$. A split $s$ in the dimension of the IBF fixes $X_1^2 + X_2^2 = s$, and therefore $X_2 = \sqrt{s - X_1^2}$, thus creating a radial partition.

It is worthwhile noting at this point that the additive function shown in Figure 4 is similar to the case of linear combinations used in the CART-LC algorithm (see Cantu-Paz and Kamath 2003), however without parameterizing the interactive basis function - or, alternatively, implicitly assuming that $a_1 = a_2 = 1$:

$$h_1(b_1(X_1), b_1(X_2)) = b_1(X_1) + b_2(X_2) = a_1 X_1 + a_2 X_2$$

In this way, the implementation of IBFs as non-parametric functions leads to a semi-parametric version of DTs. With respect to the complexity of the search, the introduction of IBFs increases the complexity at each node of the tree (when using the conventional search algorithm) by $O(n(p + D))$, where $n$ is the number of cases, $p$ is the number of features, and $D$ is the number of IBFs under consideration. In contrast, the exhaustive search for oblique decision boundaries has complexity of $O(2^p \times \binom{n}{p})$ according to Murthy et al. (1994), and $O(Cp^2 nlog(n))$ in HHCART (where $C$ is the number of classes in classification problems, Wickramarachchi et al. 2016). As can be appreciated, the increase in complexity with our approach is fairly modest.

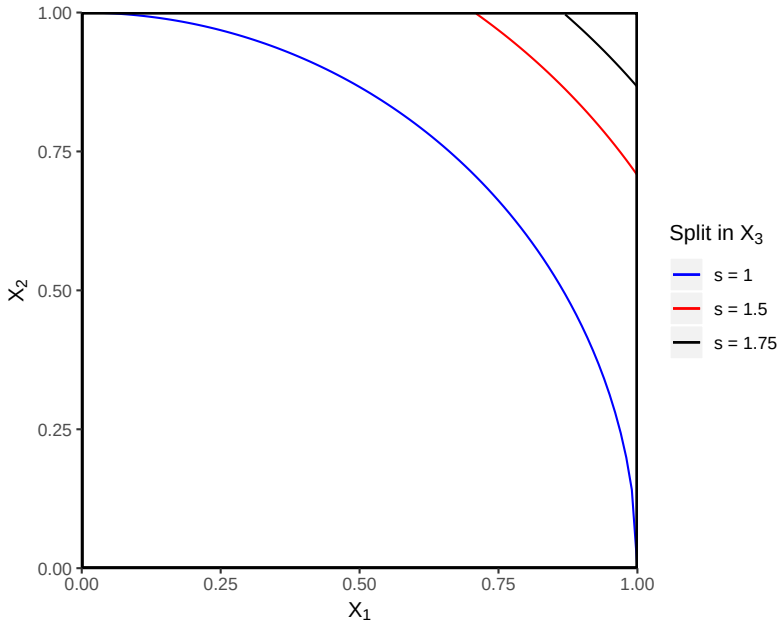


Figure 6: Radial IBF and projections with different splitting points (s)

## 4 Practical Considerations

An important consideration in the practical implementation of IBFs concerns the centering and/or scaling of the input features. Centering and scaling are typically monotonical transformations of the data. For instance, a feature vector can be centered on its minimum value or its mean, thus shifting the origin. Scaling can be done for example by scaling all values to the unit interval, or by dividing by the standard deviation to normalize the scale.

Scaling and centering the features has no impact in the training of DT with linear partitions, whether orthogonal or oblique. Any monotonic transformation of the data will simply shift the splitting point that creates a partition accordingly. The same does not necessarily happen with non-linear partitions. Recall that the curvature $\kappa$ of a plane curve is related to the radius of the curve as follows for a given arc length $l$:

$$\kappa(l) = \frac{1}{R(l)}$$

Intuitively, the curvature of a straight line is constant at zero. When the radius of a curve for $l$ is large the curvature is small, and viceversa. It follows then that the geometric representation of a non-linear partition in the subspace $X$ depends on the origin (i.e., center) and scale in which the independent variables have been measured. This effect is illustrated in Figure 7, where a set of radial

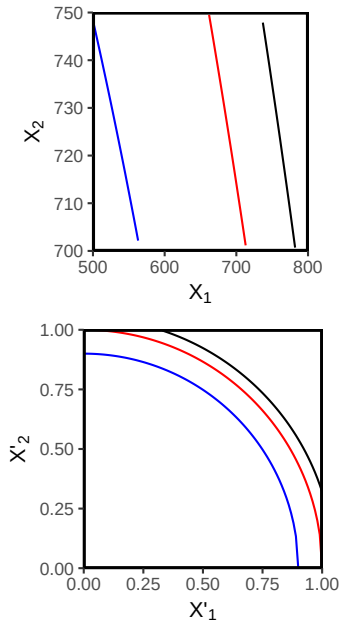


Figure 7: Behavior of radial IBF with non-centered/non-scaled variables (top panel) and centered/scaled variables (bottom panel)

IBFs are plotted. In one case (top panel) the features $X_1$ and $X_2$ are measured in possibly different scales. Since the origin for these variables is at zero, the radius of the curves is large for partitions whithin the space of interest, and therefore the partitions generated are locally quasi-linear. The bottom panel of the figure shows analog radial IBFs, but now on features $X_1'$ and $X_2'$. These features are obtained by centering variables $X_1$ and $X_2$ on their minimum values, and scaling them to the unit length. Now the radii of the IBFs are smaller, thus resulting on greater curvature and locally non-linear partitions.

For this reason, we recommend centering and scaling all features prior to analysis, in order to increase the chances that non-linear partitions are effective, and reduce the possibility that they resemble linear partitions locally.

## 5 Benchmarking

In this section we conduct a benchmarking exercise using a collection of publicly available datasets. The source of the data is the paper of Fernandez-Delgado et al. (2014), who conducted an extensive experiment testing machine learning algorithms. These authors conducted their experiments with 121 datasets. An extensive list of classifiers were applied to these datasets, and then the classifiers were ranked based on their average accuracy and probability that they succeed

by chance (measured by means of Cohen's kappa). Their results indicate that the best classifiers belong to the family of Random Forests and Support Vector Machines.

Our focus in the present paper is on comparing DT-based algorithms only. Fernandez-Delgado et al. (2014) already provide a comprehensive ranking of algorithms, and our objective instead is to assess how IBFs perform in DT, relative to conventional orthogonal partitions. Since one advantage of using IBFs is that they can be used in virtually any implementation of DTs, we choose to illustrate their applicability using three different methods: a conventional tree algorithm (implemented in the R package `tree`, see Ripley 2018), random forest (implemented in the R package `randomForest`, see Liaw and Wiener 2002), and an evolutionary algorithm (implemented in the package `evtree`, see Grubinger, Zeileis, and Pfeiffer 2014). It is worthwhile to note that the top algorithm in the experiments of Fernandez-Delgado (2014) was the parallel implementation of the random forest method; however, these authors did not assess the evolutionary algorithm as we do. Other R packages exist that implement non-orthogonal partitions, however they are limited to binary classification (in the `obliqueRF` package of Menze and Splitthoff 2012) or to two features (in the spatial oblique decision trees package `SPODT` of Gaudart et al. 2015).

The experiments were conducted using a parallel implementation of the code, using 4 Intel i7-6600U cores clocked at 2.60 GHz. For the present study we use a subset of the 121 datasets from the collection of Fernandez-Delgado et al. (2014). After excluding a number of datasets that led to errors when running the evolutionary tree algorithm, we work with 93 datasets. The main reason for the errors was a large number of observations, which may be a limitation of our computer resources rather than the algorithm. Table 1 lists the characteristics of the datasets, including the number of observations ($n$), the number of features ($f$), the number of classes ($k$) of the independent variable, and the proportion of the majority class in the dataset ($m$).

Table 1: Datasets for Benchmarking Experiment (n: number of observations, f: number of features, k: number of classes, m: proportion of majority class)

| Dataset # | Name | n | f | k | m |
|---|---|---|---|---|---|
| 1 | abalone | 4177 | 8 | 3 | 0.35 |
| 2 | acute-inflammation | 120 | 6 | 2 | 0.51 |
| 3 | acute-nephritis | 120 | 6 | 2 | 0.58 |
| 4 | adult_train | 32561 | 14 | 2 | 0.76 |
| 5 | annealing_train | 798 | 31 | 5 | 0.76 |
| 6 | audiology-std_train | 171 | 59 | 18 | 0.26 |
| 7 | balance-scale | 625 | 4 | 3 | 0.46 |
| 8 | bank | 4521 | 16 | 2 | 0.88 |
| 9 | blood | 748 | 4 | 2 | 0.76 |

Table 1: Datasets for Benchmarking Experiment (n: number of observations, f: number of features, k: number of classes, m: proportion of majority class) *(continued)*

| Dataset # | Name | n | f | k | m |
|---|---|---|---|---|---|
| 10 | breast-cancer | 286 | 9 | 2 | 0.70 |
| 11 | breast-cancer-wisc | 699 | 9 | 2 | 0.66 |
| 12 | breast-cancer-wisc-diag | 569 | 30 | 2 | 0.63 |
| 13 | breast-cancer-wisc-prog | 198 | 33 | 2 | 0.76 |
| 14 | breast-tissue | 106 | 9 | 6 | 0.21 |
| 15 | car | 1728 | 6 | 4 | 0.70 |
| 16 | cardiotocography-10clases | 2126 | 21 | 10 | 0.27 |
| 17 | cardiotocography-3clases | 2126 | 21 | 3 | 0.78 |
| 18 | chess-krvk | 28056 | 6 | 18 | 0.16 |
| 19 | chess-krvkp | 3196 | 36 | 2 | 0.52 |
| 20 | congressional-voting | 435 | 16 | 2 | 0.61 |
| 21 | conn-bench-sonar-mines-rocks | 208 | 60 | 2 | 0.53 |
| 22 | conn-bench-vowel-deterding_train | 528 | 11 | 11 | 0.09 |
| 23 | contrac | 1473 | 9 | 3 | 0.43 |
| 24 | credit-approval | 690 | 15 | 2 | 0.56 |
| 25 | cylinder-bands | 512 | 35 | 2 | 0.61 |
| 26 | dermatology | 366 | 34 | 6 | 0.31 |
| 27 | echocardiogram | 131 | 10 | 2 | 0.67 |
| 28 | ecoli | 336 | 7 | 8 | 0.43 |
| 29 | energy-y1 | 768 | 8 | 3 | 0.47 |
| 30 | energy-y2 | 768 | 8 | 3 | 0.50 |
| 31 | fertility | 100 | 9 | 2 | 0.88 |
| 32 | flags | 194 | 28 | 8 | 0.31 |
| 33 | glass | 214 | 9 | 6 | 0.36 |
| 34 | haberman-survival | 306 | 3 | 2 | 0.74 |
| 35 | hayes-roth_train | 132 | 3 | 3 | 0.39 |
| 36 | heart-cleveland | 303 | 13 | 5 | 0.54 |
| 37 | heart-hungarian | 294 | 12 | 2 | 0.64 |
| 38 | heart-switzerland | 123 | 12 | 5 | 0.39 |
| 39 | heart-va | 200 | 12 | 5 | 0.28 |
| 40 | hepatitis | 155 | 19 | 2 | 0.79 |
| 41 | horse-colic_train | 300 | 25 | 2 | 0.64 |
| 42 | ilpd-indian-liver | 583 | 9 | 2 | 0.71 |
| 43 | image-segmentation_train | 210 | 18 | 7 | 0.14 |
| 44 | ionosphere | 351 | 33 | 2 | 0.64 |
| 45 | iris | 150 | 4 | 3 | 0.33 |

Table 1: Datasets for Benchmarking Experiment (n: number of observations, f: number of features, k: number of classes, m: proportion of majority class) *(continued)*

| Dataset # | Name | n | f | k | m |
|---|---|---|---|---|---|
| 46 | led-display | 1000 | 7 | 10 | 0.11 |
| 47 | lenses | 24 | 4 | 3 | 0.62 |
| 48 | letter | 20000 | 16 | 26 | 0.04 |
| 49 | libras | 360 | 90 | 15 | 0.07 |
| 50 | lung-cancer | 32 | 56 | 3 | 0.41 |
| 51 | lymphography | 148 | 18 | 4 | 0.55 |
| 52 | magic | 19020 | 10 | 2 | 0.65 |
| 53 | mammographic | 961 | 5 | 2 | 0.54 |
| 54 | molec-biol-promoter | 106 | 57 | 2 | 0.50 |
| 55 | molec-biol-splice | 3190 | 60 | 3 | 0.52 |
| 56 | monks-1_train | 124 | 6 | 2 | 0.50 |
| 57 | monks-2_train | 169 | 6 | 2 | 0.62 |
| 58 | monks-3_train | 122 | 6 | 2 | 0.51 |
| 59 | mushroom | 8124 | 21 | 2 | 0.52 |
| 60 | nursery | 12960 | 8 | 5 | 0.33 |
| 61 | optical_train | 3823 | 62 | 10 | 0.10 |
| 62 | ozone | 2536 | 72 | 2 | 0.97 |
| 63 | page-blocks | 5473 | 10 | 5 | 0.90 |
| 64 | parkinsons | 195 | 22 | 2 | 0.75 |
| 65 | pendigits_train | 7494 | 16 | 10 | 0.10 |
| 66 | pima | 768 | 8 | 2 | 0.65 |
| 67 | pittsburg-bridges-MATERIAL | 106 | 7 | 3 | 0.75 |
| 68 | pittsburg-bridges-REL-L | 103 | 7 | 3 | 0.51 |
| 69 | pittsburg-bridges-SPAN | 92 | 7 | 3 | 0.52 |
| 70 | pittsburg-bridges-T-OR-D | 102 | 7 | 2 | 0.86 |
| 71 | pittsburg-bridges-TYPE | 105 | 7 | 6 | 0.42 |
| 72 | planning | 182 | 12 | 2 | 0.71 |
| 73 | post-operative | 90 | 8 | 3 | 0.71 |
| 74 | primary-tumor | 330 | 17 | 15 | 0.25 |
| 75 | ringnorm | 7400 | 20 | 2 | 0.50 |
| 76 | seeds | 210 | 7 | 3 | 0.33 |
| 77 | soybean_train | 307 | 35 | 18 | 0.13 |
| 78 | spambase | 4601 | 57 | 2 | 0.61 |
| 79 | spect_train | 79 | 22 | 2 | 0.67 |
| 80 | spectf_train | 80 | 44 | 2 | 0.50 |
| 81 | statlog-australian-credit | 690 | 14 | 2 | 0.68 |

Table 1: Datasets for Benchmarking Experiment (n: number of observations, f: number of features, k: number of classes, m: proportion of majority class) *(continued)*

| Dataset # | Name | n | f | k | m |
|---|---|---|---|---|---|
| 82 | statlog-german-credit | 1000 | 24 | 2 | 0.70 |
| 83 | statlog-heart | 270 | 13 | 2 | 0.56 |
| 84 | statlog-image | 2310 | 18 | 7 | 0.14 |
| 85 | statlog-landsat_train | 4435 | 36 | 6 | 0.24 |
| 86 | statlog-shuttle_train | 43500 | 9 | 7 | 0.78 |
| 87 | statlog-vehicle | 846 | 18 | 4 | 0.26 |
| 88 | steel-plates | 1941 | 27 | 7 | 0.35 |
| 89 | synthetic-control | 600 | 60 | 6 | 0.17 |
| 90 | teaching | 151 | 5 | 3 | 0.34 |
| 91 | thyroid_train | 3772 | 21 | 3 | 0.92 |
| 92 | tic-tac-toe | 958 | 9 | 2 | 0.65 |
| 93 | titanic | 2201 | 3 | 2 | 0.68 |

As noted above, centering and scaling the variables is important for the use of IBFs. The variables in the datasets provided by Fernandez-Delgado et al. (2014) have already been normalized to have a mean of zero and a standard deviation of one (p. 3139). We use the variables as they are, rather than changing them. The experiments are conducted using $k$-fold cross-validation, a technique commonly used to assess the performance of algorithms. $k$-fold cross-validation involves randomly thinning a dataset to obtain a $(100 - 100/k)\%$ training sample, in a procedure that is repeated $k$ times. In line with the experiments of Fernandez-Delgado et al. (2014) in this section we use 4-fold cross-validation.

