INTERPRETABLE IMAGE CLASSIFICATION USING SPARSE OBLIQUE DECISION TREES

Suryabhan Singh Hada *Miguel Á. Carreira-Perpiñán*

Department of CSE, University of California, Merced
{shada, mcarreira-perpinan}@ucmerced.edu

ABSTRACT

Interpreting the image datasets is a difficult task, as each image contains a lot of irrelevant data. This paper presents a simple yet effective method to interpret the image datasets. We achieve this by using sparse oblique trees as a tool to select features from the dataset. These trees are not only accurate but also very interpretable. The hierarchical structure of the tree helps to visualize the underlying patterns in the dataset. By studying the weights of the nodes, we can determine what set of features differentiate between classes or groups of classes. We effectively demonstrate our results in multiple image datasets.

Index Terms— interpretability, image classification, decision trees

1. INTRODUCTION

In the last decade, several advancements have been made in machine learning, especially in computer vision tasks. A lot of this success has been attributed to deep learning, with deeper and deeper neural networks. Although a lot of this success is due to the intelligent design of these networks, the presence of better computing devices such as GPUs and, most importantly, the larger datasets play an important role too. Current state-of-the-art models can perform good in terms of performance. However, they do not present any information regarding the dataset. This leads to the many problems, a model can learn certain biases, decreasing trust in the model. This has lead to the development of many methods that can provide some explanation about the model decision. These methods do provide some explanation, but they are restricted to provide information only about the model's prediction behavior [1, 2, 3] or what information is encoded by model's parameters [4, 5, 6, 7, 8]. These methods are unable to provide information about the data itself.

We can summarize small tabular datasets, but understanding the bigger datasets is difficult. For example, in image datasets, it is challenging to understand a given class's basic concept. Since there is so much irrelevant information, it is hard to understand what part of the image is important or what common concept defines a particular category of the class. Feature selection methods (Chi-Squared [9], Mutual Information [10], LASSO [11], and Kolmogorov-Smirnov statistic [12]) can help to understand these datasets. However, they leave a lot of questions unanswered. For example in classification task: we do not know how the classes are distributed in the input space; is one class closer to another class, or how two classes or group of classes differentiate with each other; or if there is any subgroups exist in a given class, if yes what is the difference them; or the selected features are optimal for a class, or sub-group of a class, or even a single instance?

In this work, we address these issues by using sparse oblique trees as a tool to understand the given dataset. The hierarchical-based structure allows us to understand the relationship between classes, and the sparsity helps us capture only a small number of features. For decades trees have been considered as the widely interpretable models. But these trees have been restricted to only axis-aligned trees [13, 14]. This is due to the lack of better training algorithms for oblique trees. Axis-aligned trees are interpretable only for a small dataset, but as the dataset size increases, their size grows by a large margin, making them challenging to interpret (table: 2). On the other side, an ensemble of these trees like random forest [15] perform well, but they are black-box [16], so they are not interpretable. Here, we rely on a recently proposed algorithm called *Tree Alternating Optimization (TAO)* algorithm [17]. TAO can learn far more accurate oblique trees that remain small and very interpretable. Unlike axis-aligned trees that operate only on a single feature at each node, the oblique tree operates on a small, learnable subset of features. Also, unlike traditional tree algorithms such as CART [18] or C4.5 [19], TAO monotonically decreases the objective function containing classification error and sparsity penalty on the weights of each node. It has been shown to outperform existing tree algorithms by a large margin [20], and to improve forests [21, 22, 23, 24]. Counterfactual explanations can also be solved exactly for these trees [25, 26, 27]. Finally, the stronger predictive power of sparse oblique decision trees together with their interpretability and fast inference makes them useful for other uses, such as in understanding deep neural networks [28, 29] or compressing deep neural networks [30].

Next, we describe our approach to interpret the dataset using spare oblique trees, and then we describe the dataset, and finally we demonstrate our approach on multiple datasets for image classification task.

2. PROPOSED APPROACH

Our approach is as follows:

1. Train a sparse oblique tree using TAO [17] and pick the sparsity parameter such that the resultant tree is as sparse as possible but remains accurate enough.
2. Use the weights of decision nodes to extract relevant features from the dataset.

