CEUR-WS.org/Vol-3475/paper4.pdf

An Evolutionary Geospatial Regression Tree

Margot Geerts[1,*], Seppe vanden Broucke[2,1] and Jochen De Weerdt[1]

[1]*Research Centre for Information Systems Engineering, KU Leuven, Naamsestaat 69, 3000 Leuven, Belgium*
[2]*Department of Business Informatics and Operations Management, Ghent University, Tweekerkenstraat 2, 9000 Gent, Belgium*

Abstract

Tree-based methods have become popular for spatial prediction tasks due to their high accuracy in dividing input spaces into regions with different predictions. However, traditional decision trees perform univariate splits, resulting in rectangular regions. To address this limitation and provide more intuitive and accurate decision boundaries for spatial data, we propose a novel Geospatial Regression Tree (GeoTree) with two multivariate geospatial split types, i.e. oblique and Gaussian splits. Our approach relies on evolutionary algorithms to decide on the optimal split type, chosen among axis-parallel, oblique and Gaussian splits, in each internal node. We conducted a simulation study using five synthetic datasets to demonstrate GeoTree's ability to capture orthogonal, diagonal and ellipse patterns accurately. The results confirm the proposed method's advantage in tree depth and stability. We also tested GeoTree on real-life residential property valuation and soil content data. Our experiments revealed that the geospatial split types in GeoTree improve interpretability and maintain predictive power for price prediction and soil mapping tasks based on X- and Y-coordinates. The resulting decision boundaries are more intuitive spatial patterns on a geographic map of the study area.

Keywords

Decision trees, Spatial data, Evolutionary algorithms, Machine learning, Property valuation, Soil mapping.

1. Introduction

Spatial prediction tasks pose a significant challenge due to the unique nature of the spatial data. Spatial data is characterized by spatial dependence and spatial heterogeneity, which makes it difficult to analyze. For spatial applications such as soil mapping [1], disease mapping [2, 3], property valuation [4] and land use or cover mapping [5], tree-based models have proven to be appropriate. However, traditional decision trees with univariate splits are limited in their ability to capture spatial patterns. By drawing axis-oblique decision boundaries, the accuracy of the method for spatial data can be significantly improved [6]. Furthermore, introducing multivariate geospatial splits can improve the accuracy of tree based models for spatial prediction tasks as they allow to learn from X- and Y-coordinates simultaneously.

The recent advancements in Multivariate Decision Tree (MDT) learning have primarily focused on linear combinations in splits and their optimization. On the other hand, omnivariate trees can allow test conditions to follow any non-linear function. However, most spatial data sets exhibit recognizable patterns. For example, house prices often follow street patterns, where certain (parts of) streets are more expensive than others. Similarly, circular patterns can be observed in historic city centers when zooming out to the city-level. For instance, cities such as Brussels, Belgium and London, United Kingdom, are surrounded by a ring road that separates the city center from the suburbs. Moreover, administrative boundaries frequently follow non-rectangular shapes, as well as house prices, as shown in Figure 1. Despite these spatial patterns, previous research in spatial prediction has yet to incorporate them into tree-based models.

Therefore, we propose a new multivariate decision tree algorithm that integrates three different split types: axis-parallel splits, oblique splits and Gaussian splits. In addition, we introduce a Genetic Algorithm (GA) to generate the oblique and Gaussian candidate splits within the internal nodes. The combination of these multivariate split types with the traditional axis-parallel split enables the proposed method to recognize spatial patterns in geographic location data more accurately and with simpler trees. As a result, the Geospatial Regression Tree (GeoTree) provides more intuitive decision boundaries and less complex trees, making it more interpretable than existing decision tree algorithms. We demonstrate that our new decision tree algorithm can capture spatial patterns more effectively than both traditional regression trees and oblique regression trees for regression tasks. Using synthetic data sets, we showcase the proposed method's efficient pattern learning ability, with limited tree depth and error. In particular, our advanced synthetic data set that reflects real spatial patterns demonstrates GeoTree's superior performance in error, tree depth and stability. Finally, we apply the proposed tree algorithm with the geospatial split types to real-life property valuation and soil mapping data sets, demonstrating similar predictive power compared to existing decision tree algorithms, while significantly enhancing interpretability due

STRL'23: Second International Workshop on Spatio-Temporal Reasoning and Learning, 21 August 2023, Macao, S.A.R

[*] Corresponding author.

margot.geerts@kuleuven.be (M. Geerts)

© 2023 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).
CEUR Workshop Proceedings (CEUR-WS.org)

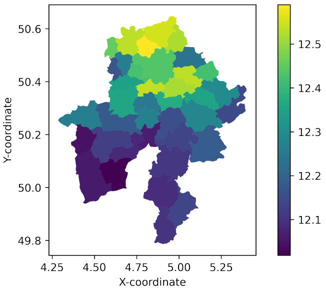


Figure 1: Median log price per commune in the province of Namur, Belgium.

to shallower and smaller decision tree models and more intuitive decision boundaries that can be easily visualized on a geographic map.

In the next section, we will review relevant literature on MDTs and iterative methods used for decision tree learning. In Section 3, we introduce the inner workings of GeoTree. We report the results for the experimental evaluation in Section 4 and critically discuss the proposed GeoTree in comparison with two baseline methods using on the one hand synthetic data sets and on the other hand real-life property valuation and soil mapping data. Finally, we summarize our findings and provide a critical analysis of GeoTree, concluding with future research directions in this area.

2. Related Work

As spatial prediction tasks benefit from the introduction of multivariate splits in tree-based models, the following section introduces MDTs and approaches to MDT learning. MDTs differ from traditional decision trees in the split type used. While traditional decision trees split the data based on a test condition of the form $x_i \leq c$, with x_i the ith feature, MDTs include multivariate conditions, $f(\mathbf{x}) \leq c$. In [7], multivariate decision tree algorithms are discussed. The authors identify three split types, univariate splits, multivariate splits that consist of linear combinations of features, and multivariate splits that include non-linear combinations. Most researchers have focused on axis-oblique splits, i.e. linear combinations, and have demonstrated the benefits for spatial analysis [6, 1] and classification in general [8]. However, non-linear combinations have been included using quadratic functions and power laws [9, 10]. Further, splits have been generalized even more by using neural networks, decision trees and random forest at internal nodes [11]. Nevertheless, this generalization increases the cost complexity of learning an MDT substantially.

