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 show how the use of IBFs can often improve the performance of models. 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).

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 $X \in \mathrm{R}^p$. 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, Θ_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 θ_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:

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

Within this framework, each branch m is a collection of sequential partitions θ_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 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 Wickramarachchi et al. 2016 for an overview).

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 base 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:

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

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 R^p defined by X still reproduce a linear orthogothal partition of this 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 as:

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

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 in 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.

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 $a_1 = a_2 = 1$:

$$h_1(b_1(X_1), b_1(X_2)) = b_1(X_1) + b_2(X_2) = a_1X_1 + a_2X_2$$

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. Compare to the exaustive search for oblique decision boundaries ($O(2^p \times \binom{n}{p})$), in Murthy, Kasif, and Salzberg 1994), or an algorithm such as HHCART, which has complexity of $O(Cp^2nlog(n))$ (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.

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 κ 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 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 that are widely used in the literature to test the performance of machine learning techniques. The datasets are listed in Table 1.

As noted above, centering and scaling the variables is important for the use of IBFs. For the experiments reported in this section, all quantitative features have been centered on the mean and scaled to the unit interval. The experiments are using k-fold cross-validation, a technique commonly used to assess the performance of algorithms that involves randomly thinning a dataset to obtain a $(100 - 100/k)\%$ training sample, in a procedure that is repeated k times. In this section we use 10-fold cross-validation.

Table 1: Datasets for Benchmarking Experiment

Dataset	Quantitative.Features	Qualitative.Features	Number.of.Cases	Source
Balance Scale	4	0	625	UCI Machine Learning Repository
Wine	13	0	178	Package rattle.data
BUPA	6	0	345	Package dprep
PIMA Indian	8	0	768	Package mlbench
Glass	9	0	214	Package mlbench
Book Club	9	1	1300	Package mlbench

Since one advantage of using IBFs is that they can be used in virtually any implementation of DTs, we choose to illustrate their broad applicability using two different methods: a conventional tree algorithm (as implemented in the R package `tree`, see Ripley 2018) and an evolutionary algorithm (as implemented in the package `evtree`, see Grubinger, Zeileis, and Pfeiffer 2014). Each dataset and method is used to train trees with oblique partitions and with IBFs. Other R packages exist that implement non-oblique 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).

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

$$
\begin{aligned}
h_1(b_1(X_1), b_1(X_2)) &= b_1(X_1) + b_1(X_2) = X_1 + X_2 \\
h_1(b_2(X_1), b_2(X_2)) &= b_2(X_1) + b_2(X_2) = X_1^2 + X_2^2 \\
h_2(b_1(X_1), b_1(X_2)) &= b_1(X_1)b_2(X_2) = X_1 X_2 \\
h_2(b_1(X_1)b_3(X_2)) &= b_1(X_1)b_3(X_2) = X_1 e^{X_2}
\end{aligned}
$$

The size of the experiment is as follows: two different methods (tree/evolutionary tree), six different datasets, two different types of partitions (oblique/IBFs), and ten replications for each (i.e., 10-fold cross-validation).

We report results regarding accuracy, size of the resulting trees, and computer time. With respect to performance, Figure 8) shows the accuracy (in percentage) of the various implementations of the methods. It can be seen there that introducing IBFs leads at least to comparable performance, and more often results in performance gains - with the gains in some cases being quite substantial. This is true regardless of whether the method is the conventional or evolutionary tree.

The gains in accuracy do not necessarily imply more complex trees. As seen in Figure 9, the *size* of the tree, measured in terms of the number of terminal nodes, can increase at times (for example, in the case of the BUPA and PIMA datasets). Other times, the use of IBFs can result in substantial gains in accuracy simultaneously with a modest reduction in the size of the tree (as in the case of the Glass dataset).

The last result that we report concerns computer time. This part of the experiment was conducted using the `microbenchmark` package (Mersmann 2018). This package times the evaluation of expressions with sub-millisecond accuracy. Each method/dataset/partition type (orthogonal, IBF) was evaluated 50 times to

produce the results shown in Figure 10. As seen there, the time required to train trees with IBFs is comparable or occasionally somewhat more favorable than with oblique partitions, perhaps as a consequence of finding better partitions in fewer dimensions.

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 11.

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:

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

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 model are shown in Figure 12, 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 13.

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 14. 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., $e^{long} + e^{lat} \geq 2.0$). The ethnic neighborhoods are shown in Figure 15. 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

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 16 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 17, 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 18, 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 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 19). 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 20) 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 substantially 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 seventeen variables, with descriptive statistics as shown in Table 2. 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 21. 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 22. The $pseudo - R^2$ of this model is 0.44. Note that three of the partitions in this model are oblique, and one 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

Table 2: 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

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 23.

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 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 to allow for oblique partitions. In this paper, we have 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.

A benchmarking exercise shows that the use of IBFs can improve the accuracy of the model and/or produce more parsimonious models, without greatly increasing computation time. Furthermore, three case studies illustrated 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

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 the result of the 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.

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.

Folch, D. C., and S. E. Spielman. 2014. "Identifying Regions Based on Flexi-

ble 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 Net-

works 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.

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, 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.

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.

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.