For a given input $\mathbf{x} \in \mathbb{R}^d$ we define the decision rule at a decision node i as follows: "if $\mathbf{w}_i^T\mathbf{x} + b_i \geq 0$, then go to right child, else go to the left child", where $\mathbf{w} \in \mathbb{R}^d$ is the weight vector of node i and $b_i \in \mathbb{R}$ is the bias. Now to extract the features we apply the following operation:

Write $\mathbf{w}$ and $\mathbf{x}$ as $\mathbf{w} = (\mathbf{w}_0\ \mathbf{w}_-\ \mathbf{w}_+)$ and $\mathbf{x} = (\mathbf{x}_0\ \mathbf{x}_-\ \mathbf{x}_+)$, where $\mathbf{w}_0 = \mathbf{0}$, $\mathbf{w}_- < \mathbf{0}$ and $\mathbf{w}_+ > \mathbf{0}$ contain the zero, negative and positive weights in $\mathbf{w}$, and $\mathbf{x}$ is arranged accordingly. Call $\mathcal{S}_0$, $\mathcal{S}_-$ and $\mathcal{S}_+$ the corresponding sets of indices in $\mathbf{w}$. Now, if $\mathbf{x}$ goes to the right, we represent the feature selected as a binary vector $\mu^+ \in \{0,1\}^d$, containing ones only at $\mathcal{S}_+$. Similarly, if $\mathbf{x}$ goes to the left binary vector $\mu^- \in \{0,1\}^d$, containing ones only at $\mathcal{S}_-$. We call

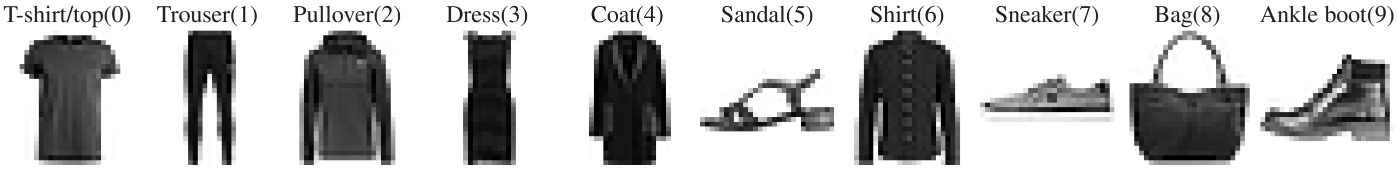

Fig. 1. Single instance of each class in Fashion-MNIST dataset.

Model	Class(Fashion-MNIST, 784 features)										Total
	T-shirt/top(0)	Trouser(1)	Pullover(2)	Dress(3)	Coat(4)	Sandal(5)	Shirt(6)	Sneaker(7)	Bag(8)	Ankle boot(9)	
TAO	154	166	283	153	293	155	157	299	177	163	440
	Class(MNIST, 784 features)										
	0	1	2	3	4	5	6	7	8	9	
TAO	254	164	329	239	290	212	224	260	293	201	377
	Class(LeNet5 features, 800 features)										
	0	1	2	3	4	5	6	7	8	9	
TAO	91	173	96	171	138	96	114	191	171	138	395
	Class(VGG16 features, 8192 features)										
	0	1	2	3	4	5	6	7	8	9	
TAO	352	444	572	710	554	544	836	435	440	426	2730
	10	11	12	13	14	15					
	422	419	439	406	451	746	-	-	-	-	
	Class(Segment, 19 features)										
	0	1	2	3	4	5	6	7	-	-	
TAO	9	7	11	11	290	8	8	9	-	-	14

Table 1. Number of feature selected by the sparse oblique tree for different dataset.

μ^+ and μ^- the NODE-FEATURES, where location of one represents features selected by $\mathbf{w}$.