Instead of non-linear functions with many parameters, constraining splits to a certain shape such as hyperplanes or Gaussians might reduce complexity. However, decision tree algorithms that have implemented Gaussian distributions also introduce soft decisions [12, 13]. In addition to introducing hard Gaussian decisions, our proposed method combines axis-parallel, axis-oblique and Gaussian splits in contrast to previous work.

Another approach to address complexity is to estimate the weights w_i or function f iteratively. While in [8] an analytical approach is taken by constructing a Householder matrix from the dominant eigenvectors of the covariance matrix of each class to find the best hyperplane, heuristics such as single-solution based metaheuristics, evolutionary algorithms and swarm intelligence methods can improve split searching [14]. Evolutionary algorithms have been more popular recently, but researchers have mostly used them for global search rather than split searching. To find near-optimal splits, genetic algorithms (GA) are employed most often within the family of evolutionary algorithms. A GA is a heuristic that lets a population of individuals evolve over generations, using three modification operators, with respect to a fitness function. In [9], two GAs are used to search function parameters and weights for non-linear splits. Whereas they require a GA at both levels, we reduce complexity by employing one GA for searching parameters for either an oblique split or a Gaussian split.

3. An Evolutionary Geospatial Regression Tree Algorithm

Our work focuses on binary decision trees for regression with numerical features. A regression tree takes as input an observation $(x_1, ..., x_d, y)$ where x_i are the features which are real-valued, d is the number of features and y is the real-valued target variable. A trained regression tree consists of test conditions of the form $x_i \leq c$ with c a constant, producing axis-parallel splits. Typically, a top-down greedy approach is used for building the tree. This means that the best split, determined by a feature i and constant c, is chosen at each node, starting at the root node, effectively splitting each node into two child nodes until the specified depth is reached or the leaf nodes are pure. The best split produces the largest reduction in mean squared error of the predictions. In essence, a regression tree sorts the observations in the training set through the tree, according to the test conditions in the internal nodes, and predicts in each leaf node the average of all observations sorted into that node.

GeoTree's key innovation lies in the inclusion of two new bivariate decision tree splits, the oblique split and a Gaussian split, in addition to the axis-parallel split. The oblique split divides the input space in two regions based on a linear combination of two features, $w_1 * x_1 +$

$w_2 * x_2 \geq c$. On the other hand, the Gaussian split defines an ellipse in a two-dimensional space, $d(x, f_1) + d(x, f_2) \geq c$. This is supported by the mathematical property that an ellipse consists of all points of which the sum of the distances to both focal points (f_1, f_2) is equal to a constant. Therefore, the Gaussian split separates input data located inside of the ellipse from input data located on or outside of the ellipse.

As for oblique splits, finding the best Gaussian split is NP-hard [15]. In addition, a brute-force search that enumerates all possible hyperplanes and ellipses, based on two and three data points respectively, has an exponential cost. Two points are necessary to define a hyperplane and three to define an ellipse, where two points function as focal points and the third is used to calculate the constant c. A GA is employed instead to generate oblique and Gaussian candidate splits in each internal node. We base the fitness function on the evaluation criterion for the decision tree, that is, the gain in mean squared error. As such, the fitness function is defined by

$$MSE_{parent} - (\frac{|\mathcal{L}|}{n} \sum_{l \in \mathcal{L}} (y_l - \bar{y})^2 + \frac{|\mathcal{R}|}{n} \sum_{r \in \mathcal{R}} (y_r - \bar{y})^2)$$

where $\mathcal{L}$ and $\mathcal{R}$ are the sets of observations sorted to the left and right respectively by the candidate split. Algorithm 1 presents the logic of the GA-based candidate split generation that returns the best split. This algorithm is invoked for both proposed splits separately, as the notion of an individual is specific to the type of split. As mentioned above, an oblique split can be defined by two two-dimensional points and a Gaussian split by three two-dimensional points. Consequently, an individual for an oblique candidate split is a list of four floats, or two points x_1 and x_2 as can be seen in Figure 2a, to make the hyperplane. An individual for a Gaussian candidate split is defined as a list of six floats, or three points x_1, x_2 and x_3 (see Figure 2b), to generate an ellipse. The population is initialized by randomly drawing n_p times from the uniform distribution defined by the lower boundaries ($\underline{b}$) and upper boundaries ($\overline{b}$). The upper and lower boundaries are defined by the minimum and maximum values of the two data dimensions. To reflect individuals, these values are repeated to result in a list of four values in case of oblique splits, while Gaussian boundaries require a list of six values. After initialization, the evolution process is repeated for the number of generations (n_g). First, a 'Hall of Fame' variable is updated to retain the individuals that have the highest fitness values from the current population. This elitist approach requires a parameter n_h to indicate the number of individuals to retain. Then, the current population is modified using selection, crossover and mutation operators. A tournament selection is performed by selecting the fittest individual in n_p tournaments of size n_t. Crossover is performed subsequently on two individuals with a probability $cxpb$. This step is implemented with a blend crossover and the required parameter α which determines the interval to draw new values from based on the parents. Next, the new offspring are mutated with a probability $mutpb$. Specifically, polynomial bounded mutation is used to ensure individuals range between the boundaries ($\underline{b}$, $\overline{b}$). This operator requires two additional parameters: the probability of each value of the individual to be mutated ($indpb$) and the crowding degree (η). Lastly, the fittest individuals are chosen among the offspring and the fittest of the previous population ('Hall of Fame'). After n_g generations, the fittest individual from the last population is returned as the candidate split. The implementation is based on the Python library DEAP [16].