In terms of the IBFs, we consider the following functions:

$$h_1(b_1(X_1), b_1(X_2)) = b_1(X_1) + b_1(X_2) = X_1 + X_2$$
$$h_1(b_2(X_1), b_2(X_2)) = b_2(X_1) + b_2(X_2) = X_1^2 + X_2^2$$
$$h_2(b_1(X_1), b_1(X_2)) = b_1(X_1)b_2(X_2) = X_1X_2$$
$$h_2(b_1(X_1)b_3(X_2)) = b_1(X_1)b_3(X_2) = X_1e^{X_2}$$

The size of the experiment is as follows: three different methods (tree/random forest/evolutionary tree), ninety three different datasets, two different types of partitions (orthogonal/IBFs), and four replications for each (i.e., 4-fold cross-validation). The summary of the results is shown in Table 2, where it can be seen that random forests outperform evolutionary trees, which in turn outperform conventional trees. This is as anticipated. Further, for the same algorithm, the use of IBF increases the average accuracy slightly, and provides similar results in terms of kappa and mean tree size.

Table 2: Summary of Benchmarking Experiment

| Method | Basis Function | Mean Accuracy (%) | Mean kappa | Mean Tree Size |
|---|---|---|---|---|
| Tree | Orthogonal | 77.13 | 0.51 | 12.25 |
| Tree | IBF | 77.71 | 0.53 | 12.82 |
| Forest | Orthogonal | 84.09 | 0.61 | 0.00 |
| Forest | IBF | 84.11 | 0.62 | 0.00 |
| Evolutionary Tree | Orthogonal | 79.05 | 0.55 | 9.74 |
| Evolutionary Tree | IBF | 79.25 | 0.54 | 9.69 |

*Note:*
Tree Size is not meaningful in the case of random forest, since the model is an ensemble of trees.

Table 3: Accuracy by Dataset

| | Tree | | Forest | | Evolutionary Tree | |
|---|---|---|---|---|---|---|
| Dataset # | Orthogonal | IBF | Orthogonal | IBF | Orthogonal | IBF |
| 1 | 61.69 | 62.07 | 65.18 | 64.39 | 64.30 | 63.86 |
| 2 | 100.00 | 100.00 | 100.00 | 100.00 | 100.00 | 100.00 |
| 3 | 100.00 | 100.00 | 100.00 | 100.00 | 98.33 | 100.00 |
| 4 | 94.72 | 95.60 | 96.98 | 97.36 | 90.33 | 90.95 |
| 5 | 70.35 | 68.02 | 83.72 | 80.81 | 69.77 | 72.09 |
| 6 | 78.21 | 86.70 | 86.86 | 87.66 | 78.69 | 86.70 |
| 7 | 88.98 | 89.49 | 90.66 | 90.24 | 90.49 | 89.80 |
| 8 | 78.48 | 76.74 | 76.87 | 77.94 | 75.40 | 78.07 |
| 9 | 76.76 | 63.73 | 74.65 | 73.94 | 76.76 | 73.59 |
| 10 | 95.29 | 94.71 | 97.57 | 97.14 | 94.14 | 96.14 |
| 11 | 92.78 | 94.54 | 95.42 | 97.01 | 93.13 | 95.25 |
| 12 | 69.39 | 71.43 | 83.16 | 82.14 | 72.45 | 75.51 |
| 13 | 73.08 | 70.19 | 75.00 | 76.92 | 75.00 | 65.38 |
| 14 | 93.63 | 95.20 | 97.86 | 99.42 | 93.34 | 95.83 |
| 15 | 76.18 | 81.12 | 87.81 | 87.66 | 82.39 | 80.56 |
| 16 | 93.55 | 91.53 | 95.01 | 95.06 | 93.60 | 91.57 |
| 17 | 97.72 | 99.00 | 98.65 | 99.34 | 96.81 | 96.25 |
| 18 | 74.04 | 65.38 | 84.62 | 82.21 | 71.63 | 67.31 |
| 19 | 62.88 | 66.29 | 96.02 | 96.59 | 65.72 | 65.53 |
| 20 | 53.94 | 53.46 | 54.62 | 52.92 | 56.93 | 54.21 |
| 21 | 84.88 | 83.14 | 87.79 | 88.52 | 85.90 | 86.19 |
| 22 | 73.83 | 74.41 | 81.45 | 78.52 | 69.53 | 72.85 |
| 23 | 93.41 | 97.53 | 98.08 | 98.63 | 95.60 | 94.78 |
| 24 | 75.00 | 78.79 | 83.33 | 84.09 | 85.61 | 84.85 |
| 25 | 82.44 | 83.93 | 87.50 | 87.50 | 84.82 | 82.44 |
| 26 | 94.01 | 95.18 | 95.57 | 97.66 | 92.45 | 95.18 |

Table 3: Accuracy by Dataset *(continued)*

| Dataset # | Orthogonal | IBF | Orthogonal | IBF | Orthogonal | IBF |
|---|---|---|---|---|---|---|
| 27 | 88.54 | 86.59 | 89.19 | 91.28 | 90.36 | 89.97 |
| 28 | 86.00 | 79.00 | 90.00 | 90.00 | 88.00 | 87.00 |
| 29 | 58.85 | 54.69 | 67.71 | 70.31 | 59.38 | 60.42 |
| 30 | 64.62 | 66.04 | 81.13 | 75.94 | 71.23 | 64.62 |
| 31 | 70.72 | 62.83 | 70.72 | 66.45 | 73.68 | 71.38 |
| 32 | 81.82 | 80.30 | 82.58 | 81.06 | 66.67 | 78.03 |
| 33 | 56.25 | 51.64 | 55.59 | 55.26 | 56.25 | 55.26 |
| 34 | 79.11 | 77.74 | 83.22 | 81.85 | 82.19 | 76.37 |
| 35 | 26.61 | 30.65 | 43.55 | 42.74 | 40.32 | 34.68 |
| 36 | 25.50 | 32.50 | 37.50 | 35.50 | 29.00 | 31.00 |
| 37 | 78.21 | 71.15 | 80.77 | 77.56 | 73.08 | 71.15 |
| 38 | 88.33 | 83.33 | 87.67 | 87.00 | 84.00 | 82.33 |
| 39 | 70.03 | 67.29 | 71.40 | 73.12 | 69.35 | 70.38 |
| 40 | 81.73 | 84.13 | 94.71 | 95.67 | 85.58 | 83.17 |
| 41 | 88.07 | 91.48 | 94.60 | 94.32 | 89.77 | 90.91 |
| 42 | 95.95 | 97.97 | 95.27 | 95.27 | 95.27 | 95.27 |
| 43 | 49.72 | 58.89 | 80.56 | 81.11 | 50.28 | 55.56 |
| 44 | 74.32 | 75.00 | 88.51 | 87.16 | 72.97 | 79.05 |
| 45 | 79.90 | 79.27 | 79.90 | 78.12 | 80.52 | 79.79 |
| 46 | 80.77 | 73.08 | 91.35 | 94.23 | 76.92 | 80.77 |
| 47 | 87.55 | 92.35 | 95.14 | 94.39 | 91.50 | 90.81 |
| 48 | 73.39 | 100.00 | 90.32 | 96.77 | 89.52 | 87.10 |
| 49 | 61.90 | 60.71 | 67.26 | 77.98 | 61.90 | 72.02 |
| 50 | 92.50 | 91.67 | 93.33 | 91.67 | 92.50 | 92.50 |
| 51 | 99.82 | 99.93 | 100.00 | 100.00 | 99.98 | 100.00 |
| 52 | 78.66 | 84.07 | 97.91 | 97.86 | 87.68 | 85.96 |
| 53 | 95.78 | 95.23 | 97.04 | 97.12 | 97.16 | 97.16 |
| 54 | 96.02 | 96.53 | 97.28 | 97.20 | 96.33 | 96.36 |
| 55 | 84.69 | 92.35 | 89.80 | 90.82 | 86.22 | 91.33 |
| 56 | 82.54 | 89.40 | 99.08 | 99.31 | 93.77 | 93.55 |
| 57 | 71.74 | 71.09 | 75.26 | 75.39 | 75.39 | 74.48 |
| 58 | 85.58 | 80.77 | 91.35 | 91.35 | 89.42 | 89.42 |
| 59 | 65.38 | 66.35 | 70.19 | 65.38 | 75.00 | 64.42 |
| 60 | 59.78 | 58.70 | 67.39 | 68.48 | 55.43 | 73.91 |
| 61 | 89.00 | 85.00 | 90.00 | 89.00 | 88.00 | 84.00 |
| 62 | 50.00 | 54.81 | 70.19 | 65.38 | 62.50 | 64.42 |
| 63 | 52.78 | 53.33 | 70.00 | 69.44 | 67.22 | 62.78 |
| 64 | 63.64 | 48.86 | 67.05 | 63.64 | 72.73 | 47.73 |

Table 3: Accuracy by Dataset *(continued)*

| Dataset # | Orthogonal | IBF | Orthogonal | IBF | Orthogonal | IBF |
|---|---|---|---|---|---|---|
| 65 | 48.17 | 40.85 | 52.44 | 50.91 | 49.70 | 47.87 |
| 66 | 86.76 | 84.41 | 95.70 | 97.99 | 87.20 | 89.89 |
| 67 | 91.83 | 88.94 | 93.75 | 93.75 | 88.46 | 88.46 |
| 68 | 81.49 | 87.99 | 94.81 | 93.83 | 81.82 | 87.99 |
| 69 | 89.46 | 91.02 | 95.30 | 94.76 | 92.43 | 91.89 |
| 70 | 67.50 | 70.00 | 78.75 | 80.00 | 77.50 | 72.50 |
| 71 | 65.41 | 59.30 | 65.70 | 64.39 | 67.01 | 66.42 |
| 72 | 72.60 | 70.70 | 77.50 | 78.40 | 76.20 | 73.00 |
| 73 | 80.22 | 75.00 | 83.21 | 85.82 | 82.46 | 80.60 |
| 74 | 92.76 | 93.98 | 97.88 | 98.48 | 95.23 | 94.67 |
| 75 | 80.16 | 83.34 | 89.95 | 89.61 | 84.94 | 84.38 |
| 76 | 71.45 | 71.92 | 74.41 | 77.84 | 70.50 | 70.26 |
| 77 | 66.60 | 70.57 | 78.45 | 79.12 | 72.84 | 72.16 |
| 78 | 88.50 | 88.67 | 99.00 | 97.83 | 86.67 | 88.00 |
| 79 | 51.97 | 59.21 | 69.08 | 67.11 | 57.24 | 56.58 |
| 80 | 99.63 | 99.47 | 99.63 | 99.73 | 99.60 | 99.71 |
| 81 | 87.24 | 93.62 | 99.06 | 98.54 | 83.16 | 89.75 |
| 82 | 77.73 | 78.95 | 78.73 | 78.95 | 78.68 | 78.95 |
| 83 | 77.26 | 85.86 | 97.35 | 97.81 | 82.99 | 89.22 |
| 84 | 75.97 | 80.84 | 80.52 | 85.06 | 76.30 | 83.77 |
| 85 | 80.84 | 85.39 | 85.39 | 86.69 | 81.49 | 86.69 |
| 86 | 99.05 | 97.42 | 99.41 | 98.81 | 99.67 | 98.28 |
| 87 | 71.14 | 76.62 | 84.98 | 85.24 | 77.54 | 79.84 |
| 88 | 72.94 | 76.04 | 85.88 | 85.98 | 78.68 | 79.10 |
| 89 | 91.48 | 97.73 | 98.30 | 98.30 | 94.32 | 96.59 |
| 90 | 56.25 | 60.44 | 69.62 | 70.62 | 60.25 | 60.56 |
| 91 | 50.35 | 51.59 | 68.36 | 68.57 | 53.74 | 54.92 |
| 92 | 56.40 | 58.09 | 61.39 | 61.46 | 58.69 | 59.10 |
| 93 | 91.00 | 86.00 | 97.00 | 98.00 | 86.00 | 89.00 |

The summary in Table 2 indicates that on average IBFs perform only slightly better compared to orthogonal partitions. Conducting a Friedman ranking test of the results of the experiments confirms that there are significant differences among the three classification methods (i.e., tree, forest, evolutionary trees; $Q = 199$, $p$-val $\approx$ 5.941e-41). However, when the test is conducted between orthogonal and IBF *within* a given classification method, the results are not significant ($Q = 2.5$, $p$-val $\approx 0.1159$ for tree; $Q = 0.19$, $p$-val $\approx 0.6625$ for forest; $Q = 0.1$, $p$-val $\approx 0.7477$ for evolutionary tree).

These results would suggest that it does not matter whether orthogonal

partitions or IBFs are used in any given analysis. An important limitation of this conclusion is that it assumes that all datasets are comparable. As seen in Table 1, however, the datasets in the experiment vary widely in terms of their number of observations and composition (number of features, classes, and proportion of majority class). Inspection of Table 3, where the accuracy results by dataset are reported, shows that IBFs sometimes out-perform orthogonal partitions - but does this happen haphazardly, or on the contrary, in a systematic way? To overcome the limitation of the Friedman test, which ignores the differences in the datasets, in the following section we take a multivariate approach to delve more deeply into the results of the benchmarking experiments.

*5.1 Accuracy*

As discussed above, the summary statistics of the results suggest that IBFs improve the accuracy of the algorithms only slightly, albeit without necessarily leading to more complex models. In this subsection, we take a multivariate approach to further investigate the results of the benchmarking experiment in terms of the accuracy of the different implementations of the algorithms. In other words, we are interested in whether the choice between orthogonal partitions and IBFs is simply a matter of a coin toss, or whether we can distinguish circumstances in which one approach is consistently better than the other.

We begin by visualizing some of the results.

Figure 8 presents the results of the experiments by method in terms of their accuracy. The horizontal axis is for datasets, ranked from the lowest average accuracy of all methods, to the highest. The plot shows the mean accuracy and standard deviation of each algorithm for each dataset. It can be seen there that the lowest accuracy is around 25% and the highest is 100% in the test sample. In general terms, as hinted by the summary results, random forests perform better and trees worse, but this is not always the case. Moreover, as will be discussed next, there are notable differences between the use of orthogonal and IBF within each method.

Figure 9 compares the results of implementing the tree algorithm with orthogonal partitions and IBFs. Taking as a baseline the accuracy of the models with orthogonal partitions, the datasets are ranked from the largest accuracy loss to the largest accuracy gain attained by the models with IBFs. As seen in the figure, the implementation with IBFs leads to accuracy gains in 53 out of 93 datasets, that is, just over 50% of the time - which would explain why the Friedman test is not significant. On the other hand, the maximum accuracy gain is 26.61% whereas the maximum loss is -14.77%. In other words, it is not only a matter of whether IBFs perform better, but also how much better.