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

List of Figures

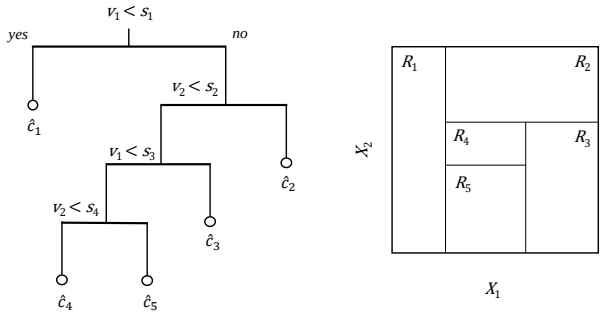

Figure 1: Prototypical Decision Tree

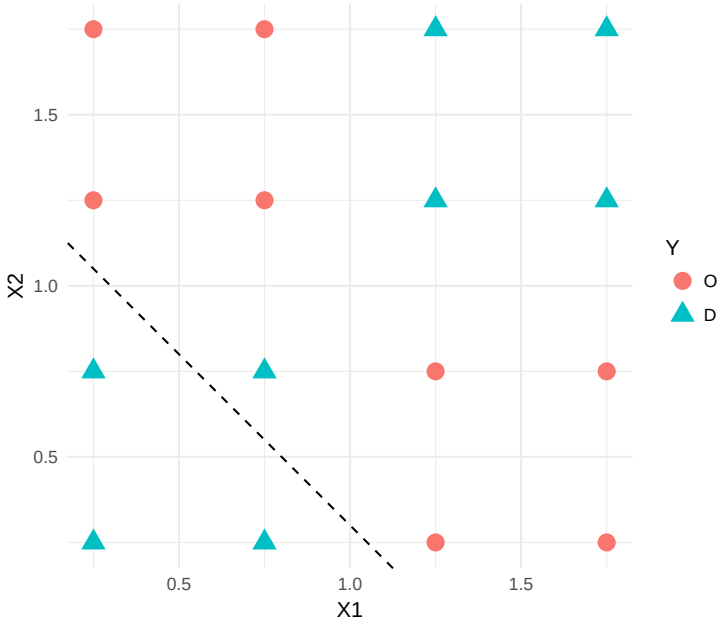

Figure 2: Chessboard example

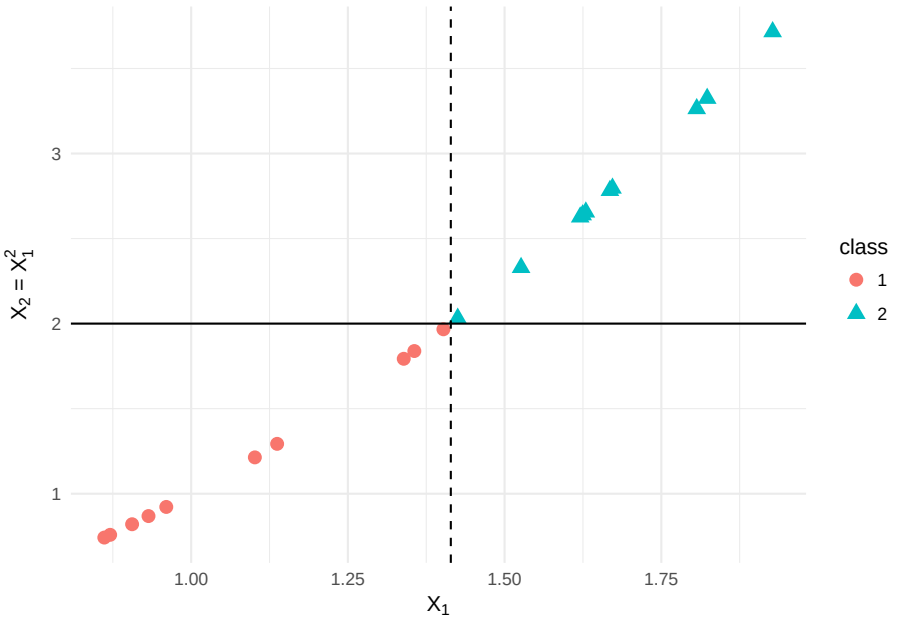

Figure 3: Example of a basis function used as a feature to train a Decision Tree

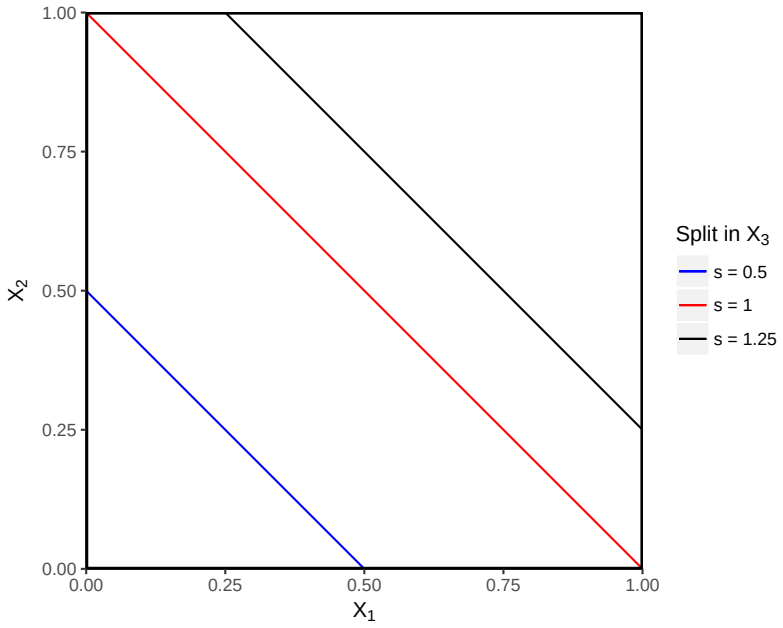

Figure 4: Oblique basis function and projections with different splitting points (s)

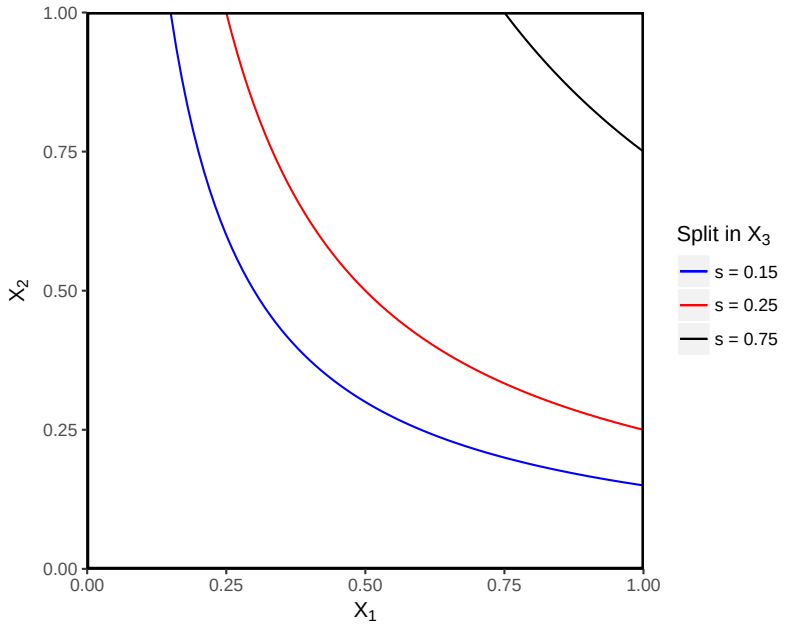

Figure 5: Hyperbolic basis function and projections with different splitting points (s)