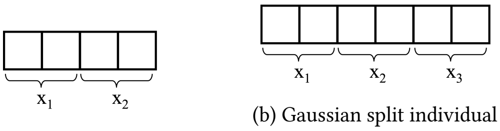
The decision tree algorithm performs a greedy search in each internal node by finding the best orthogonal, oblique and Gaussian candidate split and finally choosing the best split among the three candidate splits. Orthogonal candidate splits are efficiently searched by sorting all unique values for each variable in the data set and evaluating corresponding candidate splits. Oblique and Gaussian splits are generated by Algorithm 1 as explained before. The complete GeoTree algorithm is available at https://github.com/margotgeerts/GeoTree.

Algorithm 1 Genetic algorithm for candidate split generation.

```
Require: $\underline{b}, \overline{b}, n_p, n_g, n_h, n_t, cxpb, \alpha, mutpb, indpb, \eta$
  $Population \leftarrow Repeat(\mathcal{U}(\underline{b}, \overline{b}), n_p)$
  for $i = 0$ to $n_g$ do
      $HallofFame \leftarrow Fittest(Population, n_h)$
      $Offspring \leftarrow TournamentSelection(Population, n_t, n_p)$
      $Offspring \leftarrow Mate(Offspring, cxpb, \alpha)$
      $Offspring \leftarrow Mutate(Offspring, mutpb, indpb, \eta)$
      $Population \leftarrow SelectBest(Offspring + HallofFame, n_p)$
  end for
  $BestIndividual \leftarrow SelectBest(Population, 1)$
  return $BestIndividual$
```

4. Experimental Evaluation

The performance of GeoTree is demonstrated by first comparing the regression outcomes for five synthetic data sets. Second, we show the benefits of our method in two spatial applications, a real-life housing data set and soil mapping data.



(a) Oblique split individual consisting of four chromosomes, that form two points x_1 and x_2.

(b) Gaussian split individual consisting of six chromosomes, that comprise three points x_1, x_2 and x_3.

Figure 2: Two types of GA individuals.

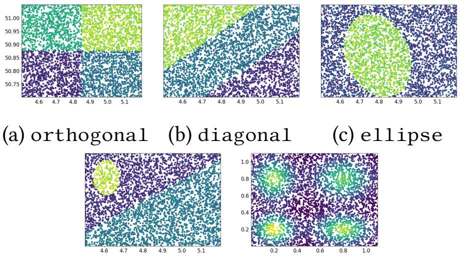
4.1. Experimental Setup

In order to thoroughly test the benefits of the proposed method with two GA-generated split types, axis-oblique and Gaussian, in combination with axis-orthogonal splits, we consider other orthogonal and oblique regression tree algorithms as baselines. The proposed Geospatial Regression Tree (GeoTree) is compared with a traditional Univariate Regression Tree (SkTree), and an Oblique Regression Tree (ObTree). The baseline methods are implemented using Scikit-learn's decision tree regressor [17] and a HHCart regression tree implementation [18].

To evaluate the models, we employ a repeated 5-fold set-up, where each fold is evaluated using five repeated experiments, resulting in 25 observations per model. We report the performance metrics on the test set among the five folds using the average of the five iterations. For all models, we use the MSE as the impurity measure. We set the hyperparameters of SkTree and ObTree to their default values. In GeoTree, we use a GA to generate oblique and Gaussian canidate splits. The parameter settings of the are shown in Table 1, which results from a grid search on a separate real-life housing data set with spatial features. The boundaries $(\underline{b}, \overline{b})$ are defined by the minimum and maximum values of the data.

Table 1
GA parameter settings employed for oblique and Gaussian candidate split generation in GeoTree.

Parameter	Value	Description
n_g	200	Number of generations
n_p	100	Population size
n_h	5	Hall of Fame size
n_t	10	Tournament size
$cxpb$	0.9	Crossover probability
α	0.05	Crossover extent
$mutpb$	0.2	Mutation probability of individuals
$indpb$	0.9	Mutation probability of chromosomes
η	0.2	Crowding degree of mutation



(a) orthogonal (b) diagonal (c) ellipse

(d) mixed (e) bivariate bumps

Figure 3: Synthetic data sets

4.2. Simulation Study

To assess the performance of GeoTree and the baselines in detecting patterns in data, we generated five synthetic data sets with distinct patterns (see Figure 3). The target variable in the different areas defined in the data sets is normally distributed with a varying mean. The results for the `orthogonal` data set shows that while GeoTree and SkTree can identify the true splits almost perfectly, ObTree struggles to do so. In contrast, GeoTree chooses correctly for orthogonal splits among the three split types and replicates the performance by SkTree. Similarly, GeoTree is effective in detecting the diagonal, elliptic, and mixed patterns in the `diagonal`, `ellipse` and `mixed` data sets. In addition, the experiments show that GeoTree can identify diagonal patterns more efficiently than ObTree. Finally, the `bivariate bumps` data set, a synthetic spatial regression data set adapted from [19], further confirms the superior performance of GeoTree in comparison to the baselines. In summary, the results show that GeoTree achieves lower error and requires smaller trees, thereby enhancing interpretability of the final model in comparison to traditional decision trees.

The data sets and results are discussed in more detail in appendix A.

4.3. Experiments on Housing and Soil Data

Two spatial applications, house price prediction and soil mapping, show the performance of the proposed method in a real-life setting. The results are discussed with regards to model accuracy, tree depth and size, and stability defined by the Interquartile Range (IQR), as well as the decision boundaries visualized in geographic space.