With respect to the random forest algorithm, Figure 10 compares the results with orthogonal partitions and IBFs. Again, taking as a baseline the accuracy of the models with orthogonal partitions, the datasets are ranked from the largest accuracy loss to the largest accuracy gain attained by the models with IBFs. As seen in the figure, the implementation with IBFs leads to accuracy gains in 44 out of 93 datasets, just below 50% of the time. The maximum accuracy gain is 10.71% whereas the maximum loss is -5.189%.

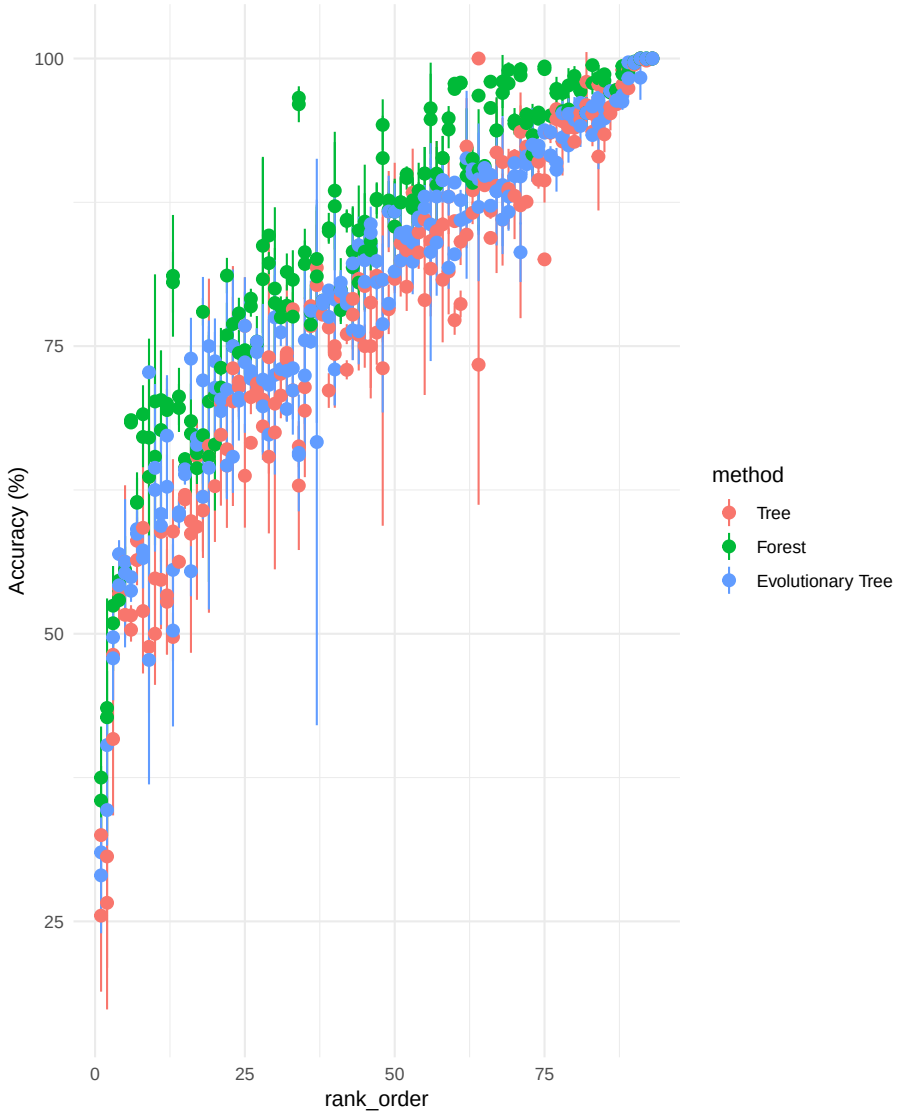


Figure 8: Benchmarking results: Classification accuracy by Algorithm

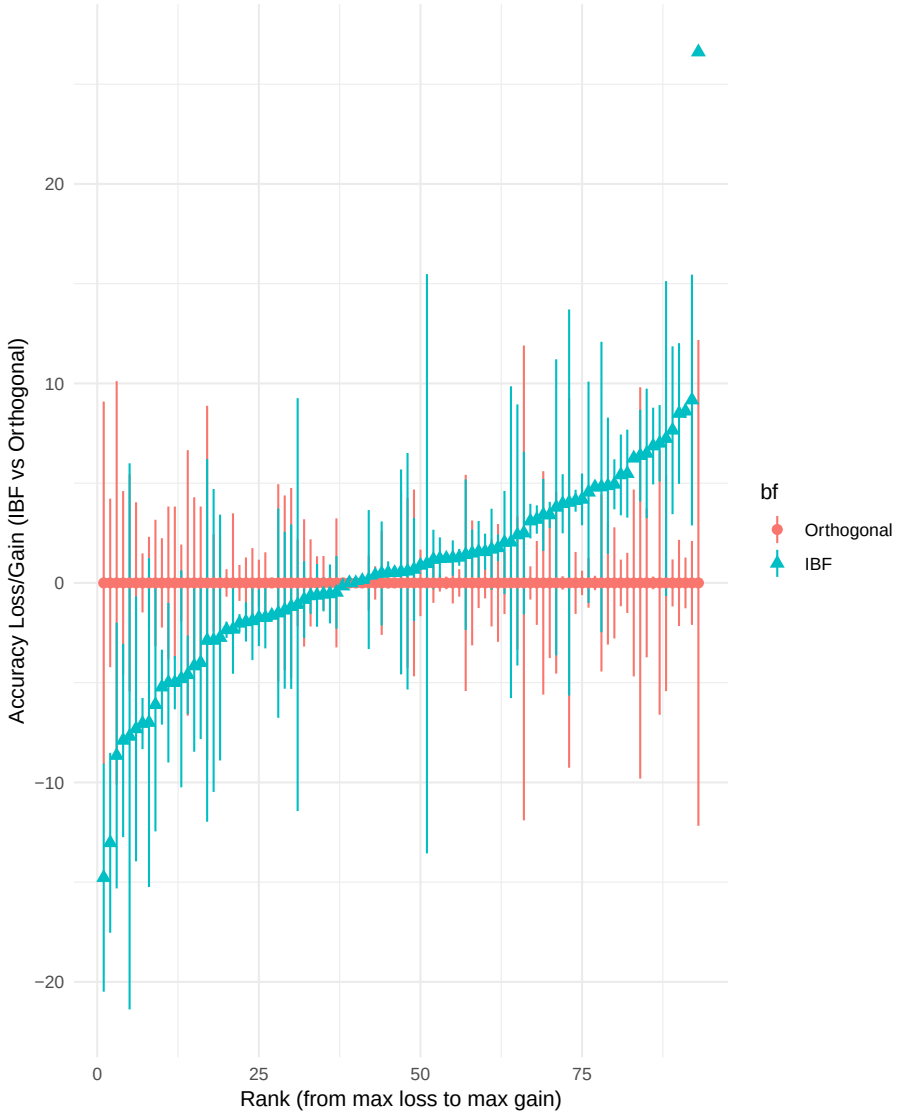


Figure 9: Accuracy comparison for method tree: orthogonal and IBF implementations

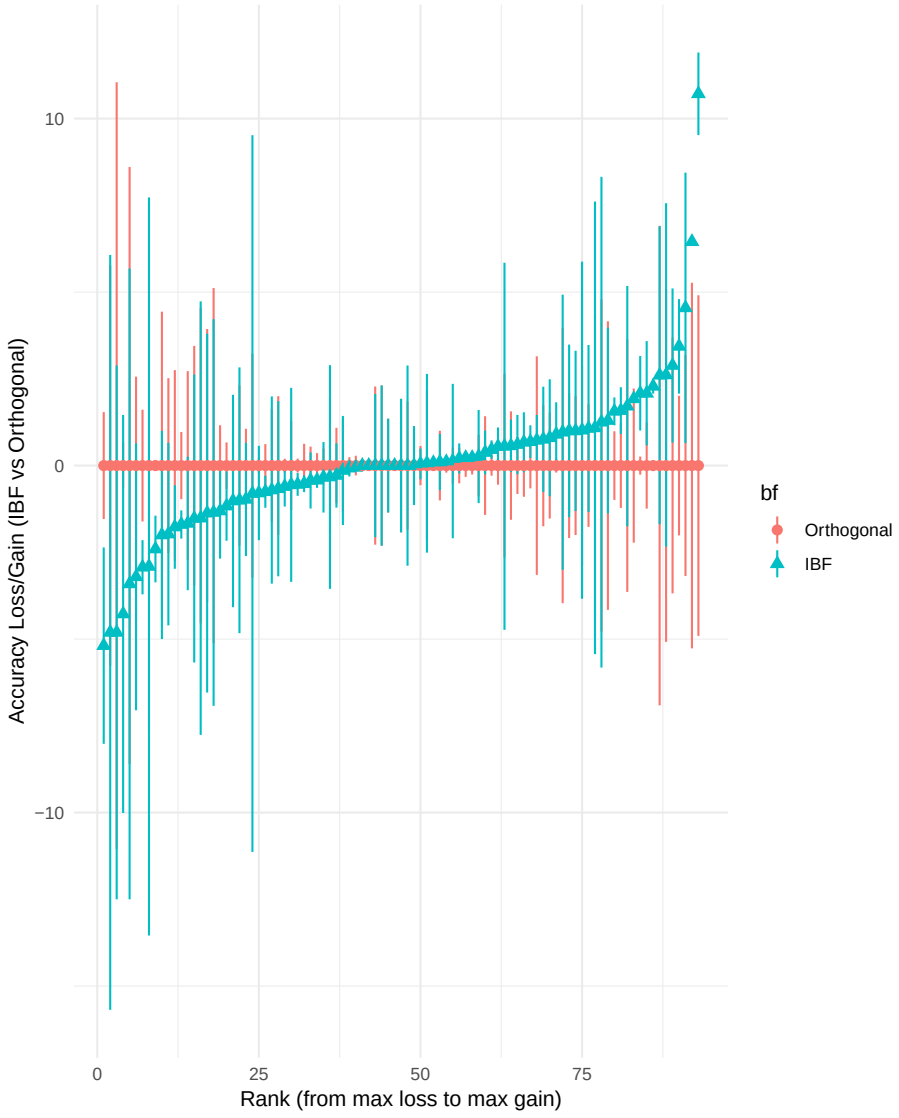


Figure 10: Accuracy comparison for method random forest: orthogonal and IBF implementations

Lastly, Figure 11 shows the loss/gain in accuracy of the evolutionary trees implemented with IBFs with respect to orthogonal partitions. As before, the datasets are ranked from the largest accuracy loss to the largest accuracy gain attained by the models with IBFs. As seen in the figure, the implementation with IBFs also leads to accuracy gains, in this case in 42 out of 93 datasets , just below 50% of the time. The maximum accuracy gain is 18.48% whereas the maximum loss is -25%.

To further disentangle the performance of the IBFs in DTs in datasets with diverse characteristics, we estimate two models using the results of the benchmarking experiments.

In these models the independent variable is a proportion, namely the number of times that the algorithm correctly predicts an observation in the test dataset, relative to the number of observations. Model 1 is estimated using the full set of results (i.e., the results from the 4-fold cross-validation), whereas Model 2 is estimated using the accuracy of the best performer within each set of 4-folds. The appropriate modeling approach for proportions is the logistic model. For these models, we use as explanatory variables the attributes of the datasets, to wit, the number of observations ($n$), the number of features ($f$), the number of classes in the dependent variable ($k$), and the proportion of the majority class ($m$), as well as their squares in order to capture non-linearities. Furthermore, we use an indicator variable ($method$) for the algorithms (i.e. tree, random forest, evolutionary tree), and whether the basis function ($bf$) was orthogonal or IBF. In addition we interact the variable for the basis function ($bf$) with other variables. The results of the models appear in Table 4. As seen there, most coefficients are significant at conventional levels of significance. The coefficients for IBF are significant and positive, which indicates that the use of IBFs increases the accuracy of the algorithms, however the net effect needs to account for the variable interactions. For this reason it is best to simulate the probability of a success to more fully understand how IBFs impact predictive accuracy.

In Figures 12, 13, 14, 15, the probabilities of correctly predicting an observation are plotted with respect to the characteristics of the datasets. The probabilities are predicted within the first decile and and ninth decile of the corresponding variable, to avoid making predictions in regions were observations are sparse (for instance, as seen in Table 1, there is only one dataset with $n > 40,000$). As well, the remaining variables are set to their median values. In this way, when predicting the probability as a function of $n$, $f$, $k$, and $m$ are set to their in-sample median values. The plots show the estimated probabilities with their respective 95% confidence intervals.

With respect to the size of the dataset ($n$), the general trend indicated by Model 1 is that accuracy tends to decline for bigger datasets (see Figure 12). The use of IBFs is a clear improvement over DTs with orthogonal partitions, however, this is not the case for evolutionary trees, an algorithm that performs significantly better with orthogonal partitions. In the case of random forests, there is a a small significant difference between orthogonal partitions and IBFs, with orthogonal partitions leading to better accuracy for smaller datasets only ($n < 1,500$). These results are replicated by Model 2, but with stronger gains

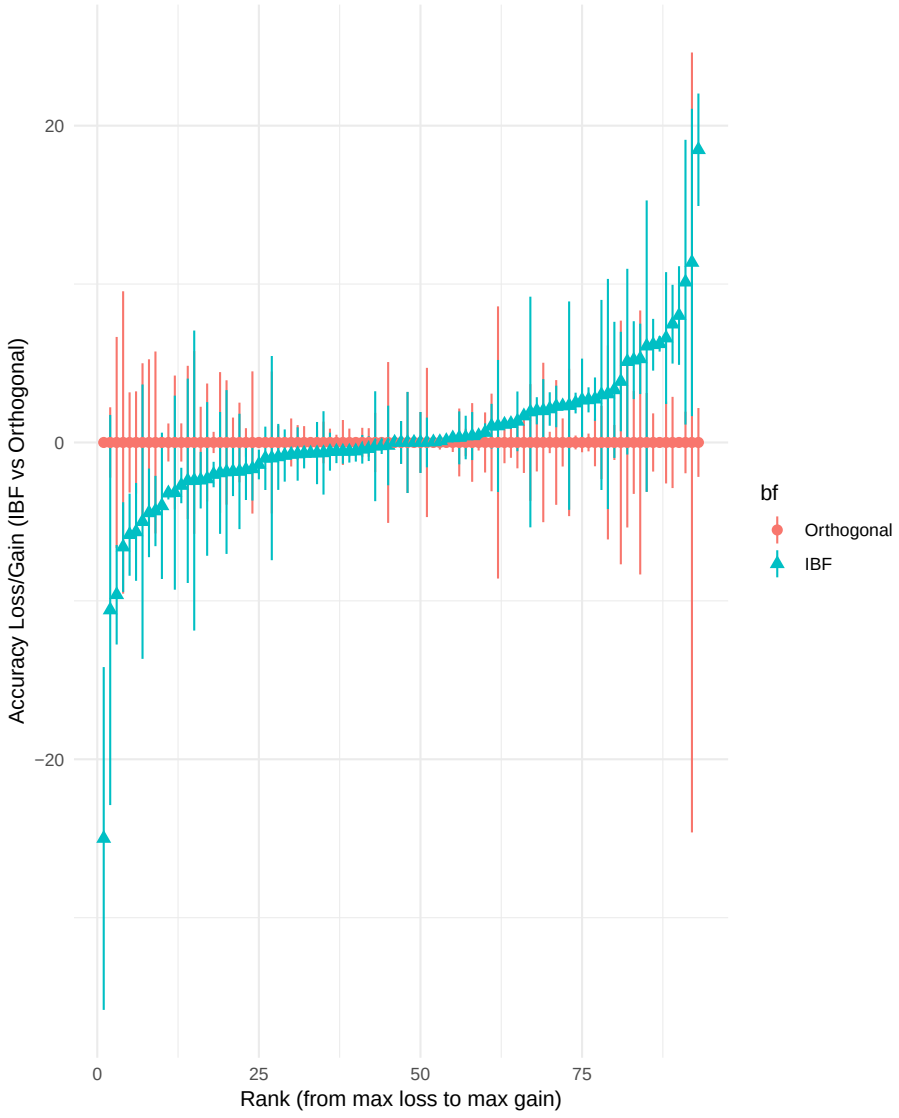


Figure 11: Accuracy comparison for method evolutionary tree: orthogonal and IBF implementations