We use NODE-FEATURES, to interpret the dataset as follows:

1. For each decision node NODE-FEATURES represents the features related to left and right subtree. By using NODE-FEATURES, we can understand what set of features separate a group of classes.
2. Features associated with a class k: for each node in the path from the root to leaf for class k collect NODE-FEATURES, and at the end take logical OR of all NODE-FEATURES. If there is more than one leaf for class k, take the union of all the features selected.
3. For features specific to a given an input $\mathbf{x}$: Repeat the process as above, but only for the leaf containing the input $\mathbf{x}$. Next, keep only those features that are active in the $\mathbf{x}$.

We can plot these features to visualize what concept is captured by these features.

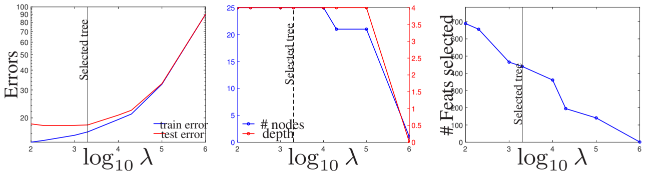

Fig. 2. Classification error (training and test) and number of nodes and of features selected by the trees as a function of λ for Fashion-MNIST dataset. The vertical line indicates the tree we selected as mimick ($\lambda = 2000$).

3. DATASETS DESCRIPTION

In this work, we use three types of image datasets. The first category is the image dataset where input features are raw pixels: Fashion-MNIST [31] and MNIST, where each input is in $\mathbb{R}^{784}$ space. The second category is the deep neural network features. For LeNet5 [32] features, we extract features from the last convolution layer of a pre-trained LeNet5 trained on MNIST. Each input in this dataset is $\mathbb{R}^{800}$ space. Next, for VGG features, we use features from a pre-trained VGG16 [33] network trained on a subset of 16 classes from the Imagenet dataset [34]. Again we use features from the last convolution layer of the network, and each input is in $\mathbb{R}^{8192}$ space. The final category is the hand-crafted features; here, we use the Segment dataset. It is as UCI [35] dataset where the instances were drawn randomly from a database of 7 outdoor images. The images were hand-segmented to create a classification for every pixel, and each input is in $\mathbb{R}^{19}$ space.

Here, we demonstrate our results in detail on Fashion-MNIST [31] (fig. 1), and summarize other datasets in the table 2 and 1.

4. EXPERIMENTS

For each dataset, we choose an initial tree structure for TAO of a certain depth and random initial values for the weights at the nodes. The decision nodes are hyperplanes, and each leaf contains a single class label. We constructed a collection of trees over a range of sparsity parameter $\lambda \in [0, \infty)$. From these trees, we pick the tree that is accurate as well as sparse enough for interpretability. In table 2 we report the results of these trees. In the same table, we also report the best accuracy, in which case the tree is deeper and less sparse. Although, we would not use those trees for interpretability.

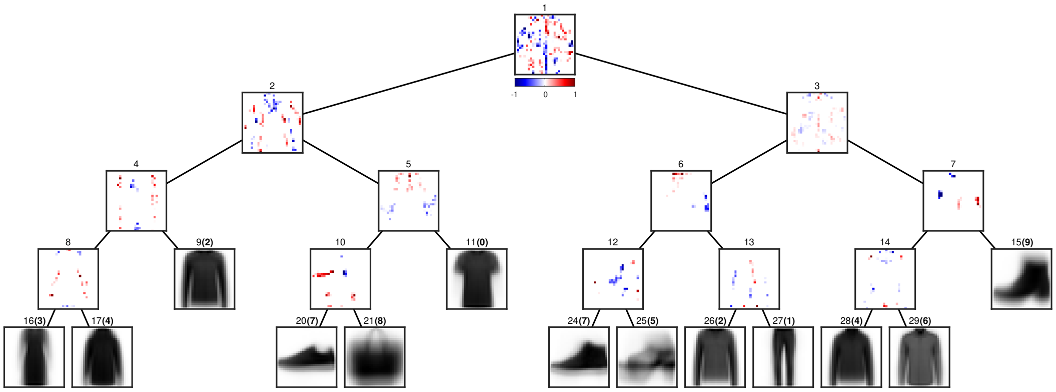

Fig. 3. Tree selected to interpret the Fashion-MNIST dataset ($\lambda = 2000$). At each decision node, we show its weight vector and node index. At each leaf, we show their index, class label, and the average of training instances reaching the leaf. We plot both the weight vector and the average, of dimension 784, as a 28×28 square. Weight vectors are colored according to their sign and magnitude (positive, negative and zero values are blue, red, and white, respectively), and averages as greyscale.