4.3.1. Housing data set

A real-life housing data set of the province of Namur in Belgium is used to test the performance of the proposed

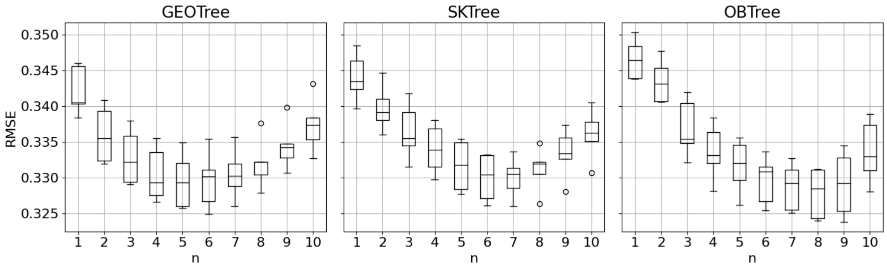
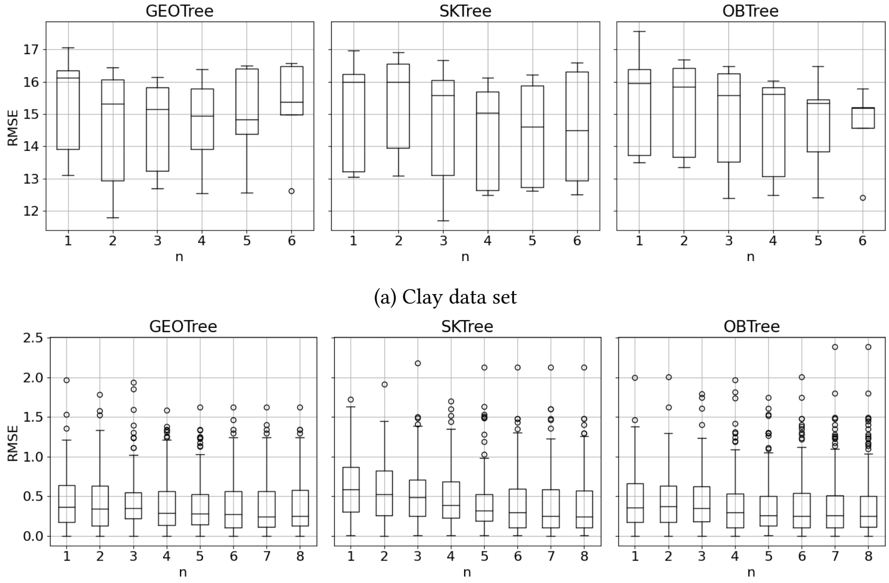


Figure 4: Box plots of test RMSE for the three models on the housing data set for different depths (n). The regression trees have a maximum depth of 10. The average of five repeated experiments is used for each of the five folds.



(a) Clay data set

(b) Meuse data set

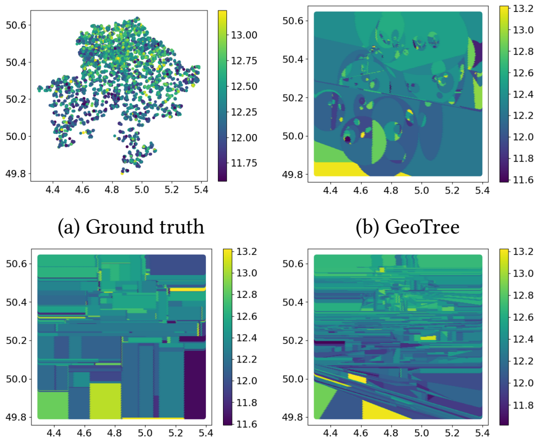
Figure 5: Box plots of test RMSE for the three models on the clay and Meuse data set for different depths (n). The regression trees have a maximum depth of 6 and 8 respectively.

model in comparison with the baseline methods. The data set contains 17 967 houses with the indexed transaction price and the X- and Y-coordinates. The house prices range between 100 000 and 600 000 euros approximately. For the experiments, the prices are log transformed and the X- and Y-coordinates are used as predictor variables. The test RMSE is presented in a box plot for all models in Figure 4. GeoTree reaches its lowest RMSE at depth 4 with a median RMSE of 0.3293. SkTree reaches a minimum RMSE of 0.3304 at depth 6, and ObTree reaches a value of 0.3285 at depth 8. The difference between the test errors of GeoTree and ObTree is not statistically significant (p-value = 0.1584), based on the corrected repeated k-fold cross-validation test [20]. These observations add to the evidence of the simulation study that the proposed GeoTree has an advantage in depth compared to the baseline methods. A smaller depth in turn increases interpretability. Moreover, these experiments show that the baseline methods have a similar test error. Table 2 also reveals that while the RMSE is similar among the three models, GeoTree is able to achieve this error level using significantly smaller trees indicated by the lower amount of leaf nodes. In addition, IQR of GeoTree is comparable to the IQR of the baseline methods, illustrated in Figure 4 by the box sizes. Although the box plots show more upward outliers for GeoTree than the baseline trees, this is only depths larger than the optimal depth. Therefore, we can conclude that GeoTree is as stable as the baseline methods.

Figure 6 presents the decision boundaries by the three

Table 2
Results of GeoTree and baselines for housing and soil data sets. Depth indicates the optimal depth, based on the median RMSE shown in boxplots. Tree size is defined by the average number of leaf nodes at the respective optimal depth among 5x5 folds and standard deviation. The RMSE column displays the average RMSE at the respective depth among 5x5 folds and standard deviation.