Table 4: Accuracy: Results of logistic model (dependent variable: proportion of correct predictions)

| | Model 1 | | Model 2 | |
|---|---|---|---|---|
| | Estimate | p-value | Estimate | p-value |
| Intercept | 0.620609 | 0.000000 | 1.158850 | 0.000000 |
| n | -0.000102 | 0.000000 | -0.000110 | 0.000000 |
| n^2 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| f | 0.030822 | 0.000000 | 0.031200 | 0.000000 |
| f^2 | -0.000256 | 0.000000 | -0.000255 | 0.000000 |
| k | -0.159074 | 0.000000 | -0.202363 | 0.000000 |
| k^2 | 0.006816 | 0.000000 | 0.009204 | 0.000000 |
| m | 3.714093 | 0.000000 | 3.182831 | 0.000000 |
| m^2 | -3.733085 | 0.000000 | -3.358070 | 0.000000 |
| Forest | 0.637537 | 0.000000 | 0.536046 | 0.000000 |
| Evolutionary Tree | 0.155661 | 0.000000 | -0.054712 | 0.000000 |
| IBF | 1.048966 | 0.000000 | 0.374667 | 0.000000 |
| n:IBF | 0.000010 | 0.000000 | 0.000021 | 0.000000 |
| n^2:IBF | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| f:IBF | 0.002506 | 0.000056 | 0.006843 | 0.000000 |
| f^2:IBF | -0.000030 | 0.000376 | -0.000048 | 0.006899 |
| k:IBF | -0.141183 | 0.000000 | -0.082562 | 0.000000 |
| k^2:IBF | 0.005269 | 0.000000 | 0.002569 | 0.000000 |
| m:IBF | -1.901150 | 0.000000 | -0.316850 | 0.037922 |
| m^2:IBF | 1.329481 | 0.000000 | -0.050558 | 0.686334 |
| Forest:IBF | -0.097848 | 0.000000 | -0.014762 | 0.237508 |
| Evolutionary Tree:IBF | -0.123462 | 0.000000 | 0.087647 | 0.000000 |

*Note:*
AIC : Model 1 = 392644.965 ; Model 2 = 81034.401
BIC : Model 1 = 392770.599 ; Model 2 = 81129.537
Log Likelihood : Model 1 = -196300.483 ; Model 2 = -40495.2
Deviance : Model 1 = 379653.338 ; Model 2 = 77861.835
Num. obs. : Model 1 = 2232 : Model 2 = 558

for IBFs.

Figure 13 shows the results for the number of features $f$. As seen there, overall accuracy (Model 1) tends to improve as the number of features increases, irrespective of the method or basis function. Similar to the size of the sample, using IBFs uniformly tends to improve accuracy for the algorithm tree. Evolutionary trees are significantly more accurate when using orthogonal partitions. And orthogonal partitions in random forests work slightly better than IBFs, but only when the number of features is relatively small, approximately less than 15, after which point the difference is no longer significant. This is similar in the case of Model 2, again, with a further favorable edge in the case of IBFs.

The third dimension that we investigate is the number of classes. As shown in Figure 14, there is a general tendency for accuracy to deteriorate as the number of classes $k$ incrases. However, despite starting at higher levels of accuracy with respect to orthogonal partitions, this happens much more rapidly when IBFs are used. In this case, IBFs are associated with better or similar accuracy when $k \leq 3$ but have significantly lower levels of accuracy when the dependent variables has more classes. The speed at which the performance of IBFs declines is somewhat slower for the best performing cases (Model 2).

Lastly, with respect to the proportion of the majority class $m$, we can see in Figure 15 that the probability tends to behave in a non-linear fashion, with accuracy being higher when one class has a majority but does not dominate. In this case, IBFs tend to perform significantly better than non-linear partitions. The superiority of IBFs is uniform for the algorithm tree, whereas for evolutionary tree and random forest, orthogonal partitions are significantly more accurate than IBFs. When looking at the best performers (Model 2), the superiority of IBFs is more uniform.

It is important to keep in mind when interpreting these results, that each plot was estimated *while setting three dimensions to their median values*. For this reason, one must be cautious when trying to generalize. The results, nonetheless, are illustrative of potentially the range of different conditions in which the use of IBFs might be desirable.

*5.2 Tree Size*

The last part of our benchmarking experiment is related to the size of trees. Since random forests are ensembles of trees, they are excluded from this. As hinted at by Table 2, the size of the trees did not change much when using IBFs. To further investigate this, we estimate a linear regression model, specified in a similar fashion to the model in the preceding section, with tree size as the dependent variable. The results of this exercise are shown in Table 5. As seen there, the algorithms in general tend to generate more complex trees (more terminal nodes) when the number of observations is large, and there are non-linear relationships with respect to number of classes $k$, and proportion of majority class $m$. The only other noteworthy result is that evolutionary tree tends to produce less complex trees (on average with 2.5 less terminal nodes) than the tree algorithm. Since no other coefficients are significant, we conclude that there is no increase in model complexity when using IBFs.

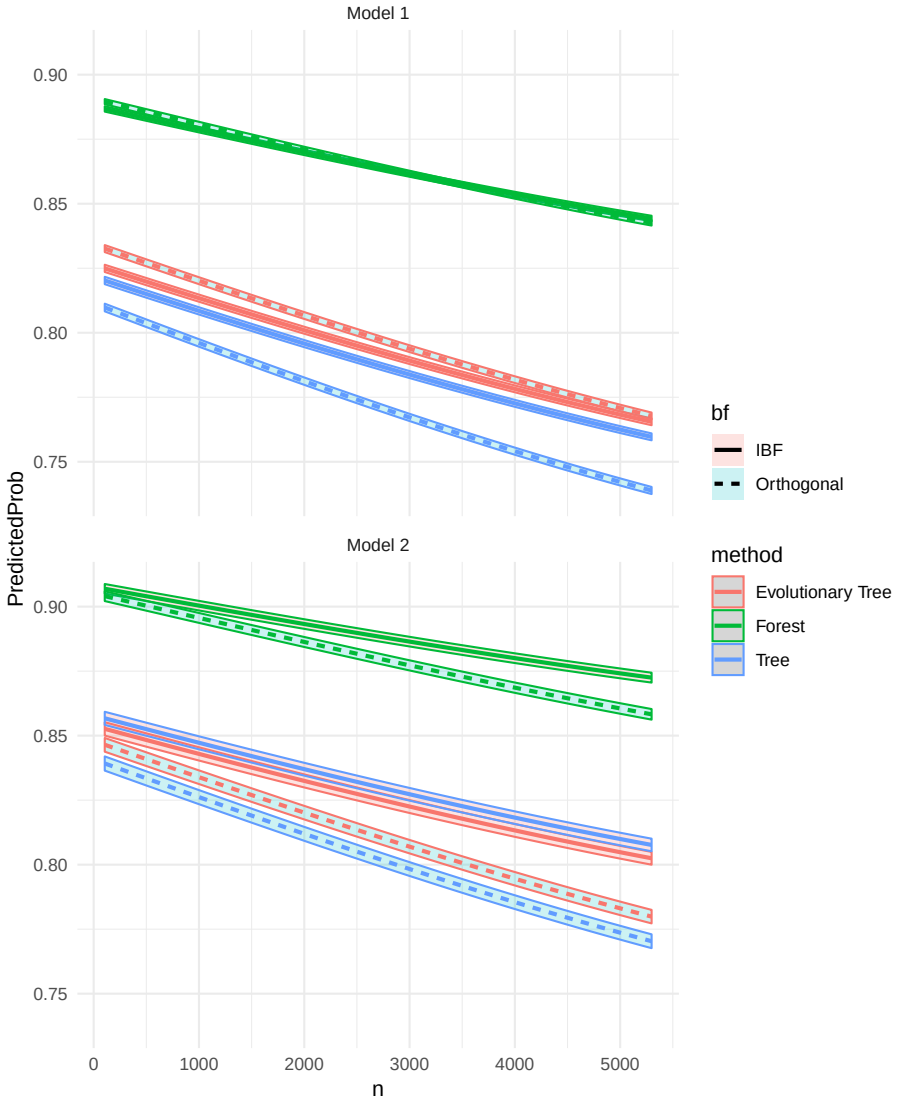


Figure 12: Probability of correctly predicting an observations as a function of number of observations (n)

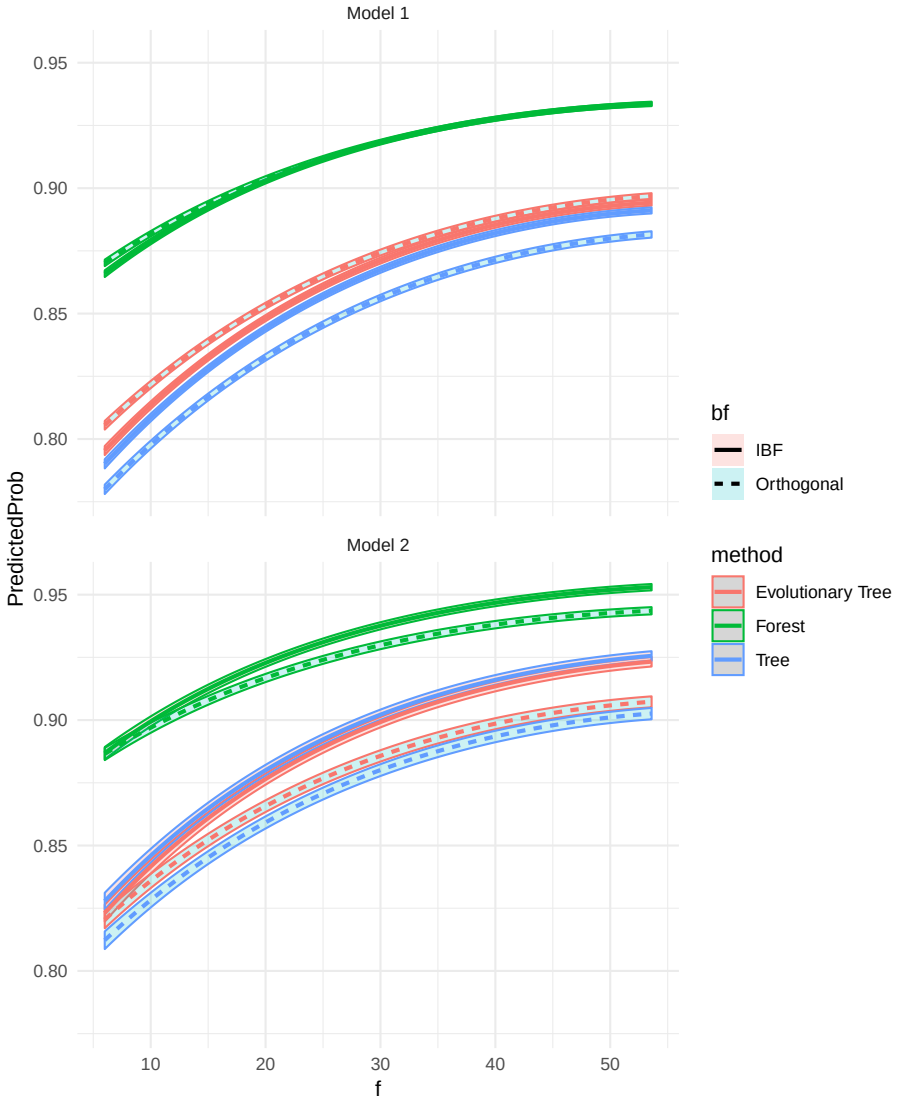


Figure 13: Probability of correctly predicting an observations as a function of number of features (f)

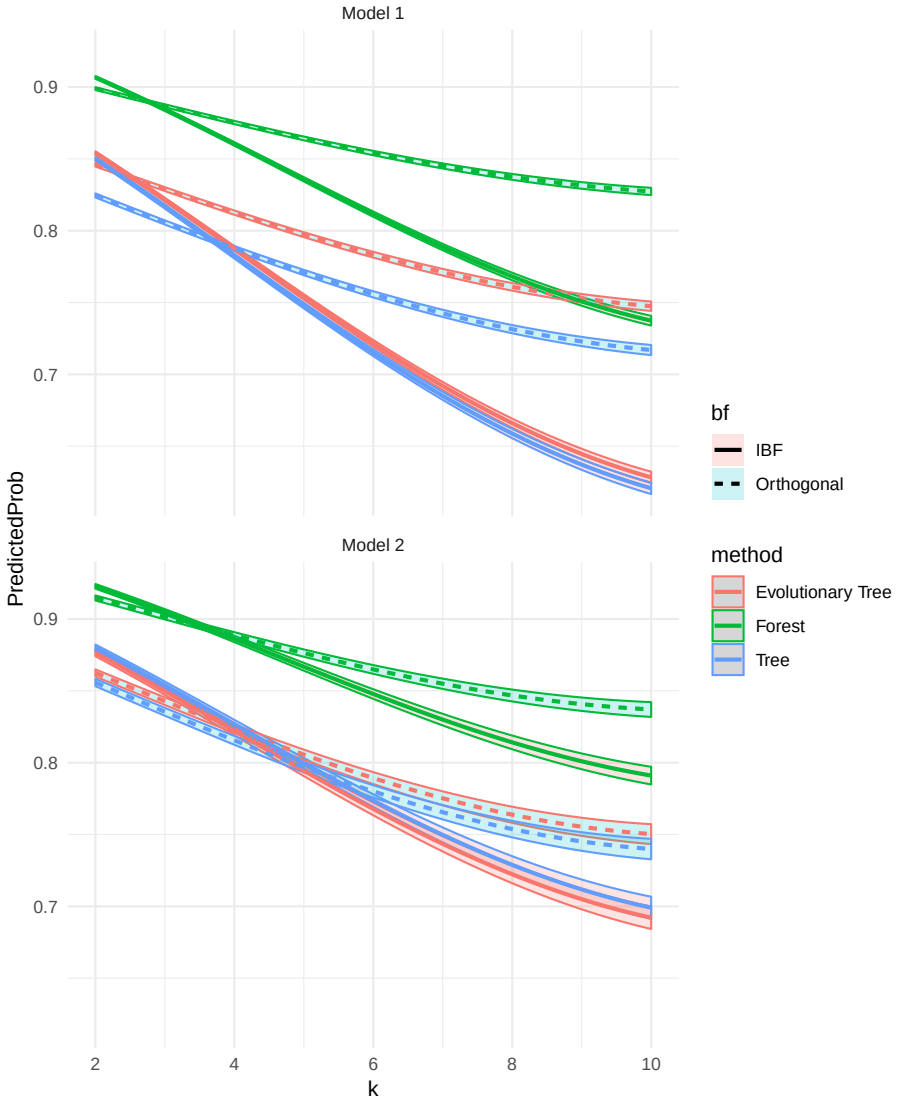


Figure 14: Probability of correctly predicting an observations as a function of number of classes (k)