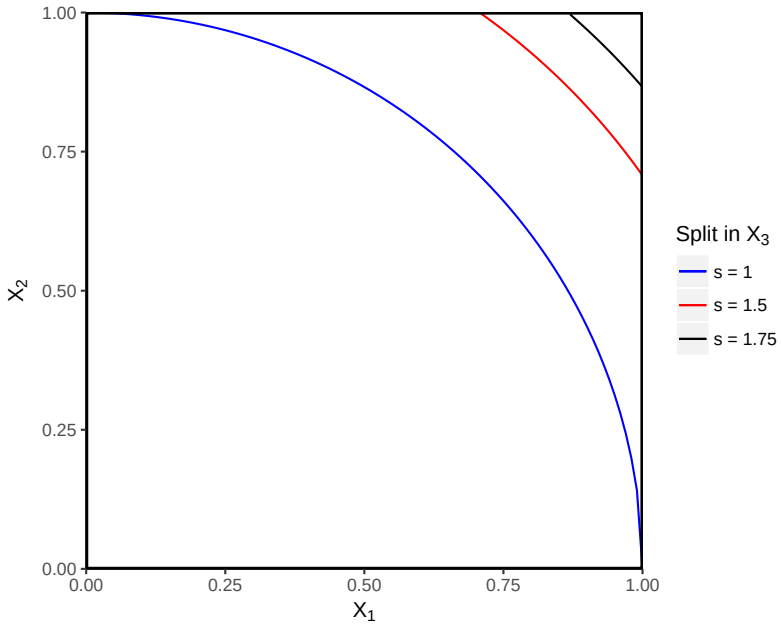

Figure 6: Radial IBF and projections with different splitting points (s)

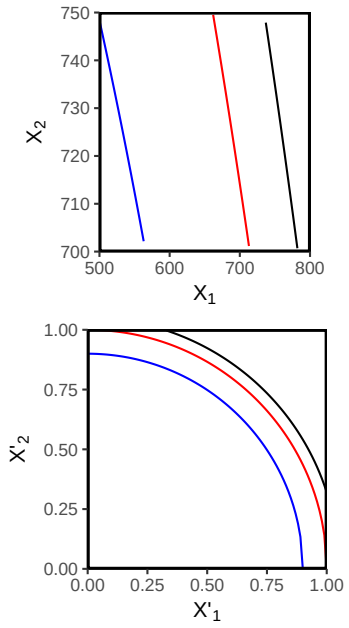

Figure 7: Behavior of radial IBF with non-centered/non-scaled variables (top panel) and centered/scaled variables (bottom panel)