Model	Training Error	Test Error	Height	# Nodes
		Fashion MNIST		
TAO(best)	5.88	14.11	8	211
TAO	16.84	18.08	5	25
CART	2.14	20.88	30	9443
		MNIST		
TAO(best)	1.57	5.26	8	177
TAO	9.23	9.58	6	39
CART	0.42	12.26	20	6857
		LeNet5 features		
TAO(best)	0.02	1.78	6	166
TAO	2.35	2.65	6	23
CART	0.31	6.56	20	3165
		VGG features		
TAO(best)	0	7.62	6	51
TAO	2.35	2.65	6	39
CART	0.31	6.56	20	3165
		Segment		
TAO	3.23	3.57	6	27
CART	0.05	5.62	14	129

Table 2. Training and test errors, and parameter for CART and TAO on different datasets. If TAO(best) is present in a row, it represents best tree for classification, but we do not use that tree for interpretability.

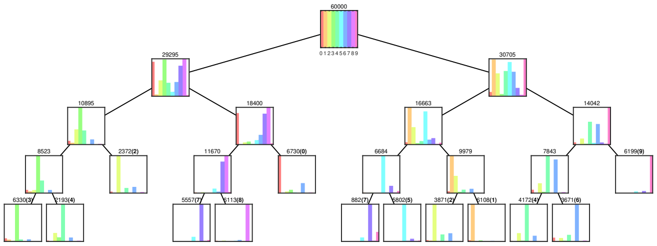

Fig. 4. Class histograms; we show the number of training instances reaching the node and, for leaves, their label. You may need to zoom in the plot.

As mentioned above in this work, we present our approach in detail for the fashion-MNIST dataset and summarize results for other datasets in table 1. For the Fashion-MNIST dataset, we use the initial tree of depth 5 (total 31 nodes), and train it over a range of sparsity parameter λ as mentioned above. In fig. 2 we show the training/test errors as a function of sparsity parameters. Using these plots, we pick the tree with the lowest test error ($\lambda = 2000$). In fig. 3, we show the weight vector of each decision node, and in fig. 4 we show the histogram of instances at each node.

We also trained a CART tree for each dataset, and as mentioned earlier, as the dataset becomes complicated, the CART tree becomes really difficult to interpret due to the large number of nodes as shown in the table: 2.

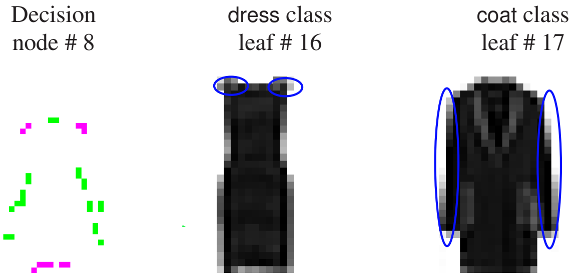

Fig. 5. *Left:* weight vector of the decision node # 8 plotted as 28×28 image. Magenta color represents negative weights (left child) and green color represents positive weights (right child). *Middle:* one instance of class dress, and blue ellipses show the location of magenta color weights in the image. *Right:* one instance of class coat, and blue ellipses show the location of green color weights.

4.1. Sub-groups in Sneakers and pullover class

Determining sub-groups within a given class is very important to interpret the dataset. As shown in the fig. 3, there exist sub-groups within a single class. For instance, both in class sneakers and pullover, there exist two sub-groups in each class. In class sneakers by looking at the average image in the leaves (number 20 and 24), it is easy to see two types of sneakers in this class. These subgroups are different in terms of the height of the ankle part of the shoe. Since there are so many instances, it is difficult to predict this kind of sub-groupings. However, the sparse oblique trees allow us to