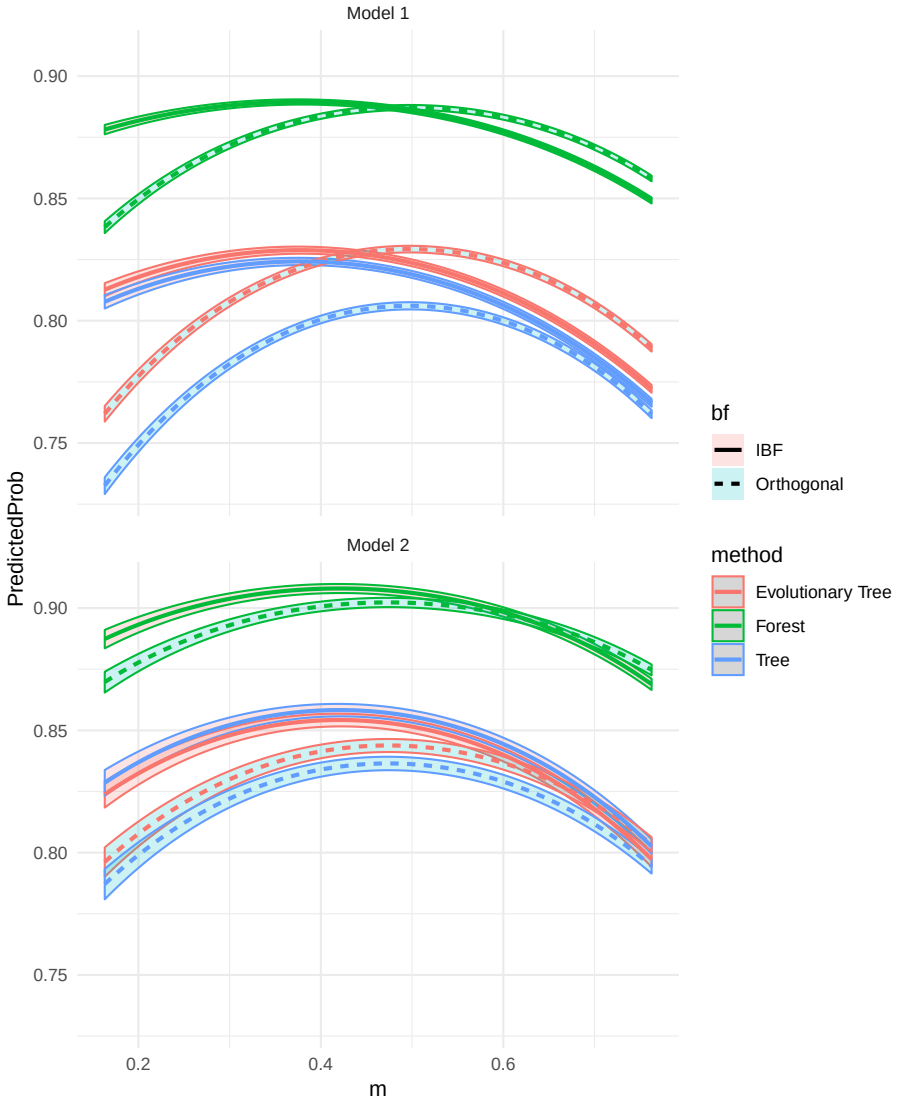


Figure 15: Probability of correctly predicting an observations as a function of proportion majority class (m)

Table 5: Tree size: Results of linear regression model (dependent variable: tree size)

| | Estimate | p.value |
|---|---|---|
| Intercept | 6.835511 | 0.029993 |
| n | 0.000449 | 0.002481 |
| n^2 | 0.000000 | 0.010134 |
| f | 0.042594 | 0.447129 |
| f^2 | -0.001065 | 0.167214 |
| k | 0.692758 | 0.025402 |
| k^2 | -0.029588 | 0.011604 |
| m | 16.064753 | 0.062771 |
| m^2 | -17.663186 | 0.016387 |
| Evolutionary Tree | -2.513441 | 0.000033 |
| IBF | 3.587083 | 0.420331 |
| n:IBF | -0.000123 | 0.556426 |
| n^2:IBF | 0.000000 | 0.682911 |
| f:IBF | -0.106129 | 0.180536 |
| f^2:IBF | 0.001254 | 0.250141 |
| k:IBF | -0.073707 | 0.866342 |
| k^2:IBF | 0.001511 | 0.927309 |
| m:IBF | -6.795145 | 0.577623 |
| m^2:IBF | 6.833829 | 0.511032 |
| Evolutionary Tree:IBF | -0.615591 | 0.470897 |

*Note:*
AIC = 392644.965
BIC = 392770.599
Log Likelihood = -196300.483
Deviance = 379653.338
Num. obs. = 2232

## 6 Sample Applications

As illustrated in the benchmarking exercise above, the modelling strategy proposed here works well on multifeature datasets. Furthermore, inducing oblique and/or non-linear partitions can improve the performance of DTs. In this section, we complement the results of the benchmarking experiment by presenting three empirical examples. One of these examples is a classification problem, and two examples are regressions on quantitative variables. Two of the examples are of geographical interest and have the advantage that the features are geospatial, which greatly facilitates the visualization of the results. The third example is a multifeature set. In this case, we solve the issue of visualizing non-oblique partitions by means of a device that we call *decision charts*.

### *6.1 Classification Example: Ethnic Neighborhoods*

The first example that we present is concerned with a spatial classification problem. This kind of problem is often found in geography, public health, and sociology, among other disciplines, and addresses the objective of defining spatial neighborhoods. Examples of this kind of research in the literature include include Gaudart et al. (2005), a team of reserchers who developed oblique DTs to identify high risk clusters of malaria. Research by Folch and Spielmann (2014) led to an algorithm to identify regions with flexible constraints. Wong and Huang (2017) identify geographical spheres of influence based on social media data. One more example is the research by Logan et al. (2011) that aimed at identifying ethnic neighborhoods using historical datasets.

The example presented here is similar in spirit to the research of Logan et al. (2011), and makes use of a portion of the same historical dataset (see John R. Logan et al. 2011). The dataset consists of 21520 individual records for part of Newark, coded by ethnicity according to the 1880 US Census. Of these, 7,659 records are classified as White Americans, 4,411 are classified as Irish, and 9,450 are classified as German. The geographical distribution of these groups in the region of Newark under study is shown in Figure 16.

Since the objective of the example is to find homogeneous neighborhoods, this examples considers only two features, namely the coordinates of the observations in longitude and latitude. The following IBFs are introduced as additional features:

$$
\begin{aligned}
h_1(b_1(long), b_1(lat)) &= b_1(long) + b_1(lat) = long + lat \\
h_2(b_1(long), b_1(lat)) &= b_1(long) b_1(lat) = long \times lat \\
h_1(b_2(long), b_2(lat)) &= b_2(long) + b_2(lat) = long^2 + lat^2 \\
h_1(b_3(long), b_3(lat)) &= b_3(long) + b_3(lat) = e^{long} + e^{lat} \\
h_2(b_3(long) b_3(lat)) &= b_3(long) b_3(lat) = e^{long} \times e^{lat}
\end{aligned}
$$

After some experimentation, the coordinates of the observations were centered at the top right corner of the set of points, and scaled to the unit on the longest extent of the dataset. For comparison purposes, we begin by training a DT for this problem using conventional orthogonal partitions. The results of this

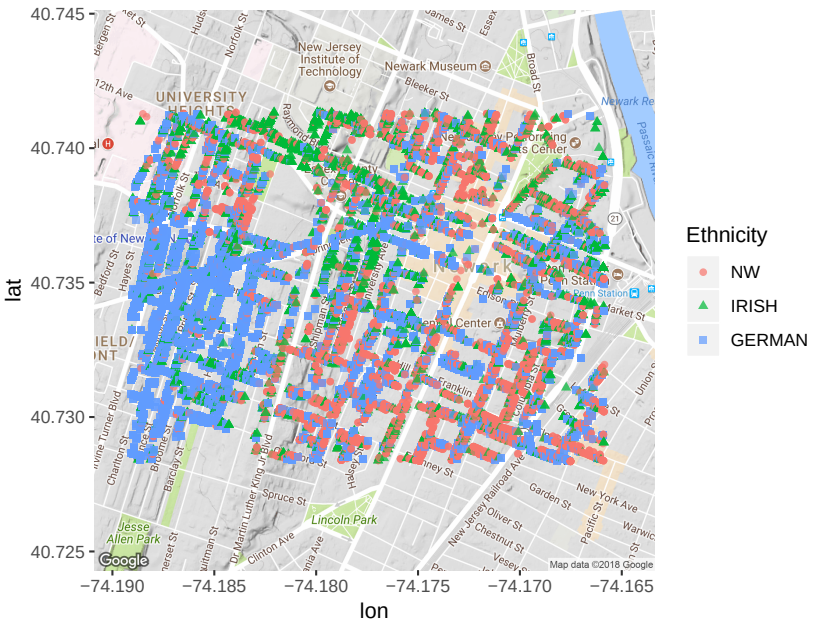


Figure 16: Three ethnic groups in Newark (1880 US Census)

model are shown in Figure 17, a model with six terminal nodes that identify two predominantly German neighborhoods, three predominantly White American neighborhoods, and one predominantly Irish neighborhood. The model, after examining the interior branches, did not require pruning.

This model has an in-sample error rate of 0.39 and a cross-validation error rate of 0.39. The population of two of the White American Neighborhoods is over 60% of that ethnic group. Another neighborhood has a plurality (49%) of White Americans and a more or less even distribution of Irish (22%) and Germans (29%). One neighborhood is strongly Irish (62% of population) with White American and German minorities (27% and 11% respectively). Another neighborhood is strongly German, with over 70% of the residents of that ethnicity, whereas one more has a plurality of Germans (41%) but is otherwise quite mixed. The partitions of the DT in effect delimit the ethnic neighborhoods, as seen Figure 18.

Next, the same dataset is used to train a tree with IBFs as listed above. The model was examined to determine that pruning was not required. The results of this model are shown in Figure 19. The model also has six terminal nodes which identify two German neighborhoods, three White American neigborhoods, and one Irish neighborhoods. Note that two of the partitions are orthogonal (i.e., $lat < -0.37$ and $lat \geq -0.16$), two partitions are linear but oblique (i.e., $long + lat \geq 0.51$ and $long + lat \geq 0.21$), and one partition is non-linear (i.e.,

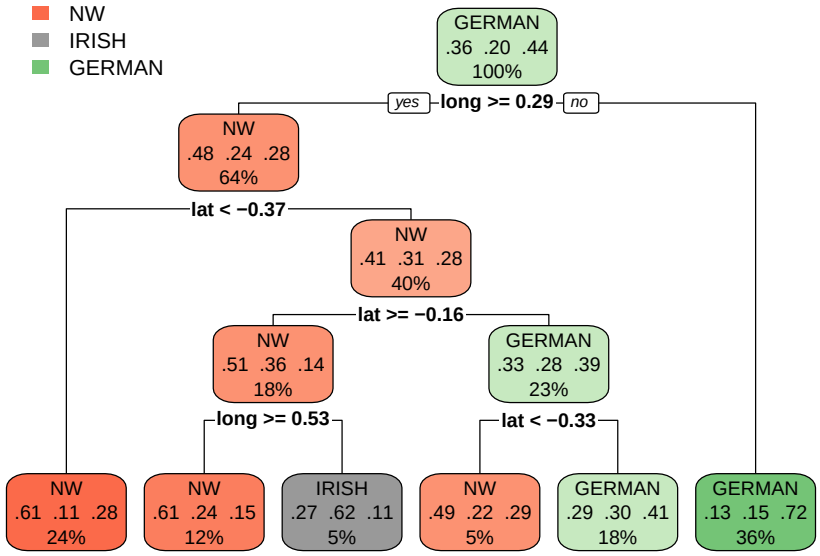


Figure 17: Decision tree with orthogonal partitions for spatial classification, Newark ethnic groups

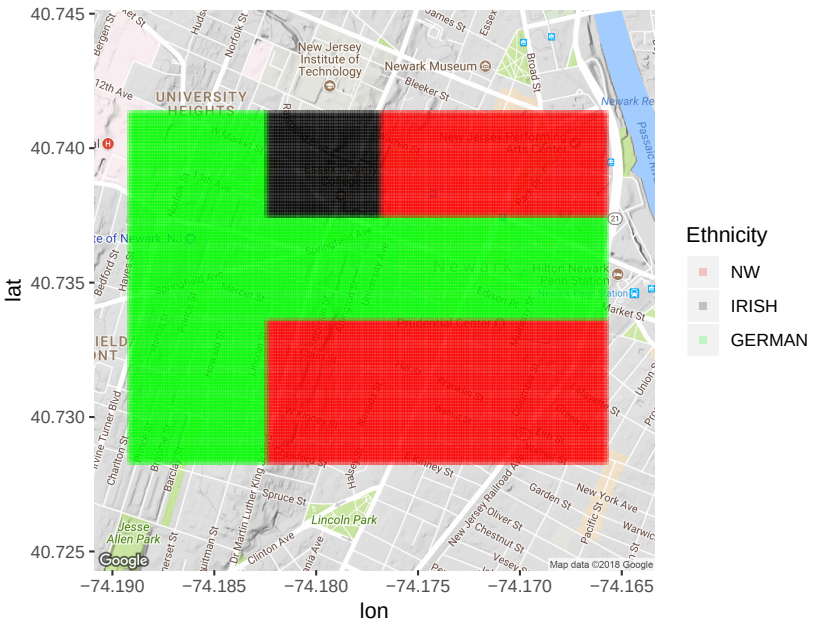


Figure 18: Ethnic neighborhoods in Newark, 1880, using orthogonal partitions

$e^{long} + e^{lat} \geq 2.0$). The ethnic neighborhoods are shown in Figure 20. This model has an in-sample error rate of 0.39 and a cross-validation error rate of 0.39, which is comparable to the orthogonal model. In this case, the analyst must decide whether the neighborhoods identified by the DT with IBFs are a more realistic representation of a spatial process.

### *6.2 Regression Example: Spatial Market Segmentation*

The case presented next is similar to the spatial classification problem above, except that it is now a regression situation. The example relates to an issue widely discussed in the real estate and property valuation literature, and is concerned with the identification of spatial submarkets. Numerous papers exist on this topic, including Gabriel (1984), Feitelson (1993), Paez et al. (2001), Bourassa et al. (2003), Helbich et al. (2013), and Wheeler et al. (2014). More recently, DTs have been applied to identify spatial market segments in a real estate setting by Fuss and Koller (2016), however using orthogonal partitions.