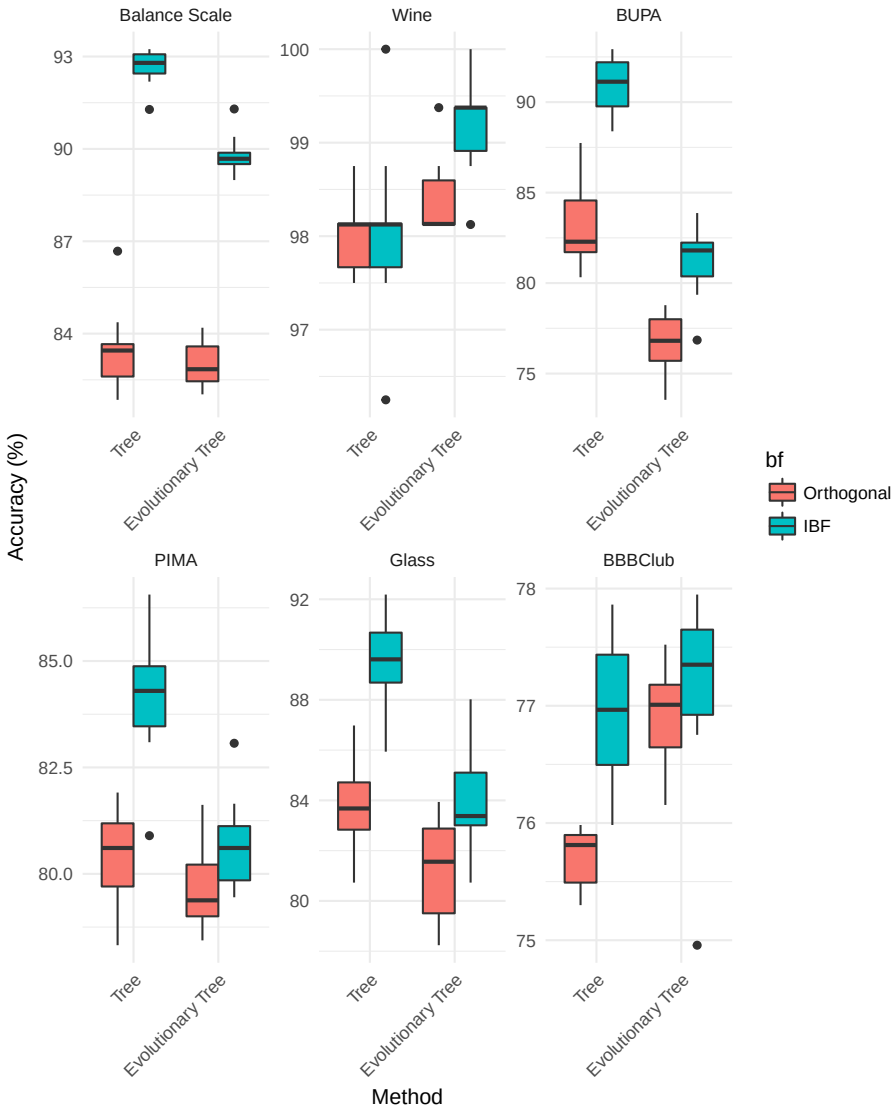

Figure 8: Benchmarking results: Classification accuracy

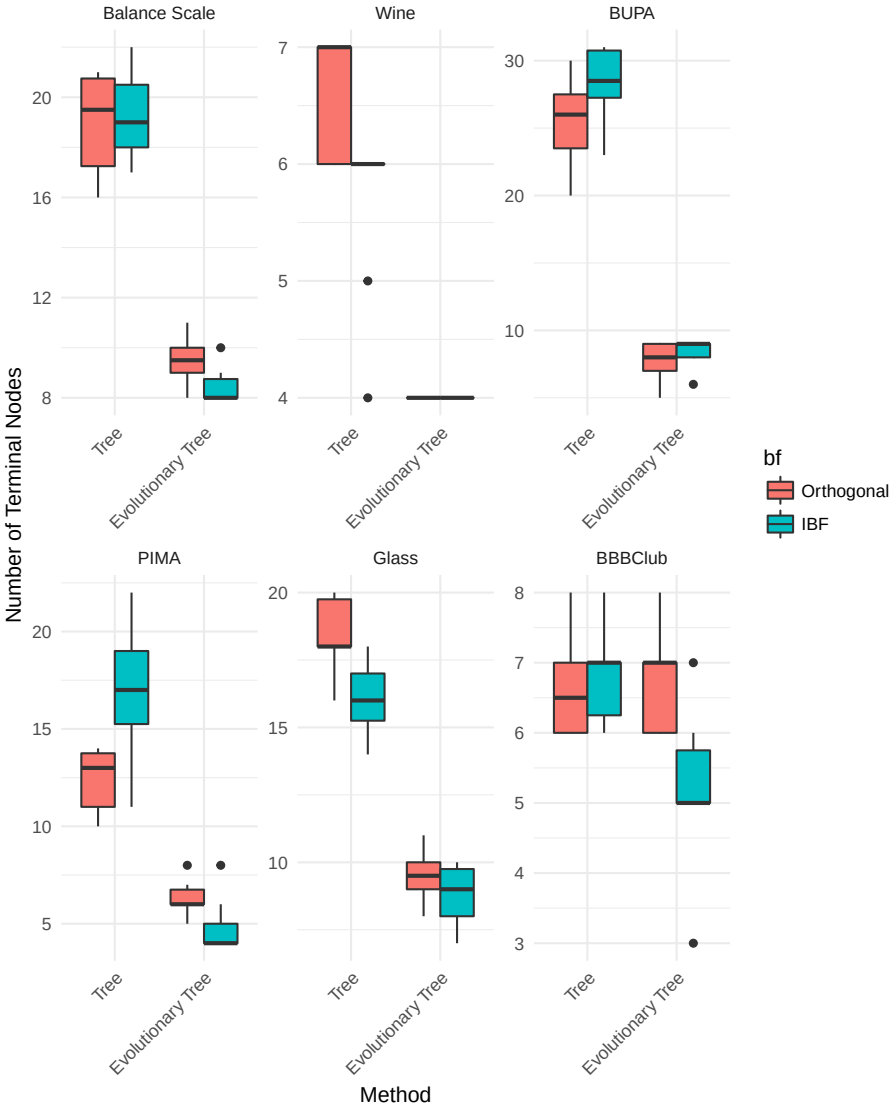

Figure 9: Benchmarking results: Tree size

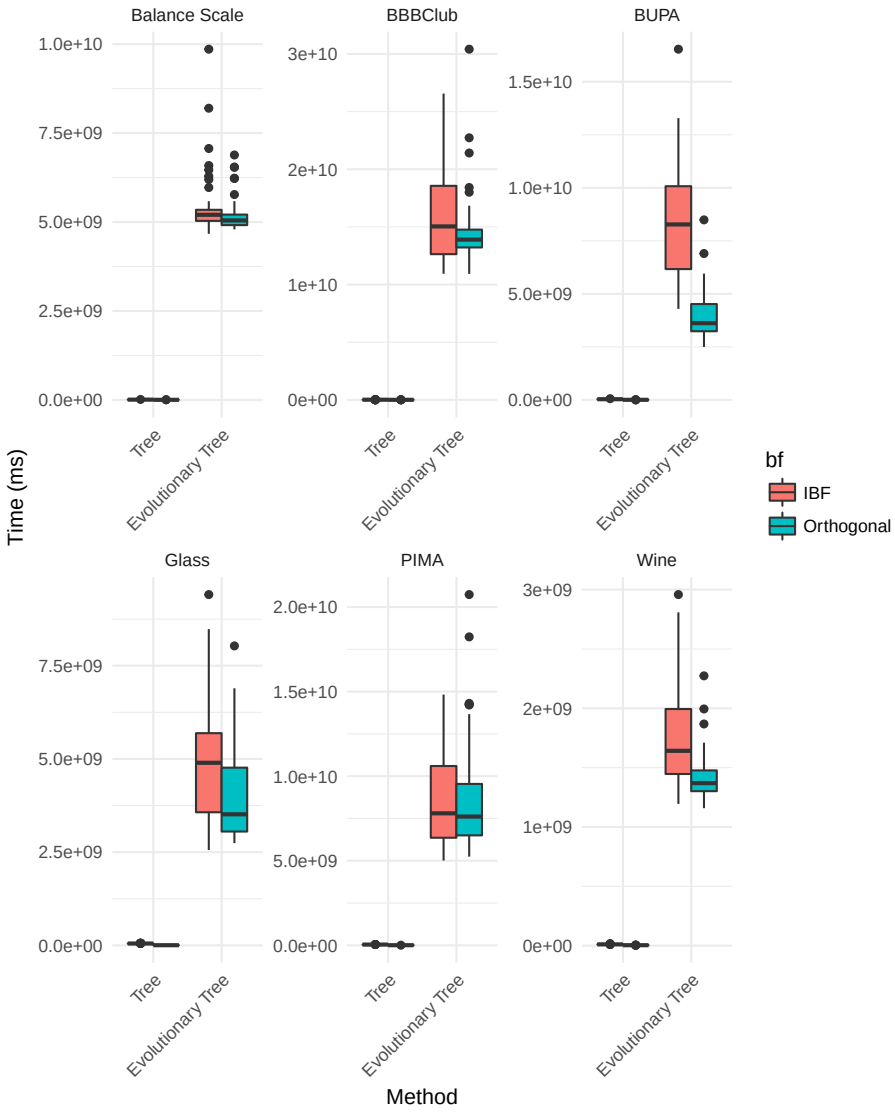

Figure 10: Benchmarking results: Computing time