Data set	Model	Depth	Tree size	RMSE
Housing	SkTree	6	61.4 ± 1.74	0.3300 ± 0.00296
	ObTree	8	256 ± 0	**0.3278 ± 0.00317**
	GeoTree	4	**16 ± 0**	0.3305 ± 0
Clay	SkTree	6	61.4 ± 2.33	**14.5645 ± 1.67785**
	ObTree	6	64 ± 0	14.6245 ± 1.17688
	GeoTree	5	**28.6 ± 1.41**	14.9365 ± 1.52441
Meuse	SkTree	8	92.7 ± 2.19	0.3816 ± 0.3739
	ObTree	8	149.92 ± 0.996	0.3991 ± 0.4176
	GeoTree	7	**59.64 ± 3.31**	**0.3805 ± 0.3484**



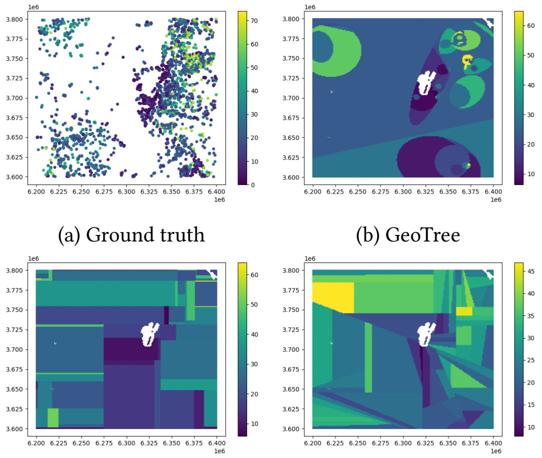
(a) Ground truth

(b) GeoTree

(c) SkTree

(d) ObTree

Figure 6: Plots of housing data with log transformed price in X-Y space (a) and decision boundaries of proposed and baseline methods on housing data (b-d).



(a) Ground truth

(b) GeoTree

(c) SkTree

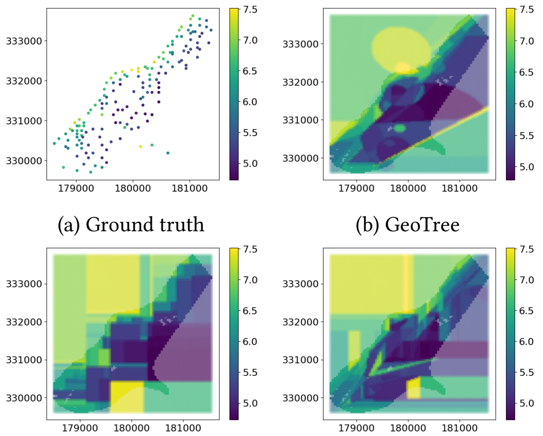
(d) ObTree

Figure 7: Plots of clay percentage data in X-Y space (a) and decision boundaries of proposed and baseline methods (b-d).

methods for the housing data set. While these figures show completely different patterns at first glance, the colors (predictions) in most areas are quite similar. For example, the lower left corner of the geographic map contains an area in bright green. In Figure 6b this area is odd-shaped, in Figure 6c it is rectangular shaped and a green triangle is visible in Figure 6d, according to the split types. Further, the decision boundaries of SkTree and ObTree are more similar, by dividing the space in many horizontally wide and vertically narrow rectangular regions. In contrast, GeoTree draws more ellipse patterns, with almost solely ellipses down the tree indicated by small ellipses. Nevertheless, larger areas are bounded by orthogonal, diagonal, and ellipse splits. This indicates that introducing ellipse shaped splits is beneficial, in particular, to capture fine-granular spatial patterns.

4.3.2. Clay data set

The first soil mapping data set contains clay percentage at 3418 locations around Lake Tahoe in California, USA, sourced from [21]. Figure 5a exhibits the RMSE on the test set with respect to different depths, with maximum depth 6, for the three decision trees. While all models show little progress in test error, GeoTree achieves its optimal depth at n = 5, and the baselines at n = 6, based on the median RMSE. The median RMSE at these depths is 14.83, 14.5 and 15.18 for GeoTree, SkTree, and ObTree respectively. The RMSE on the test set for GeoTree is not significantly different from SkTree (p-value = 0.753) and ObTree (p-value = 0.6486). The stability of the methods is



(a) Ground truth

(b) GeoTree

(c) SkTree

(d) ObTree

Figure 8: Plots of Meuse data with log transformed zinc concentrations in X-Y space (a) and decision boundaries of proposed and baseline methods for the Meuse prediction grid (b-d).

also comparable, indicated by similarly sized boxes, i.e., IQRs. Table 2 indicates as for the housing data that while the RMSE is similar, GeoTree needs on average less than half the number of leaf nodes than the baseline methods. This leads to the conclusion that the interpretability of our proposed method is boosted significantly in comparison to traditional decision trees.

Furthermore, the decision boundaries in Figure 7 also show much more intuitive patterns in geographic space for soil mapping. In contrast to straight-line separations between areas in the decision boundaries delineated by the baseline decision trees, GeoTree is able to present hotspots of high and low clay percentages in soil defined predominantly by ellipses. Again, some similarities can be found. For example, the decision boundaries in the upper right corner of the map show a yellow hotspot of clay percentage for GeoTree and ObTree. SkTree, on the other hand, seems to have found this pattern as well, but the same area is not as explored regarding the number of splits. Considering the rugged nature of geological phenomena, we conclude that the combination of elliptic splits with orthogonal and oblique splits are better suited to capture patterns in soil contents.

4.3.3. Meuse data set

Lastly, the proposed method is evaluated on one of the most widely used soil mapping data sets for spatial analysis, the Meuse data set [22]. This data set contains measurements of metals in soil around the Meuse river at 155 locations. As the number of observations is small, a leave-one-out cross-validation set-up is used instead of the 5-fold set-up. Similar to related work [1, 21], the target variable is the log transformed zinc concentration and X- and Y-coordinates are used as predictors. The RMSE on the test set is shown in Figure 5b for the three models with respect to depth n = 1 to 8. The median RMSE for GeoTree is lowest at depth 7 and slightly lower than the lowest median RMSE of the baselines, which is achieved at depth 8. Nevertheless, the variance of these error metrics is high, as can be seen in Figure 5b and Table 2, due to the small size of the data set. On average, GeoTree only requires around 60 leaf nodes to achieve a similar error level as the baselines, compared to 93 and 150 leaf nodes for SkTree and ObTree respectively (Table 2). Figure 8 shows the data and the decision boundaries of the methods for the area including the Meuse prediction grid. Although axis-parallel decision boundaries seem predominant even in Figure 8d, the ability to include Gaussian patterns makes the decision boundaries by GeoTree more easy to interpret and intuitive.