One obvious limitation of using orthogonal partitions in the case of real estate spatial submarkets is that the processes that drive land rent (and therefore property values) tend to be radial and are possibly anisotropic due to inhomogeneities in the landscape. The canonical urban economic theory for monocentric cities (which has since been expanded to polycentric cities) states that land rent decays

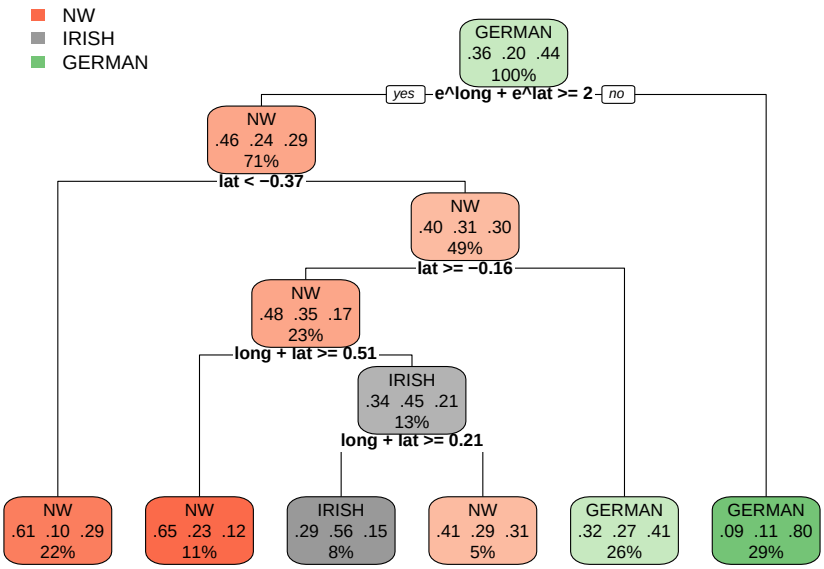


Figure 19: Decision tree with non-orthogonal/non-linear partitions for spatial classification, Newark ethnic groups

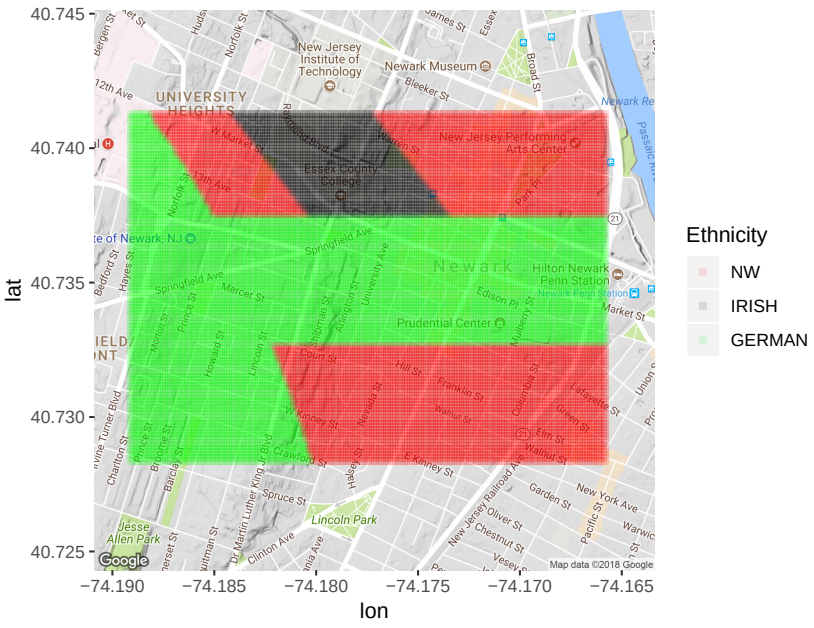


Figure 20: Ethnic neighborhoods in Newark, 1880, using non-orthogonal partitions

from business districts (Alonso 1964), a prediction that is borne by numerous empirical studies.

The illustration presented here deals with land prices in Sapporo, Japan. The dataset consists of 429 observations of geocoded land prices (in ¥/$m^2$). Informed by the literature on property valuation, the coordinates (in longitude and latitude) were centered on the geometric mean of the locations of the observations, a location that coincides with the central business district of the city. Further, the coordinates were scaled to the unit on the longest extent of the dataset. Land prices were log-transformed to mitigate their lack of normality. Figure 21 shows the locations of the observations and prices. As can be seen there, Sapporo is a typical monocentric city, with the highest prices concentrated in and around the central business district.

As before, we train a DT using orthogonal partitions. The results of this model are shown in Figure 22, where it can be seen that the tree has thirteen terminal nodes, or equivalently spatial submarkets. This was the best-performing tree size, and there was no need to prune. The $pseudo - R^2$ of this model is 0.71. The submarkets are shown in Figure 23, where it can be seen that the submarkets manage to capture the monocentric structure of land prices in Sapporo, albeit in cubist style.

The next step was to train a DT, but this time using IBFs as additional features in the dataset. The basis functions used are the same five IBFs used

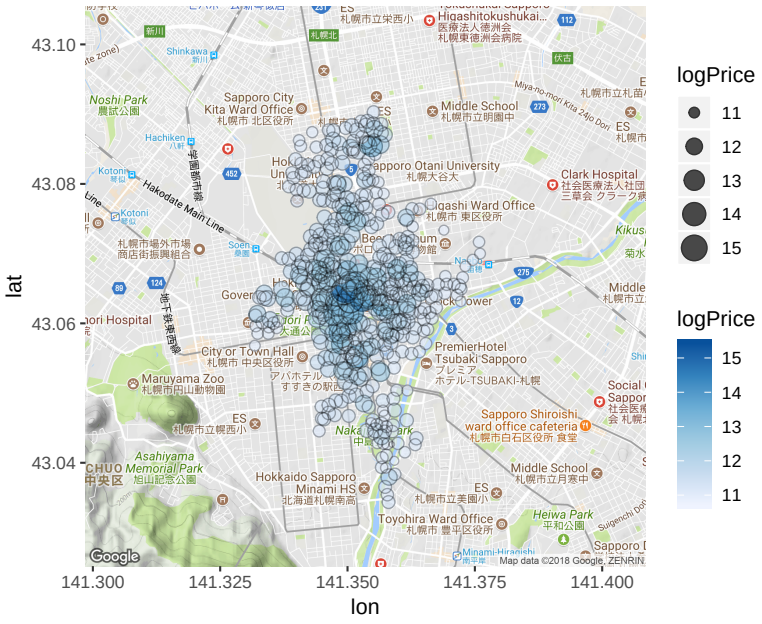


Figure 21: Land prices in Sapporo (log)

in the preceding example. The resulting DT was checked to see if pruning was appropriate, but the full tree gave the best results (see Figure 24). It is worthwhile noting that only one of seven partitions is orthogonal. The remaining six partitions are all non-linear. The number of terminal nodes/submarkets is eight compared to thirteen in the model with orthogonal partitions. Both models have a comparable performace, with a $pseudo - R^2 = 0.71$, however the use of IBFs results in a more parsimonious model. Furthermore, the resulting market segments (see Figure 25) are much more appealing, and conform better to our theoretical understanding of the radial and anisotropic mechanisms of land price determination.

### *6.3 Regression Example: Voter Turnout*

The previous two examples were of DTs trained using only two features (the coordinates of the observations) and augmented by means of IBFs. Both examples dealt with forms of spatial segmentation which incidentally facilitate the visualization of the non-orthogonal and non-linear partitions that result from the use of IBFs. In the last example the we present, we use a multifeature dataset of voter turnout in a recent provincial election in Ontario, Canada.

Voter turnout is an issue of interest to behavioral and political scientists, who identify this aspect of democracies as one of three key indicators of their performance (Powell 1982). Moreover, voter turnout tends to vary quite substan-

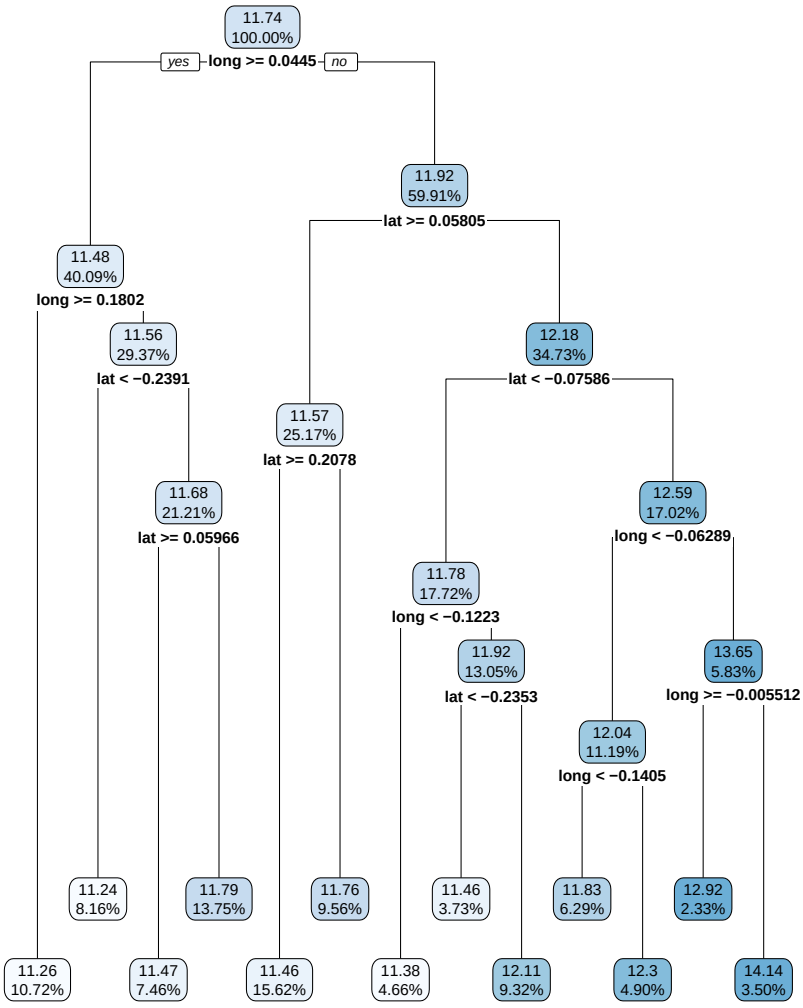


Figure 22: Decision tree with orthogonal partitions for submarket identification, Sapporo land prices

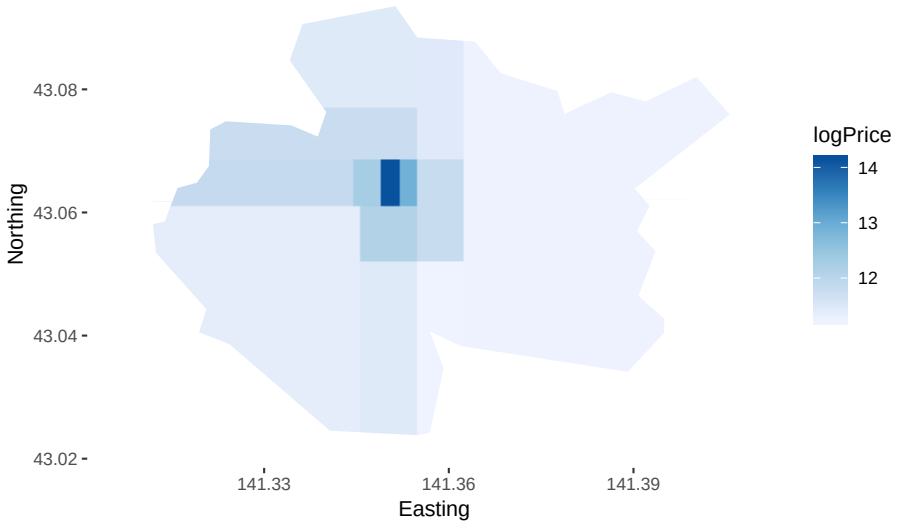


Figure 23: Spatial land price submarkets in Sapporo using orthogonal partitions

tially by region, a fact that has motivated a voluminous literature (starting with the pioneering research of Powell 1982; Powell 1986; and Jackman 1987) that aims at understanding the factors that correlate with voter turnout. Numerous studies exist that empirically test the relationships between voter turnout and a variety of socio-economic, demographic, and other variables (for a relevant review see Geys 2006; and more recently Stockemer 2017).

The case presented in this section is of the 2018 provincial election in Ontario, Canada. Data were obtained from three sources. First, a geography file of Electoral Districts in Ontario was obtained from Elections Ontario (https://www.elections.on.ca/en/voting-in-ontario/electoral-district-shapefiles/limited-use-data-product-licence-agreement/download-shapefiles.html). The effectively final counts of the election were obtained from a political blog and cross-checked with information from Elections Ontario for accuracy (https://quandyfactory.com/blog/201/unofficial_ontario_2 Finally, socio-economic and demographic information was retrieved from the 2016 Canadian Census. Census data were obtained at the level of Census Subdivisions. Electoral Districts are sometimes larger than Census Subdivisions, so census variables were converted to the Electoral Districts by aggregation in the case of absolute values (e.g., population, population by educational achievement), or by calculating their area-weighted averages in the case of rates (e.g., median income).

The dataset consists of 124 records (one for each electoral district), and

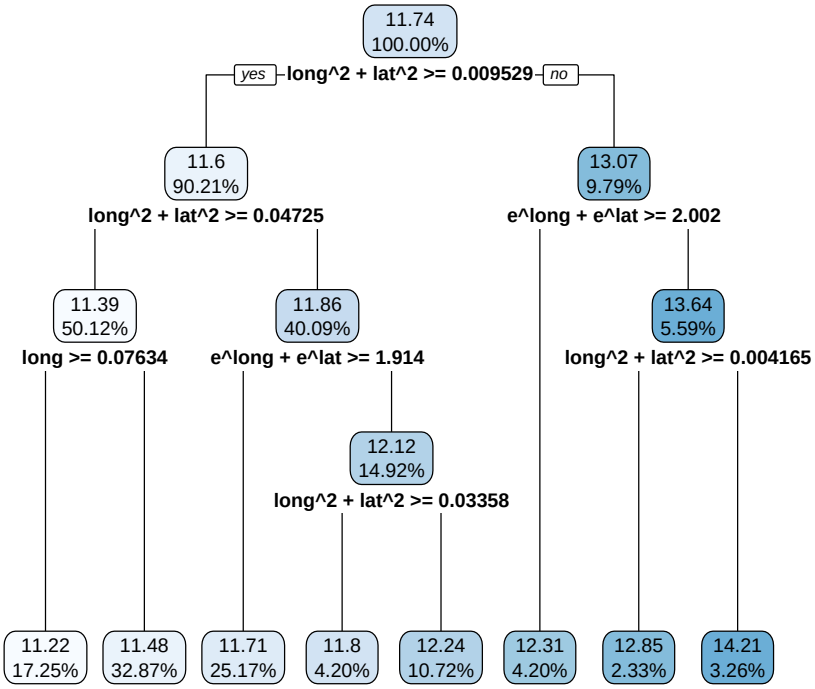


Figure 24: Decision tree with non-orthogonal/non-linear partitions for submarket identification, Sapporo land prices

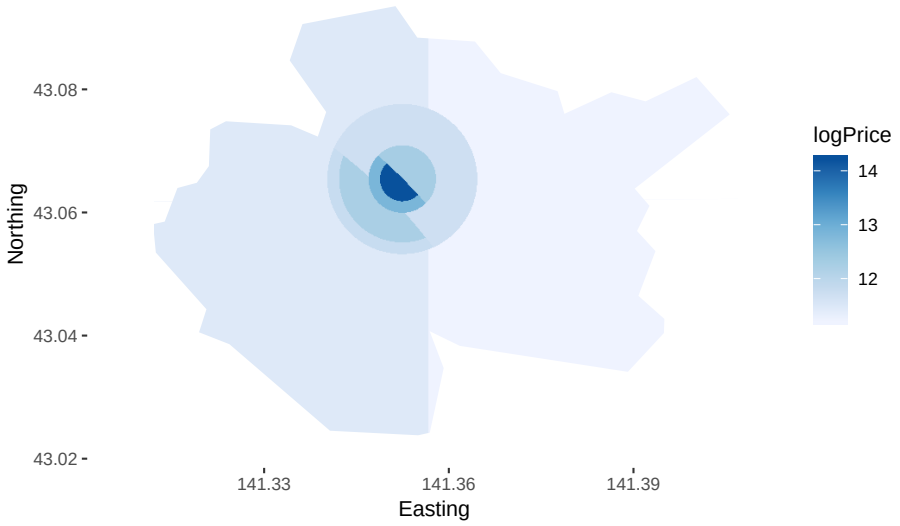


Figure 25: IBF-based non-orthogonal land price regions (log) in Sapporo

seventeen variables, with descriptive statistics as shown in Table 6. Voter turnout is calculated as the proportion of total votes to number of registered electors. The selection of variables reflects an interest in geographical context (e.g., population density and median commute duration), the effect of government policy (% of government transfers relative to income and average income taxes as percent of income), in addition to features relating to age, income, poverty, and academic achievement. These variables were centered on their minimum values and scaled to the unit interval prior to the analysis.