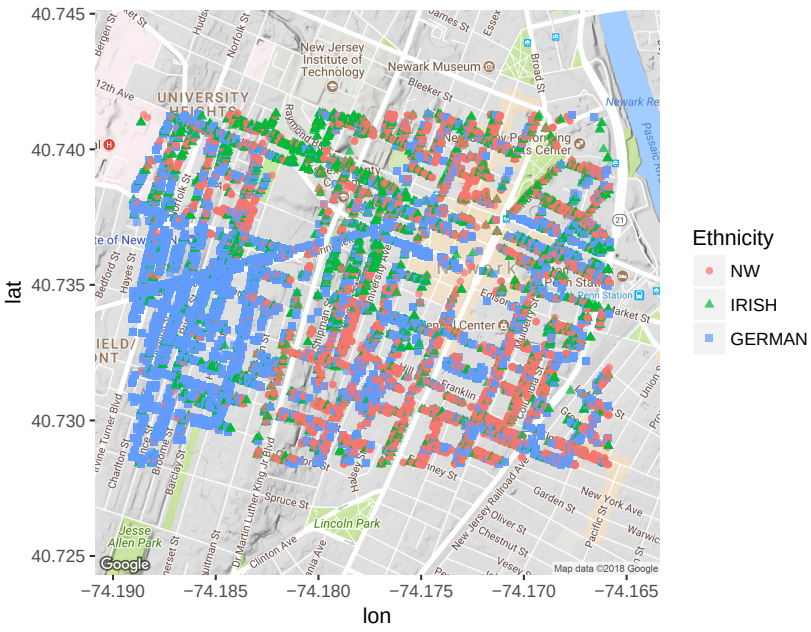

Figure 11: Three ethnic groups in Newark (1880 US Census)

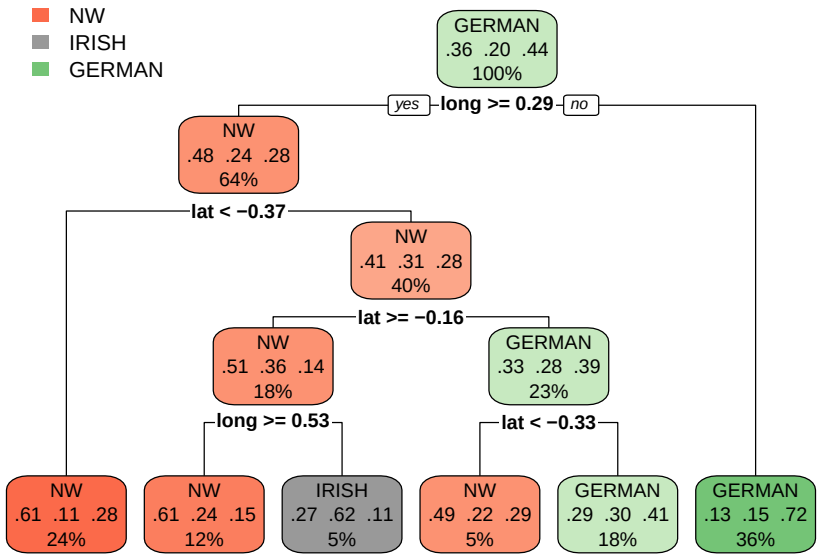

Figure 12: Decisiton tree with orthogonal partitions for spatial classification, Newark ethnic groups