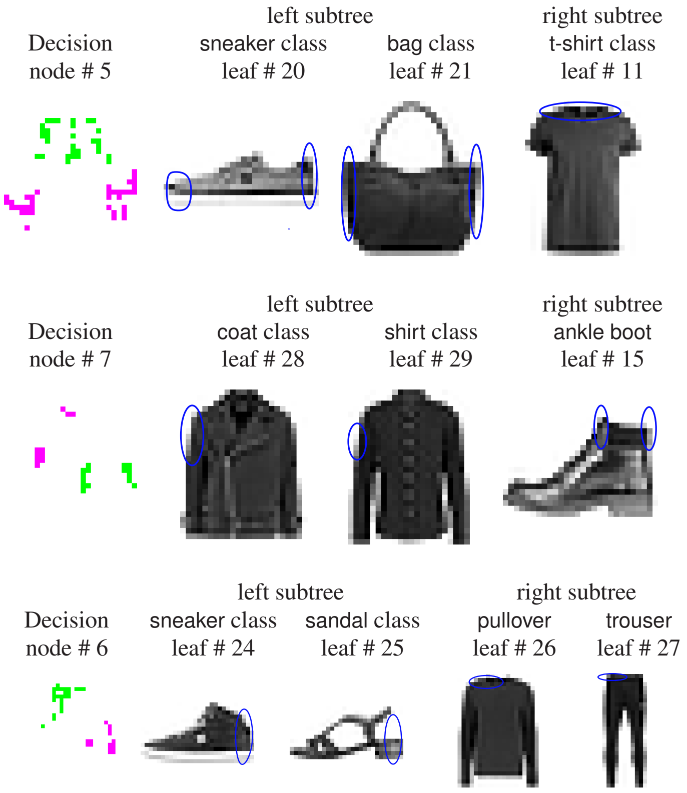

Fig. 6. Same as fig. 5, but for a group of classes.

visualize these groups easily.

4.2. Difference between pair of classes or group of classes

There is a lot of irrelevant information in a given image in the image dataset, making it hard to understand what concept differentiates two classes. So, it is very difficult to summarize the difference between classes by just looking at the images. However, by examining the weights of the two classes' decision nodes, we can understand what concept separates the two classes. For instance, as shown in the fig. 5 the weights of decision node number 8 clearly describe the difference between class dress and coat. The tree mostly focuses on the sleeves; if it is present, then it is a coat (green weights); otherwise, it is a dress. This makes sense when we look at the instance of the dress class where most instances do not have long sleeves, unlike to coat class where all instances have long sleeves.

Similarly, by examining the decision nodes closer to the root, we can determine the difference between a group of classes. We show few such examples in fig. 6. As shown in the fig. 3 node # 5 separates class sneaker and bag to the left side, tshirt to the right side. The decision node separates tshirt from other two, based on the presence of ink on the *neck*, as both bag and sneaker do not have ink on that region. In the second row the decision node # 7 separates the left group (coat and shirt) from the right group (ankle boots) in terms of ink present at the top of the ankle boots. If the ink is present at that location, the node sends the instance to the right child; otherwise, the left child. In third row of the fig. 6 we see one interesting example, where the tree needs very few pixels to differentiate between two groups: Sneaker and sandals on the left, and pullover and trouser on the right. The node checks if the instance has *collar* or the *belt*, then the instance goes to the right side; otherwise, if the instance has ink at the *right side of shoe* then it goes to the left. These insights about the data are difficult to attain with any other classifier.