5. Conclusion

In this paper, we address the incompatibility of tree-based methods for spatial data by proposing a novel Geospatial Regression Tree algorithm (GeoTree). While decision trees are known for dividing the input space into rectangular regions, this feature is not always advantageous for spatial prediction, as spatial variables exhibit different patterns in reality. To address this issue, previous research has suggested replacing axis-parallel splits with axis-oblique splits. However, we propose a more advanced approach that combines bivariate oblique and Gaussian splits with axis-parallel splits. The combination of split types improves the regression tree's ability to capture geospatial patterns and introduces more intuitive and accurate decision boundaries. We rely on evolutionary algorithms to generate geospatial candidate splits and choose the best split among the three split types at each internal node. Our proposed algorithm is compared with Scikit-learn's univariate regression tree and the HH-Cart oblique tree using both synthetic data and real-life house price and soil mapping data. The results show that GeoTree outperforms the oblique decision tree on all synthetic data sets, particularly on the diagonal data set. Using real-life spatial applications, we have demonstrated that GeoTree achieves similar error metrics with less complex and more intuitive models. Furthermore, GeoTree models are more shallow and smaller in terms of leaf nodes, which increases interpretability, and allows for intuitive visualization of decision boundaries on a geographic map. We have also found that the stability of GeoTree is comparable to that of the baseline despite its non-analytical approach. In future work, we plan to improve GeoTree by including additional features, extending evaluation, and exploring the potential of ensemble learning. Overall, we believe that the proposed GeoTree algorithm has the potential to advance the field of spatial prediction and improve the interpretability and accuracy of models.

References

[1] A. B. Møller, A. M. Beucher, N. Pouladi, M. H. Greve, Oblique geographic coordinates as covariates for digital soil mapping, SOIL 6 (2020) 269–289. doi:10.5194/soil-6-269-2020.

[2] J. Gaudart, B. Poudiougou, S. Ranque, O. Doumbo, Oblique decision trees for spatial pattern detection: optimal algorithm and application to malaria risk, BMC Med. Res. Methodol. 5 (2005) 22. doi:10.1186/1471-2288-5-22.

[3] J. A. Goungounga, J. Gaudart, M. Colonna, R. Giorgi, Impact of socioeconomic inequalities on geographic disparities in cancer incidence: comparison of methods for spatial disease mapping, BMC Med. Res. Methodol. 16 (2016) 136. doi:10.1186/s12874-016-0228-x.

[4] Q. Gao, V. Shi, C. Pettit, H. Han, Property valuation using machine learning algorithms on statistical areas in Greater Sydney, Australia, Land Use Policy 123 (2022) 106409. doi:10.1016/j.landusepol.2022.106409.

[5] Z. Bassa, U. Bob, Z. Szantoi, R. Ismail, Land cover and land use mapping of the iSimangaliso Wetland Park, South Africa: comparison of oblique and orthogonal random forest algorithms, J. Appl. Remote Sens. 10 (2016) 015017. doi:10.1117/1.JRS.10.015017.

[6] J. Gaudart, N. Graffeo, D. Coulibaly, G. Barbet, S. Rebaudet, N. Dessay, O. K. Doumbo, R. Giorgi, SPODT: An R Package to Perform Spatial Partitioning, J. Stat. Softw. 63 (2015). doi:10.18637/jss.v063.i16.

[7] L. Canete-Sifuentes, R. Monroy, M. A. Medina-Perez, A Review and Experimental Comparison of Multivariate Decision Trees, IEEE Access 9 (2021) 110451–110479. doi:10.1109/ACCESS.2021.3102239.

[8] D. Wickramarachchi, B. Robertson, M. Reale, C. Price, J. Brown, HHCART: An oblique decision tree, Comput. Stat. Data Anal. 96 (2016) 12–23. doi:10.1016/j.csda.2015.11.006.

[9] Y. Dhebar, K. Deb, Interpretable rule discovery through bilevel optimization of split-rules of nonlinear decision trees for classification problems, IEEE Trans. Cybern. 51 (2020) 5573–5584. doi:10.1109/TCYB.2020.3033003.

[10] S.-C. Ng, K.-S. Leung, Induction of quadratic decision trees using genetic algorithms and k-d trees, WSEAS Trans. Comput. 3 (2004) 839–845.

[11] A. Magana-Mora, V. B. Bajic, OmniGA: Optimized Omnivariate Decision Trees for Generalizable Classification Models, Sci. Rep. 7 (2017) 3898. doi:10.1038/s41598-017-04281-9.

[12] J.-Y. Chang, C.-W. Cho, S.-H. Hsieh, S.-T. Chen, Genetic algorithm based fuzzy id3 algorithm, in: ICONIP, Springer, Berlin, 2004, pp. 989–995. doi:10.1007/978-3-540-30499-9_153.

[13] J. Yoo, L. Sael, Gaussian Soft Decision Trees for Interpretable Feature-Based Classification, in: PAKDD, Springer, Cham, 2021, pp. 143–155. doi:10.1007/978-3-030-75765-6_12.

[14] R. Rivera-Lopez, J. Canul-Reich, E. Mezura-Montes, M. A. Cruz-Chávez, Induction of decision trees as classification models through metaheuristics, Swarm Evol. Comput. 69 (2022) 101006. doi:10.1016/j.swevo.2021.101006.

[15] S. K. Murthy, S. Kasif, S. Salzberg, A System for Induction of Oblique Decision Trees, J. Artif. Intell. Res. 2 (1994) 1–32. doi:10.1613/jair.63.

[16] F.-M. De Rainville, F.-A. Fortin, M.-A. Gardner, M. Parizeau, C. Gagné, DEAP, in: GECCO, ACM Press, New York, USA, 2012, p. 85. doi:10.1145/2330784.2330799.

[17] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, others, Scikit-learn: Machine learning in Python, J. Mach. Learn. Res. 12 (2011) 2825–2830.

[18] ECNU, Oblique Decision Tree in Python, 2021. URL: https://github.com/zhenlingcn/scikit-obliquetree.