We started by training an initial DT which resulted in a model with eleven terminal nodes. This model was examined, and based on performance was eventually pruned to give the model shown in Figure 26. This model has a $pseudo - R^2$ of 0.24. As seen in the figure, the model is relatively simple, and identifies two covariates that correlate with voter turnout, namely % of population living in low income and population density. The lowest voter turnout rates are associated with Electoral Districts with higher rates of population living in poverty, whereas the highest turnout rates are associated with Electoral Districts characterized by higher population density and low poverty rates.

The next step was to train a model that included IBFs as above. The initial model had eleven terminal nodes, but upon examination was pruned to produce the model shown in Figure 27. The $pseudo - R^2$ of this model is 0.44. Note that three of the partitions in this model are oblique, and one

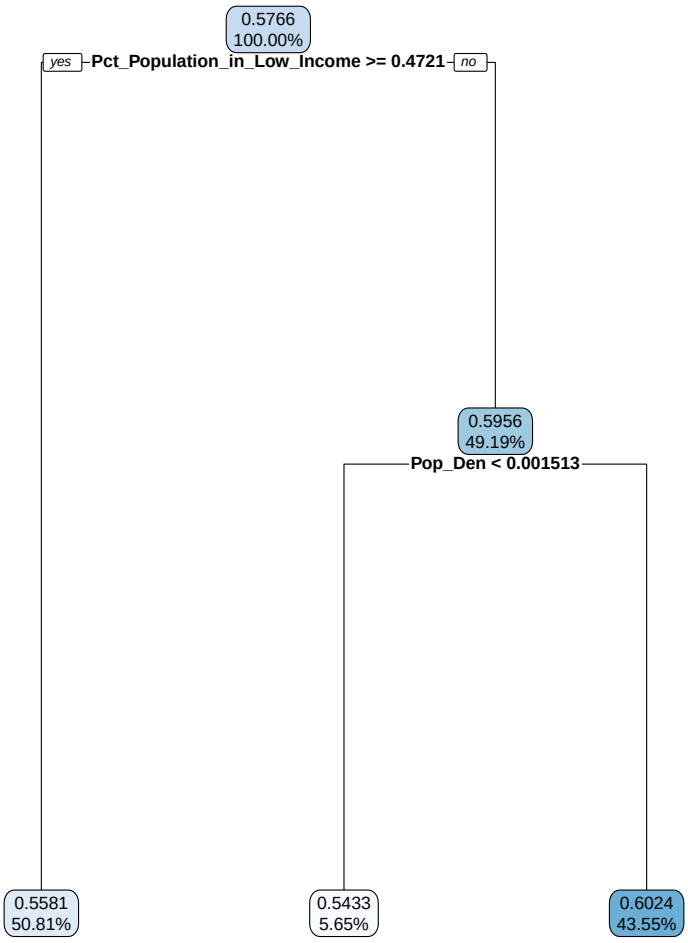


Figure 26: Decision tree with orthogonal partitions, Ontario voter turnout

Table 6: Descriptive statistics for electoral districts in Ontario, 2018

| Variable | Abbv | Min | Max | Mean | std |
|---|---|---|---|---|---|
| Voter_Turnout | Turnout | 0.44 | 0.67 | 0.58 | 0.047 |
| Pop_Den | PD | 0.091 | 4150 | 1349 | 1567 |
| Median_Age | MA | 12 | 51 | 39 | 4.9 |
| Median_Income | MI | 11776 | 40831 | 29901 | 5275 |
| Male_Median_Income | MMI | 12972 | 51442 | 36029 | 6960 |
| Female_Median_Income | FMI | 10478 | 33728 | 25042 | 4193 |
| Pct_Government_Transfer_Payments | PGTP | 6.1 | 23 | 13 | 3.6 |
| Median_Commute_Dur | MCD | 7.3 | 35 | 23 | 6.6 |
| Avg_Inc_Taxes_as_Pct_Inc | AITPI | 3.8 | 24 | 16 | 3.4 |
| Pct_Population_in_Low_Income | PPLI | 1.8 | 24 | 13 | 4.5 |
| Prop_no_certificate | PNC | 0.044 | 0.29 | 0.11 | 0.036 |
| Prop_HS_diploma | PHSD | 0.17 | 0.32 | 0.24 | 0.04 |
| Prop_Post_HS_diploma | PPHSD | 0.49 | 0.79 | 0.64 | 0.067 |
| Prop_trade_certificate | PTC | 0.043 | 0.15 | 0.08 | 0.027 |
| Prop_college_diploma | PCD | 0.17 | 0.33 | 0.24 | 0.045 |
| Prop_university_bachelor | PUB | 0.068 | 0.29 | 0.17 | 0.065 |
| Prop_postgraduate | PPG | 0.029 | 0.19 | 0.11 | 0.048 |

is non-linear. Population density (PD), which appeared in the model with orthogonal partitions, is still part of this model in the last interior node. However, percentage of population living in poverty is not. Instead, average income taxes as percentage of income (AITPI), median age (MA), and several academic achievement variables entered the model (proportion of trade certificates: PTC; proportion of college diploma: PCD; proportion of high school diploma: PHSD; proportion of university bachelor: PUB; and propotion with no certificate: PNC). These variables are often correlated with population living in poverty, but give a more refined view of the characteristics of populations that associate with voter turnout.

In the case of a multifeature dataset, interpretation of a DT with IBFs is not as straightforward as it was for orthogonal partitions, particularly considering that the variables were centered and scaled. In order to enhance the interpretability of the results of models with IBFs, we propose to use a device we call decision charts. These charts allow us to plot the decision boundaries in the space of the two basis implied by each node of the tree. By recentering and rescaling the decision boundaries back to the units of the original basis vectors, a simple protocol can be devised to interpret the results of a model. The maps in the previous two examples showing ethnic boundaries and market segments are essentially decision charts for the case of two features. An example of a set of decision charts for the multifeature voter turnout model is shown in Figure 28.

As seen in the figure, low voter turnouts are associated with Electoral Districts where income taxes as percentage of income are on average low, and

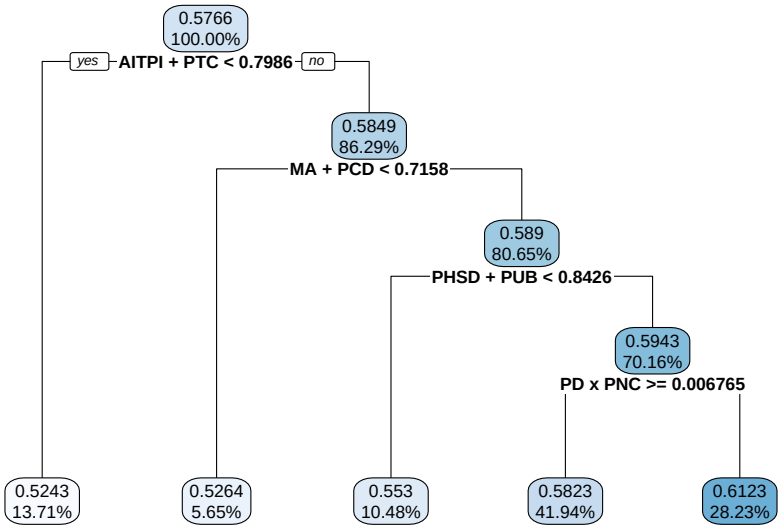


Figure 27: Decision tree with non-orthogonal/non-linear partitions, Ontario voter turnout

that *simultaneously* have relatively high proportions of residents with trade certificates (an indication of the presence of blue collar workers). High voter turnout rates, in contrast, are associated with Electoral Districts that tend to be older and better educated (tops of Charts 2 and 3), and with high population density *or* low population density and a high proportion of people without certificate (bottom of Chart 4).

## 7 Conclusions and Directions for Future Research

Decision Trees are a popular data analysis technique used in a wide range of applications. The somewhat restrictive nature of binary orthogonal partitions has been recognized in the past, and a number of methods have been proposed in the literature that allow for oblique partitions. In this paper, we proposed a new strategy to induce oblique and non-linear partitions for DTs. This is achieved by introducing Interactive Basis Functions. The use of IBFs is attractive because it can induce partitions of different shapes at a relatively low computational cost. Furthermore, the underlying recursive-partition algorithm is not changed, which means that IBFs can be used with any implementation of DTs in existing software.

An extensive benchmarking exercise using 93 publicly available classification datasets shows that the use of IBFs can improve the accuracy of the model without

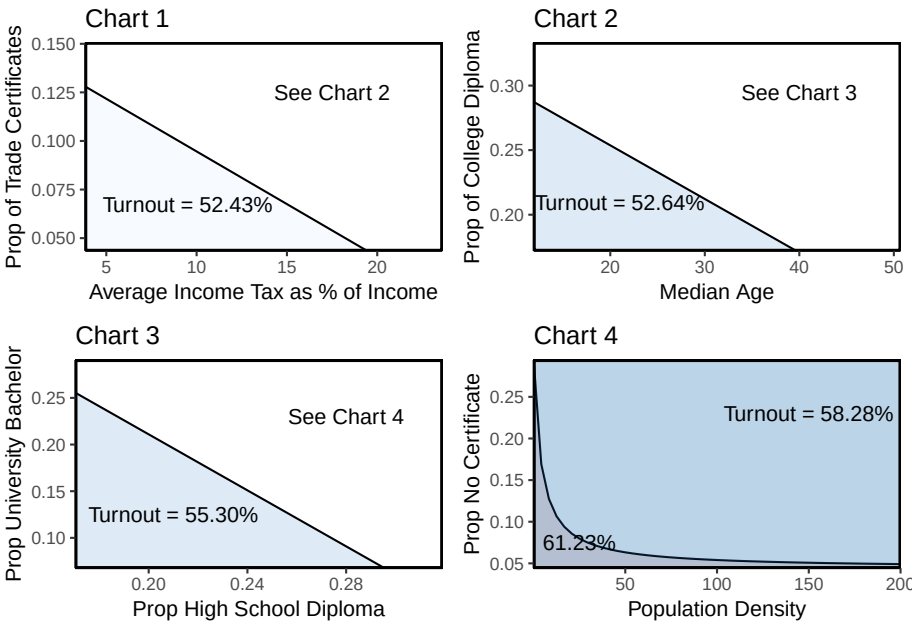


Figure 28: Decision charts for voter turnout

compromising model parsimony, at only very modest increases in computation time. Analysis of the results of the benchmarking experiments suggests that IBFs are a superior alternative to conventional trees in a wide range of situations, and may also improve the performance of evolutionary trees and random forests. IBFs in particular seem to perform better for 1) larger datasets; 2) with a large number of features $f$; 3) when the number of classes $k$ is small; and 4) the target variable is not dominated by a single class, in other words when the proportion of the majority class $m$ does not exceed 40-50%.

In addition to the benchmarking experiments, three case studies helped to illustrate the application and potential advantages of IBFs. In one spatial classification example, the accuracy of the model was maintained but potentially more appealing boundaries were identified. In a market segmentation example, likewise, accuracy was maintained but a more parsimonious and theoretically consistent model was obtained. And in a multifeature model of voter turnout, IBFs led to a better model fit and a more parsimonious model.

One downside of using oblique/non-linear partitions in DTs is that the interpretability of the typical visual output (as a tree with binary branches) becomes compromised. To remedy this situation we introduced a new tool called decision charts, a set of visual devices where new inputs can be located to help an analyst to reach a decision using the underlying DT.

As we noted in the paper, the additive IBF is equivalent to previous efforts

to induce oblique partitions, with a key simplification: the assumption that the function is non-parametric. Obviously, parameterizing an IBF could increase its flexibility, however at a higher computational cost. Whether this higher computational cost exceeds that of, say, the CART-LC method, is an open question. A research challenge would be to parameterize other IBFs for increasingly flexible non-linear partitions. Finally, a limitation of the development presented in this paper is that it only applies to quantitatie features, and therefore it would be interesting to explore possible extensions for qualitative features. This is also a matter for future research.

## Acknowledgments

A. Paez would like to thank Dr. Tony Lea, Mr. Sean Howard, Mr. Peter Miron, and other participants in the Environics Analytics Methodological Seminars, for useful discussions in the early stages of this research. F.A. Lopez, M. Ruiz and M. Camacho acknowledge the financial support from projects ECO2015-65758-P, ECO2015-65637-P, and ECO2016-76178-P respectively. This study is a result of activity carried out under the program Groups of Excellence of the Region of Murcia, the Fundacion Seneca, Science and Technology Agency of the region of Murcia project 19884/GERM/15. The following R packages were used during this research, and the authors wish to acknowledge their developers: `tidyverse` (Wickham 2017), `plyr` (Wickham 2011), `readxl` (Wickham and Bryan 2018), `ggthemes` (Arnold 2018), `tree` (Ripley 2018), `evtree` (Grubinger, Zeileis, and Pfeiffer 2014), `rpart` (Therneau, Atkinson, and Ripley 2017), `rpart.plot` (Milborrow 2018), `ggmap` (Kahle and Wickham 2013), `readr` (Wickham, Hester, and Francois 2017), `rgdal` (Bivand, Keitt, and Rowlingson 2018), `knitr` (Xie 2015; Xie 2018), `kableExtra` (Zhu 2018), `mlbench` (Newman et al. 1998), `dprep` (Acuna and CASTLE research group at The University of Puerto Rico-Mayaguez 2015), `rattle.data` (Williams 2011), `microbenchmark` (Mersmann 2018), `RColorBrewer` (Neuwirth 2014), `scales` (Wickham 2018), and `gridExtra` (Auguie 2017).

## References

Acuna, Edgar, and the CASTLE research group at The University of Puerto Rico-Mayaguez. 2015. *Dprep: Data Pre-Processing and Visualization Functions for Classification*. https://CRAN.R-project.org/package=dprep.