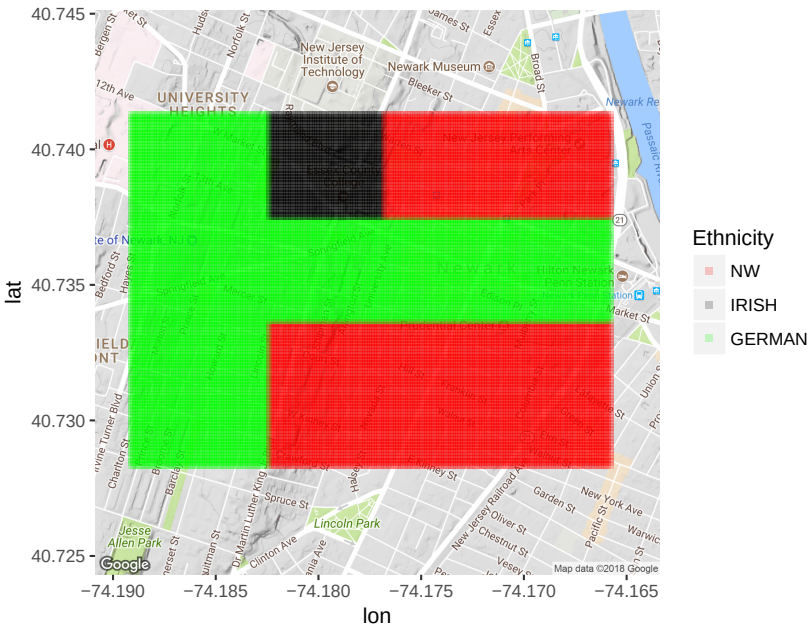

Figure 13: Ethnic neighborhoods in Newark, 1880, using orthogonal partitions

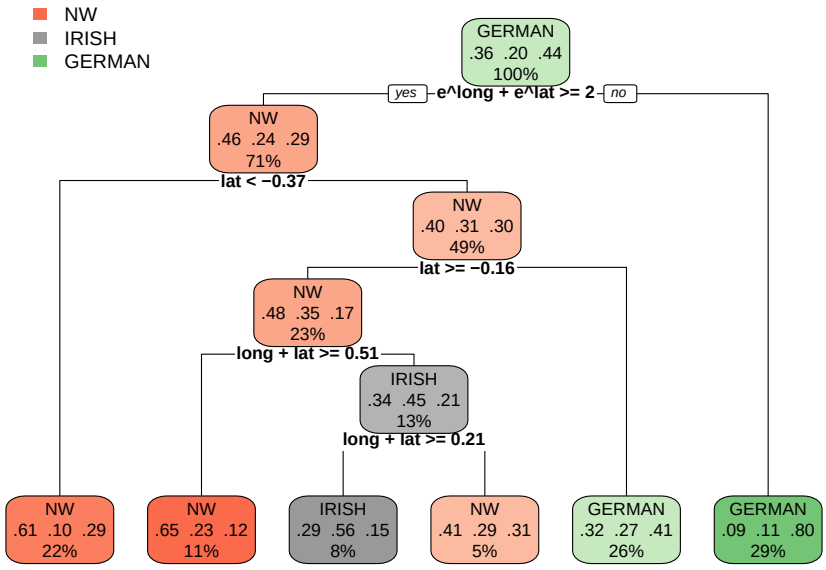

Figure 14: Decision tree with non-orthogonal/non-linear partitions for spatial classification, Newark ethnic groups

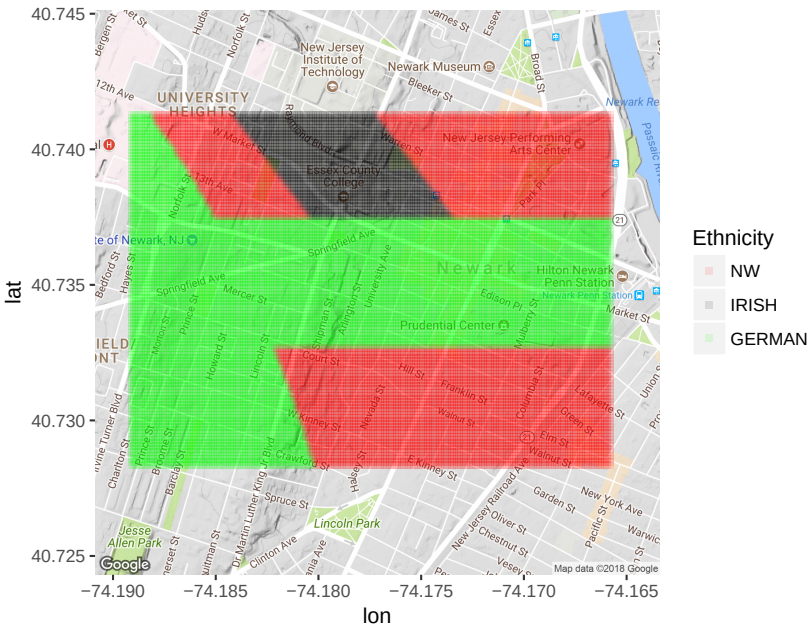

Figure 15: Ethnic neighborhoods in Newark, 1880, using non-orthogonal partitions

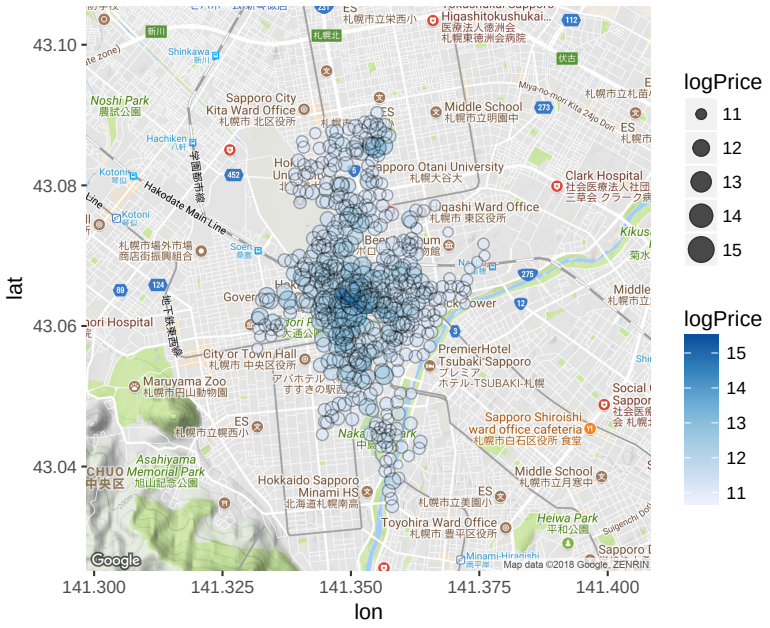

Figure 16: Land prices in Sapporo (log)