[19] M. H. Huque, H. D. Bondell, R. J. Carroll, L. M. Ryan, Spatial regression with covariate measurement error: A semiparametric approach, Biometrics 72 (2016) 678–686. doi:10.1111/biom.12474.

[20] R. R. Bouckaert, E. Frank, Evaluating the replicability of significance tests for comparing learning algorithms, in: PAKDD, Springer, Berlin, 2004, pp. 3–12.

[21] T. Hengl, M. Nussbaum, M. N. Wright, G. B. Heuvelink, B. Gräler, Random forest as a generic framework for predictive modeling of spatial and spatio-temporal variables, PeerJ 2018 (2018). doi:10.7717/peerj.5518.

[22] E. J. Pebesma, R. S. Bivand, Classes and methods for spatial data in R, R News 5 (2005) 9–13. URL: https://CRAN.R-project.org/doc/Rnews/.

A. Simulation Study

A.1. Data sets

We have generated five synthetic data sets to demonstrate the geospatial patterns that GeoTree can detect more accurately (see Figure 3). The data set generation process starts in all five cases by uniformly creating data points in a two-dimensional space. All data sets contain 4096 data points. The target variable generation differs for each data set. The first, `orthogonal` data set contains two orthogonal boundaries through the middle of the first and second axis, dividing the two-dimensional space in four equal parts (see Figure 3a). The y-values are drawn from $\mathcal{N}(10, 2.0^2)$, $\mathcal{N}(20, 2.0^2)$, $\mathcal{N}(30, 2.0^2)$ and $\mathcal{N}(40, 2.0^2)$. For the second data set, the `diagonal` data set, four points in the two-dimensional space are selected to generate two parallel hyperplanes (see Figure 3b). The three respective areas separated by these hyperplanes are assigned y-values from $\mathcal{N}(40, 2.0^2)$, $\mathcal{N}(10, 2.0^2)$ and $\mathcal{N}(20, 2.0^2)$. The third, `ellipse` data set consists of an ellipse shape created by three random points in the input space (see Figure 3c). The points inside the ellipse are assigned a y-value from $\mathcal{N}(30, 2.0^2)$, while the y-values for the points outside of the ellipse are drawn from $\mathcal{N}(10, 2.0^2)$. In the fourth, `mixed` data set a hyperplane connecting two randomly sampled points divides the space in two areas (see Figure 3d). From the points above the hyperplane, three points are randomly sampled to generate an ellipse. The y-values in the three resulting areas are drawn from $\mathcal{N}(40, 2.0^2)$, $\mathcal{N}(10, 2.0^2)$ and $\mathcal{N}(20, 2.0^2)$. Lastly, we calculate the y-values based on the bivariate bump function from [19], a study in which it was used previously for spatial regression. The y-value for each point i is calculated as $y_i = s_{i1} \cdot s_{i2}$, with

$$s_{ij} = \frac{1}{1 + x_{ij}} + \left(5.5 \cdot e^{-50 \cdot \left(x_{ij} - 0.2\right)^2}\right) + \left(5 \cdot e^{-25 \cdot \left(x_{ij} - 0.8\right)^2}\right)$$

for j = 1, 2 and x_{i1} and x_{i2} the coordinates of point i. The resulting y-values for the `bivariate bumps` data set range between 1 and 40 (see Figure 3e).

A.2. Results for `orthogonal`, `diagonal`, `ellipse`, `mixed`, and `bivariate bumps`

Figure 9 presents the error metrics of GeoTree, SkTree and ObTree on the `orthogonal` data set. The decision trees have depth 2 which is the optimal depth for this data set. As the data set contains four areas with different target values, a decision tree should be able to recreate this pattern with four leaf nodes. These box plots show that while GeoTree can replicate SkTree's performance, ObTree's errors are much higher. The decision boundaries of the three models on the `orthogonal` data set also reveal the perfect ability of GeoTree and SkTree to find the orthogonal patterns, in contrast to ObTree.

The test RMSE of the three models is presented in box plots as shown in Figure 10 for the `diagonal`, `ellipse` `mixed` and `bivariate bumps` data set. The maximum depth is set for all methods dependent on the data set, 8 for the `diagonal` data set, 9 for the `ellipse` and `mixed` data sets. GeoTree performs significantly better than the baseline trees on the `diagonal` data set (see Figure 10a). Not only can the GA-based tree find the diagonal patterns with three splits (i.e. depth 2), the median RMSE (2.0066) is considerably lower at this depth than the minima that SkTree (3.2372) and ObTree (3.2788) achieve at reasonable depth. What is apparent is that ObTree does not find the diagonal boundaries with a shallow tree, despite the ability to learn axis-oblique splits. While ObTree's test error continues to improve at depth 9, SkTree achieves a minimum RMSE at depth 8.

For the `ellipse` data set, GeoTree's error starts considerably lower at depth 1 than the baseline methods and continues this trend across depths. This is due to the first split where GeoTree is able to draw an ellipse in the input space, replicating the ellipse pattern in the data set. While a decreasing trend in SkTree's and ObTree's errors is visible in Figure 10b, they do not reach a similar level. In addition, GeoTree's Interquartile Range (IQR) is smaller than the baseline methods as of depth 2.

Figure 10c shows a similar trend as the previous plots, GeoTree's error declines rapidly, while the decline in the error of the baseline methods is more gradual. In addition, SkTree's and ObTree's errors do not reach the same level as GeoTree's error. GeoTree reaches the lowest median test RMSE at depth 4 (2.4685), whereas SkTree reaches a test RMSE of 3.6054 at depth 8 and ObTree's error ends at 4.267.

Lastly, Figure 10d provides the results of the three methods trained on the `bivariate bumps` data set with depth 30. What stands out in this figure is that, in contrast to the other methods, ObTree can not improve the train error after depth 14 and therefor stops the training process. Nevertheless, the lowest median test RMSE is 1.72 at depth 11, compared to 0.75 and 1.16 for GeoTree and SkTree respectively. The depths where GeoTree and SkTree reach these values are considerably higher (29 and 24). However, at depth 11, where ObTree reaches a minimum, GeoTree's median test RMSE is 1.31, and at depth 24, where SkTree reaches a minimum, GeoTree has a value of 0.75. ObTree's IQR of the test error is also considerably higher than the other methods.