Alonso, William. 1964. *Location and Land Use*. Book. Cambridge: Harvard University Press.

Arnold, Jeffrey B. 2018. *Ggthemes: Extra Themes, Scales and Geoms for 'Ggplot2'*. https://CRAN.R-project.org/package=ggthemes.

Auguie, Baptiste. 2017. *GridExtra: Miscellaneous Functions for "Grid" Graphics*. https://CRAN.R-project.org/package=gridExtra.

Bektas, B. A., A. Carriquiry, and O. Smadi. 2013. "Using Classification Trees for Predicting National Bridge Inventory Condition Ratings." Journal Article.

*Journal of Infrastructure Systems* 19 (4): 425–33. doi:10.1061/(asce)is.1943-555x.0000143.

Bivand, Roger, Tim Keitt, and Barry Rowlingson. 2018. *Rgdal: Bindings for the 'Geospatial' Data Abstraction Library*. https://CRAN.R-project.org/package=rgdal.

Bourassa, S. C., M. Hoesli, and V. S. Peng. 2003. "Do Housing Submarkets Really Matter?" Journal Article. *Journal of Housing Economics* 12 (1): 12–28. ISI:000182413400002 C:/Papers/Journal of Housing Economics/Journal of Housing Economics (2003) 12 (1) 12-28.pdf.

Breiman, L., J. Friedman, C.J. Stone, and R.A. Olshen. 1984. *Classification and Regression Trees*. Book. New York: CRC Press.

Cantu-Paz, E., and C. Kamath. 2003. "Inducing Oblique Decision Trees with Evolutionary Algorithms." Journal Article. *IEEE Transactions on Evolutionary Computation* 7 (1): 54–68. doi:10.1109/tevc.2002.806857.

Choubin, B., H. Darabi, O. Rahmati, F. Sajedi-Hosseini, and B. Klove. 2018. "River Suspended Sediment Modelling Using the Cart Model: A Comparative Study of Machine Learning Techniques." Journal Article. *Science of the Total Environment* 615: 272–81. doi:10.1016/j.scitotenv.2017.09.293.

Dietterich, Thomas G. 2000. "Ensemble Methods in Machine Learning." Conference Proceedings. In *International Workshop on Multiple Classifier Systems*, 1–15. Springer.

Feitelson, E. 1993. "An Hierarchical Approach to the Segmentation of Residential Demand - Theory and Application." Journal Article. *Environment and Planning A* 25 (4): 553–69. ISI:A1993LA98700008.

Fernández-Delgado, Manuel, Eva Cernadas, Senén Barro, and Dinani Amorim. 2014. "Do We Need Hundreds of Classifiers to Solve Real World Classification Problems?" Journal Article. *The Journal of Machine Learning Research* 15 (1): 3133–81.

Folch, D. C., and S. E. Spielman. 2014. "Identifying Regions Based on Flexible User-Defined Constraints." Journal Article. *International Journal of Geographical Information Science* 28 (1): 164–84. doi:10.1080/13658816.2013.848986.

Friedman, Jerome H, Jon Louis Bentley, and Raphael Ari Finkel. 1977. "An Algorithm for Finding Best Matches in Logarithmic Expected Time." Journal Article. *ACM Transactions on Mathematical Software (TOMS)* 3 (3): 209–26.

Füss, R., and J. A. Koller. 2016. "The Role of Spatial and Temporal Structure for Residential Rent Predictions." Journal Article. *International Journal of Forecasting* 32 (4): 1352–68. doi:10.1016/j.jiforecast.2016.06.001.

Gabriel, S. A. 1984. "A Note on Housing-Market Segmentation in an Israeli Development Town." Journal Article. *Urban Studies* 21 (2): 189–94. ISI:A1984SS20300008.

Gaudart, J., N. Graffeo, D. Coulibaly, G. Barbet, S. Rebaudet, N. Dessay, O. K. Doumbo, and R. Giorgi. 2015. "SPODT: An R Package to Perform Spatial Partitioning." Journal Article. *Journal of Statistical Software* 63 (16): 1–23. <Go to ISI>://WOS:000349847100001.

Gaudart, Jean, Belco Poudiougou, Stéphane Ranque, and Ogobara Doumbo. 2005. "Oblique Decision Trees for Spatial Pattern Detection: Optimal Algorithm

and Application to Malaria Risk." Journal Article. *BMC Medical Research Methodology* 5 (1): 22. doi:10.1186/1471-2288-5-22.

Geys, B. 2006. "Explaining Voter Turnout: A Review of Aggregate-Level Research." Journal Article. *Electoral Studies* 25 (4): 637–63. doi:10.1016/j.electstud.2005.09.002.

Ghasri, M., T. H. Rashidi, and S. T. Waller. 2017. "Developing a Disaggregate Travel Demand System of Models Using Data Mining Techniques." Journal Article. *Transportation Research Part a-Policy and Practice* 105: 138–53. doi:10.1016/j.tra.2017.08.020.

Grubinger, T., A. Zeileis, and K. P. Pfeiffer. 2014. "Evtree: Evolutionary Learning of Globally Optimal Classification and Regression Trees in R." Journal Article. *Journal of Statistical Software* 61 (1): 1–29. <Go to ISI>://WOS:000349840200001.

Helbich, M., W. Brunauer, J. Hagenauer, and M. Leitner. 2013. "Data-Driven Regionalization of Housing Markets." Journal Article. *Annals of the Association of American Geographers* 103 (4): 871–89. doi:10.1080/00045608.2012.707587.

Henrichon, E. G., and Fu King-Sun. 1969. "A Nonparametric Partitioning Procedure for Pattern Classification." Journal Article. *IEEE Transactions on Computers* C-18 (7): 614–24. doi:10.1109/T-C.1969.222728.

Jackman, Robert W. 1987. "Political Institutions and Voter Turnout in the Industrial Democracies." Journal Article. *American Political Science Review* 81 (2): 405–23.

James, G., D. Witten, T. J. Hastie, and R. J. Tibshirani. 2013. *An Introduction to Statistical Learning with Applications in R*. Book. Springer Texts in Statistics. New York: Springer-Verlag. doi:10.1007/978-1-4614-7138-7.

Kahle, David, and Hadley Wickham. 2013. "Ggmap: Spatial Visualization with Ggplot2." *The R Journal* 5 (1): 144–61. http://journal.r-project.org/archive/2013-1/kahle-wickham.pdf.

Kurt, Imran, Mevlut Ture, and A. Turhan Kurum. 2008. "Comparing Performances of Logistic Regression, Classification and Regression Tree, and Neural Networks for Predicting Coronary Artery Disease." Journal Article. *Expert Systems with Applications* 34 (1): 366–74. doi:https://doi.org/10.1016/j.eswa.2006.09.004.

Liaw, Andy, and Matthew Wiener. 2002. "Classification and Regression by randomForest." *R News* 2 (3): 18–22. https://CRAN.R-project.org/doc/Rnews/.

Logan, J. R., S. Spielman, H. W. Xu, and P. N. Klein. 2011. "Identifying and Bounding Ethnic Neighborhoods." Journal Article. *Urban Geography* 32 (3): 334–59. <Go to ISI>://WOS:000290446700003.

Logan, John R., Jason Jindrich, Hyoungjin Shin, and Weiwei Zhang. 2011. "Mapping America in 1880: The Urban Transition Historical Gis Project." Journal Article. *Historical Methods: A Journal of Quantitative and Interdisciplinary History* 44 (1): 49–60. doi:10.1080/01615440.2010.517509.

Loh, W. Y. 2011. "Classification and Regression Trees." Journal Article. *Wiley Interdisciplinary Reviews-Data Mining and Knowledge Discovery* 1 (1): 14–23. doi:10.1002/widm.8.

Manwani, N., and P. S. Sastry. 2012. "Geometric Decision Tree." Journal Article. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*

*(Cybernetics)* 42 (1): 181–92. doi:10.1109/TSMCB.2011.2163392.

Menze, Bjoern H, B Michael Kelm, Daniel N Splitthoff, Ullrich Koethe, and Fred A Hamprecht. 2011. "On Oblique Random Forests." Conference Proceedings. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 453–69. Springer.

Menze, Bjoern, and Nico Splitthoff. 2012. *ObliqueRF: Oblique Random Forests from Recursive Linear Model Splits*. https://CRAN.R-project.org/package=obliqueRF.

Mersmann, Olaf. 2018. *Microbenchmark: Accurate Timing Functions*. https://CRAN.R-project.org/package=microbenchmark.

Milborrow, Stephen. 2018. *Rpart.plot: Plot 'Rpart' Models: An Enhanced Version of 'Plot.rpart'*. https://CRAN.R-project.org/package=rpart.plot.

Murthy, Sreerama K., Simon Kasif, and Steven Salzberg. 1994. "A System for Induction of Oblique Decision Trees." Journal Article. *JAIR* 2: 1–32.

Neuwirth, Erich. 2014. *RColorBrewer: ColorBrewer Palettes*. https://CRAN.R-project.org/package=RColorBrewer.

Newman, D.J., S. Hettich, C.L. Blake, and C.J. Merz. 1998. "UCI Repository of Machine Learning Databases." University of California, Irvine, Dept. of Information; Computer Sciences. http://www.ics.uci.edu/~mlearn/MLRepository.html.

Paez, A., T. Uchida, and K. Miyamoto. 2001. "Spatial Association and Heterogeneity Issues in Land Price Models." Journal Article. *Urban Studies* 38 (9): 1493–1508. doi:10.1080/00420980120076768.

Powell, G Bingham. 1982. *Contemporary Democracies*. Book. Harvard University Press.

———. 1986. "American Voter Turnout in Comparative Perspective." Journal Article. *American Political Science Review* 80 (1): 17–43.

Praskievicz, S. 2018. "River Classification as a Geographic Tool in the Age of Big Data and Global Change." Journal Article. *Geographical Review* 108 (1): 120–37. doi:10.1111/gere.12251.

Qiu, Xueheng, Le Zhang, Ponnuthurai Nagaratnam Suganthan, and Gehan AJ Amaratunga. 2017. "Oblique Random Forest Ensemble via Least Square Estimation for Time Series Forecasting." Journal Article. *Information Sciences* 420: 249–62.

Ren, Ye, Le Zhang, and Ponnuthurai N Suganthan. 2016. "Ensemble Classification and Regression-Recent Developments, Applications and Future Directions." Journal Article. *IEEE Computational Intelligence Magazine* 11 (1): 41–53.

Ripley, Brian. 2018. *Tree: Classification and Regression Trees*. https://CRAN.R-project.org/package=tree.

Robertson, B. L., C. J. Price, and M. Reale. 2013. "CARTopt: A Random Search Method for Nonsmooth Unconstrained Optimization." Journal Article. *Computational Optimization and Applications* 56 (2): 291–315. doi:10.1007/s10589-013-9560-9.

Rokach, Lior. 2010. "Ensemble-Based Classifiers." Journal Article. *Artificial*

*Intelligence Review* 33 (1-2): 1–39.

Stockemer, D. 2017. "What Affects Voter Turnout? A Review Article/Meta-Analysis of Aggregate Research." Journal Article. *Government and Opposition* 52 (4): 698–722. doi:10.1017/gov.2016.30.

Suhail, Z., E. R. E. Denton, and R. Zwiggelaar. 2018. "Tree-Based Modelling for the Classification of Mammographic Benign and Malignant Micro-Calcification Clusters." Journal Article. *Multimedia Tools and Applications* 77 (5): 6135–48. doi:10.1007/s11042-017-4522-3.

Therneau, Terry, Beth Atkinson, and Brian Ripley. 2017. *Rpart: Recursive Partitioning and Regression Trees*. https://CRAN.R-project.org/package=rpart.

Wheeler, D. C., A. Paez, J. Spinney, and L. A. Waller. 2014. "A Bayesian Approach to Hedonic Price Analysis." Journal Article. *Papers in Regional Science* 93 (3): 663–83. doi:10.1111/pirs.12003.

Wickham, Hadley. 2011. "The Split-Apply-Combine Strategy for Data Analysis." *Journal of Statistical Software* 40 (1): 1–29. http://www.jstatsoft.org/v40/i01/.

———. 2017. *Tidyverse: Easily Install and Load the 'Tidyverse'*. https://CRAN.R-project.org/package=tidyverse.

———. 2018. *Scales: Scale Functions for Visualization*. https://github.com/hadley/scales.

Wickham, Hadley, and Jennifer Bryan. 2018. *Readxl: Read Excel Files*. https://CRAN.R-project.org/package=readxl.

Wickham, Hadley, Jim Hester, and Romain Francois. 2017. *Readr: Read Rectangular Text Data*. https://CRAN.R-project.org/package=readr.

Wickramarachchi, D. C., B. L. Robertson, M. Reale, C. J. Price, and J. Brown. 2016. "HHCART: An Oblique Decision Tree." Journal Article. *Computational Statistics & Data Analysis* 96: 12–23. doi:10.1016/j.csda.2015.11.006.

Williams, Graham J. 2011. *Data Mining with Rattle and R: The Art of Excavating Data for Knowledge Discovery*. Use R! Springer. http://www.amazon.com/gp/product/1441998896/ref=as_li_qf_sp_asin_tl?ie=UTF8&tag=togaware-20&linkCode=as2&camp=217145&creative=399373&creativeASIN=1441998896.

Wong, D. W. S., and Q. Y. Huang. 2017. "'Voting with Their Feet': Delineating the Sphere of Influence Using Social Media Data." Journal Article. *Isprs International Journal of Geo-Information* 6 (11): 16. doi:10.3390/ijgi6110325.

Xie, Yihui. 2015. *Dynamic Documents with R and Knitr*. 2nd ed. Boca Raton, Florida: Chapman; Hall/CRC. https://yihui.name/knitr/.

———. 2018. *Knitr: A General-Purpose Package for Dynamic Report Generation in R*. https://yihui.name/knitr/.

Yang, L. J., S. S. Liu, S. Tsoka, and L. G. Papageorgiou. 2017. "A Regression Tree Approach Using Mathematical Programming." Journal Article. *Expert Systems with Applications* 78: 347–57. doi:10.1016/j.eswa.2017.02.013.

Zhang, Le, and Ponnuthurai N Suganthan. 2015. "Oblique Decision Tree Ensemble via Multisurface Proximal Support Vector Machine." Journal Article. *IEEE Transactions on Cybernetics* 45 (10): 2165–76.

Zhang, Le, and Ponnuthurai Nagaratnam Suganthan. 2014. "Random Forests with Ensemble of Feature Spaces." Journal Article. *Pattern Recognition* 47 (10):

3429–37.

———. 2017. “Benchmarking Ensemble Classifiers with Novel Co-Trained Kernel Ridge Regression and Random Vector Functional Link Ensembles [Research Frontier].” Journal Article. *IEEE Computational Intelligence Magazine* 12 (4): 61–72.

Zhang, Le, Jagannadan Varadarajan, Ponnuthurai Nagaratnam Suganthan, Narendra Ahuja, and Pierre Moulin. 2017. “Robust Visual Tracking Using Oblique Random Forests.” Conference Proceedings. In *Proceedings of the Ieee Conference on Computer Vision and Pattern Recognition*, 5589–98.

Zhu, Hao. 2018. *KableExtra: Construct Complex Table with 'Kable' and Pipe Syntax*. https://CRAN.R-project.org/package=kableExtra.