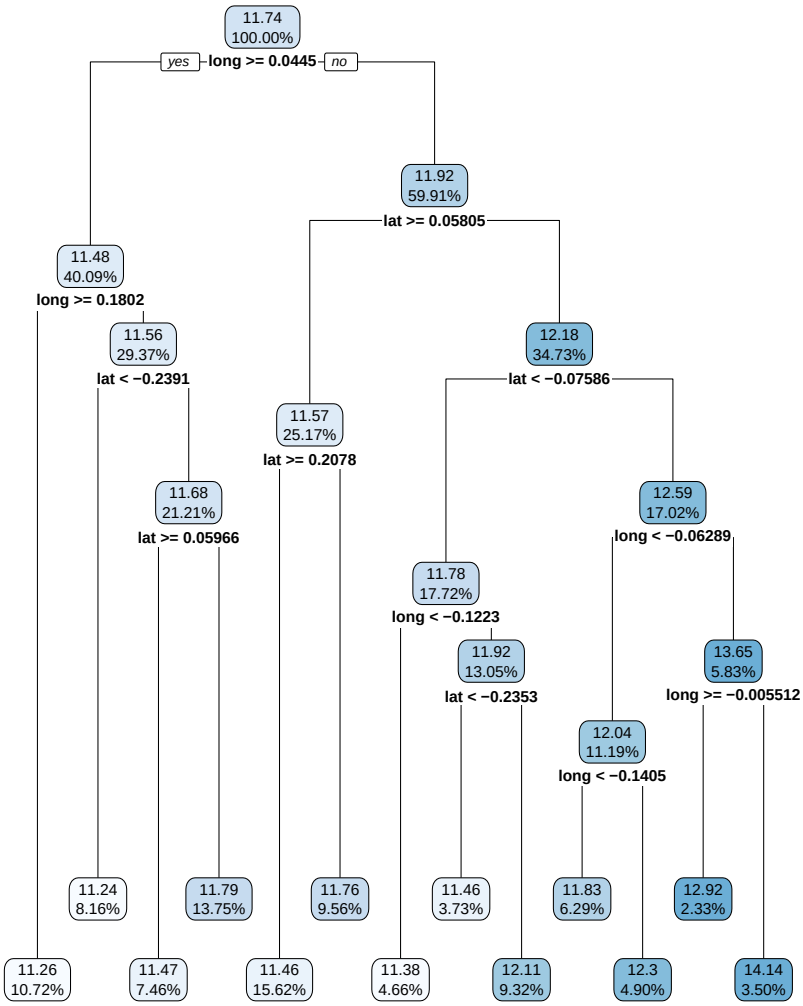

Figure 17: Decision tree with orthogonal partitions for submarket identification, Sapporo land prices

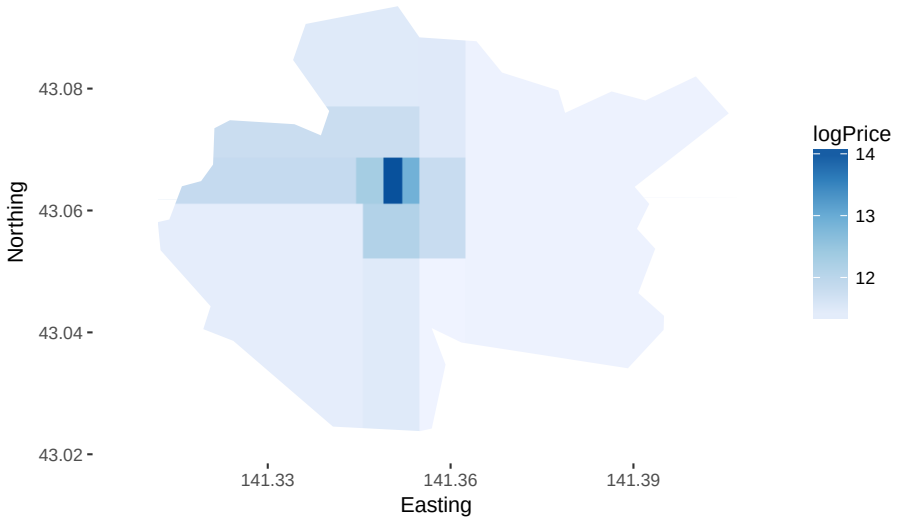

Figure 18: Spatial land price submarkets in Sapporo using orthogonal partitions

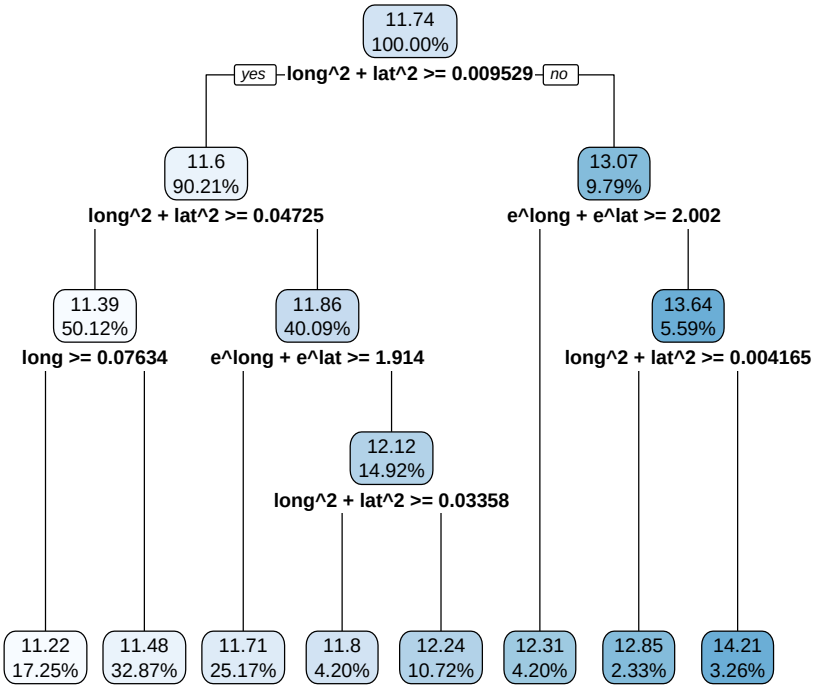

Figure 19: Decision tree with non-orthogonal/non-linear partitions for submarket identification, Sapporo land prices