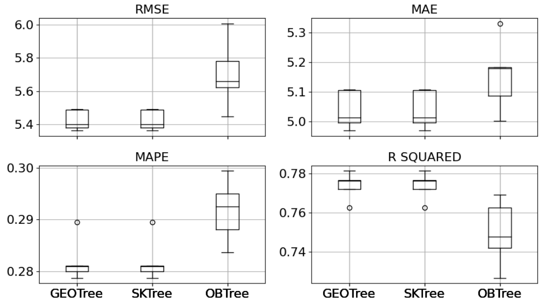
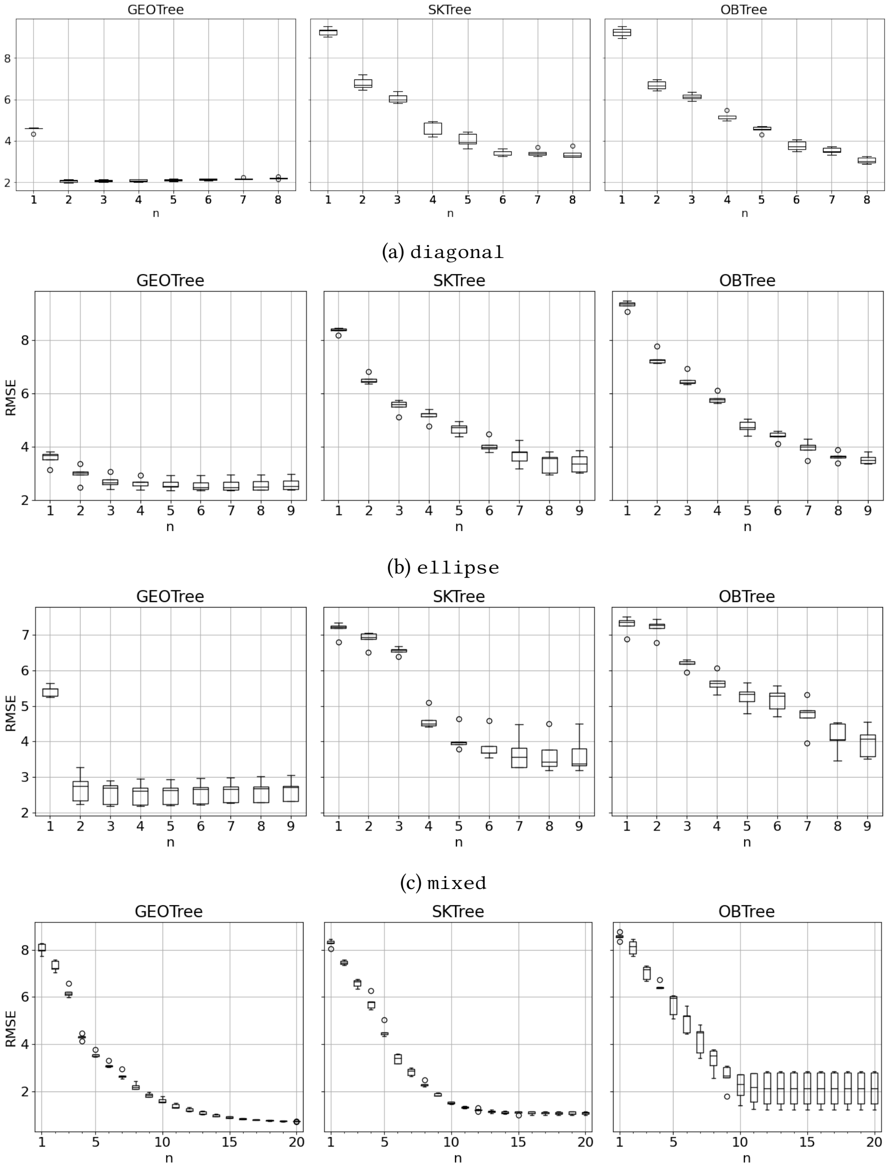


Figure 9: Box plots of RMSE, MAE, MAPE and r^2 on the `orthogonal` data set for the three models at depth 2. The average metric on the test set of five repeated experiments is used for each of five folds.

A.3. Evaluation

The experiments on the synthetic data sets confirm GeoTree's ability to efficiently detect orthogonal, diagonal and ellipse patterns. Figure 9 and Figure 10 show that GeoTree finds the patterns in fewer splits and reaches a substantially lower error on the test set with reasonable depth than the baseline methods SkTree and ObTree. The performance on the `mixed` data set also shows that GeoTree can correctly combine the different split types. At the same time, the proposed model can perfectly replicate a traditional univariate tree's performance on a data set with orthogonal patterns. Notably, GeoTree also outperforms ObTree on the `diagonal` data set with a substantial difference. The `bivariate bumps` data set allows to confirm GeoTree's superior performance on more advanced rounded patterns compared to the baseline methods. Though GeoTree requires a deeper tree to reach the minimum test RMSE, the error is considerably lower. In addition, the proposed method has a lower median test RMSE than the baseline methods at the depth where they reach a minimum. This indicates that GeoTree still has an advantage in depth for the `bivariate bumps` data set as for the other synthetic data sets. Despite the added variability in candidate split generation due to the GA, GeoTree's stability is superior to the baseline methods in most cases illustrated by the IQR of the test error.



(a) diagonal

(b) ellipse

(c) mixed

(d) bivariate bumps

Figure 10: Box plots of test RMSE for the three models on different data sets for different depths (n). The trees have a maximum depth of 8, 9, 9 and 30 for the diagonal, ellipse, mixed and bivariate bumps data set respectively. The average of five repeated experiments is used for each of five folds.