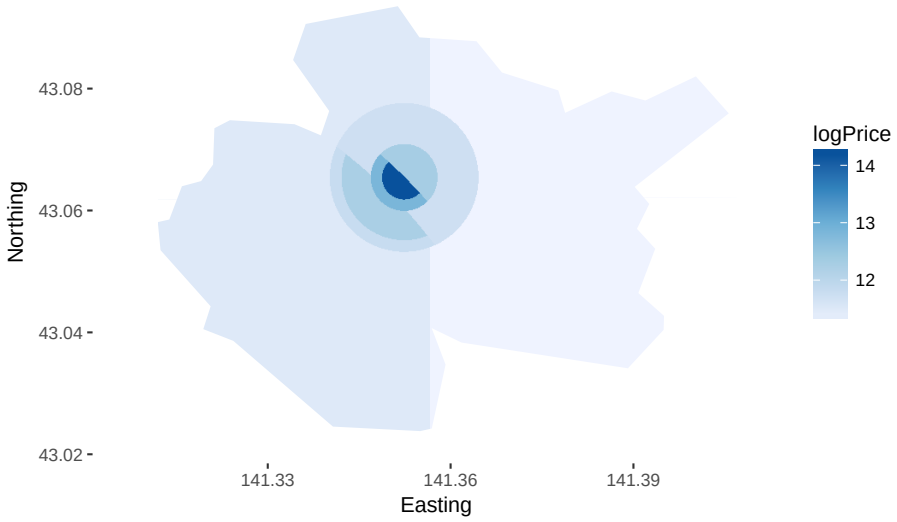

Figure 20: IBF-based non-orthogonal land price regions (log) in Sapporo

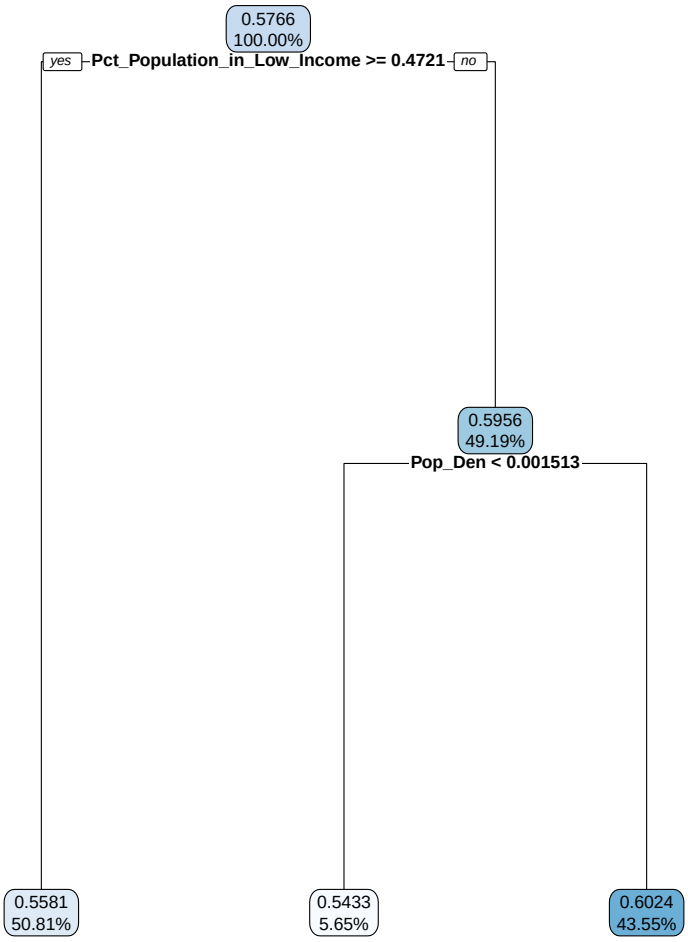

Figure 21: Decision tree with orthogonal partitions, Ontario voter turnout

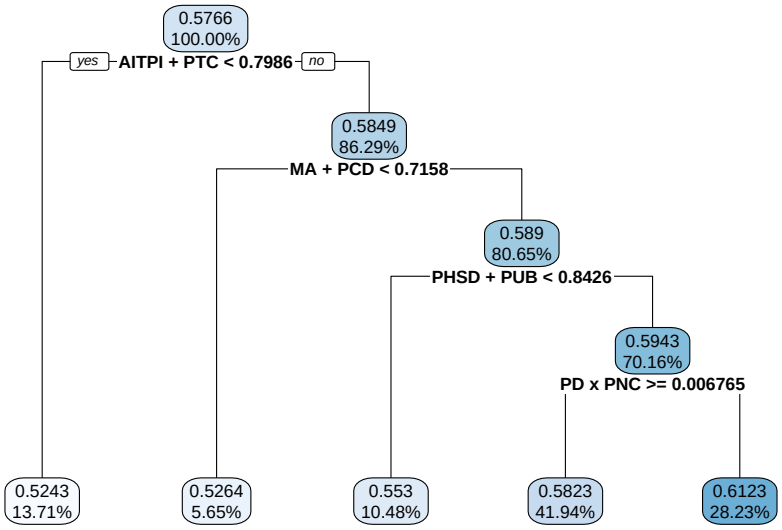

Figure 22: Decision tree with non-orthogonal/non-linear partitions, Ontario voter turnout

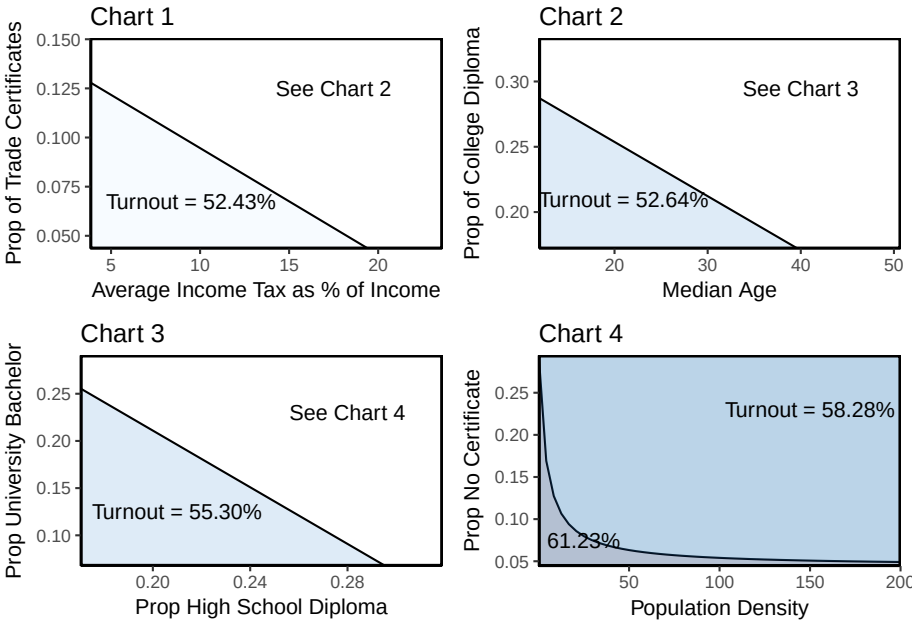

Figure 23: Decision charts for voter turnout