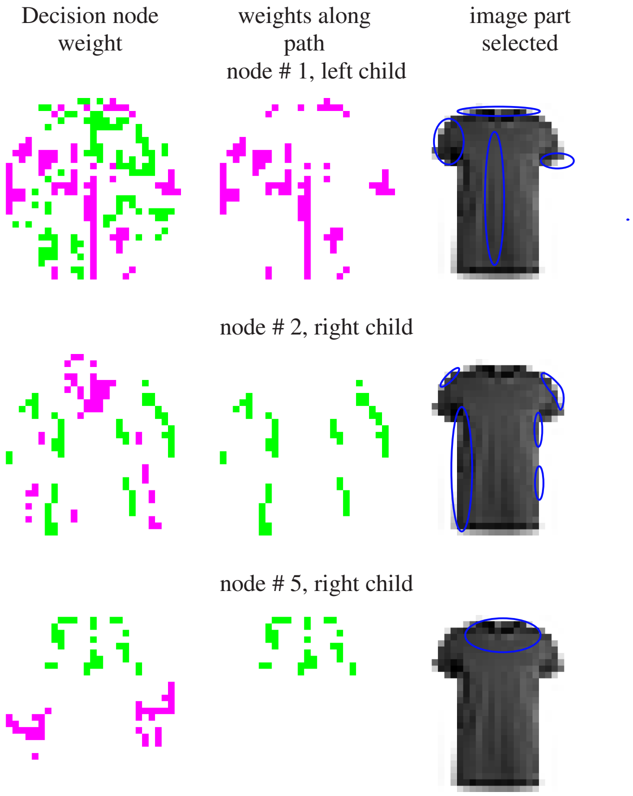

Fig. 7. *Left:* weight vectors of decision nodes plotted as 28×28 image. Magenta color represents negative weights (left child), and green color represents positive weights (right child). *Middle:* weights along the path from the root to leaf # 11. *Right:* one instance of class t-shirt, and blue ellipses show the location of weights from the middle column in the image.

4.3. Feature selection for a given instance

The proposed approach can select features that are specific to a given instance. This way we can track what part of images is selected at each decision node. This provides a if-else based rule for a given prediction. Thus provide us a set-up of rules to understand what features are required for a given instances to be classified as a specific class.

We give one such example for an instance of class t-shirt in fig. 7. As shown in the at first node the tree focuses on the *neck*, left *short sleeves* and the middle pixels in the image. Then for the second node in the path (node # 2), the tree checks for the shape of the the t-shirt to separate it from the other classes (*dress*, *coat*, *pullover*). Finally, the at node # 5 the tree use *neck* as the discriminatory features to send the instance at leaf # 11.

5. CONCLUSION

In this work, we introduce sparse oblique trees as a tool to interpret image datasets. We show that how sparse oblique trees can be used as an accurate yet interpretable model. We also propose an approach to use the weights of the decision nodes to explain the underlying difference between classes, sub-group of a class, or group of classes. Using the hierarchical structure of the oblique tree, we can extract features that are tailored not only to class but also for specific instances.

6. REFERENCES

[1] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin, ""Why should I trust you?": Explaining the predictions of any classifier," in *SIGKDD 2016*.

[2] W. James Murdoch, Peter J. Liu, and Bin Yu, "Beyond word importance: Contextual decomposition to extract interactions from LSTMs," in *ICLR 2018*.

[3] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba, "Learning deep features for discriminative localization," in *CVPR 2016*.

[4] Dumitru Erhan, Yoshua Bengio, Aaron Courville, and Pascal Vincent, "Visualizing higher-layer features of a deep network," Tech. Rep. 1341, Université de Montréal, June 2009.

[5] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," in ICLR 2014.

[6] Anh Nguyen, Alexey Dosovitskiy, Jason Yosinski, Thomas Brox, and Jeff Clune, "Synthesizing the preferred inputs for neurons in neural networks via deep generator networks," in *NIPS 2016*.

[7] Suryabhan Singh Hada and Miguel Á. Carreira-Perpiñán, "Sampling the "inverse set" of a neuron: An approach to understanding neural nets," arXiv:1910.04857, Sept. 27 2019.

[8] Suryabhan Singh Hada and Miguel Á. Carreira-Perpiñán, "Sampling the "inverse set" of a neuron," in *ICIP 2021*.

[9] George Forman, "An extensive empirical study of feature selection metrics for text classification," *J. Machine Learning Research*, vol. 3, pp. 1289–1305, Mar. 2003.

[10] Hanchuan Peng, Fuhui Long, and Chris Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 27, no. 8, pp. 1226–1238, Aug. 2005.

[11] Trevor Hastie, Robert Tibshirani, and Martin Wainwright, *Statistical Learning with Sparsity: The Lasso and Generalizations*, Monographs on Statistics and Applied Probability. Chapman & Hall/CRC, 2015.

[12] David J Dittman, Taghi M Khoshgoftaar, Randall Wald, and Jason Van Hulse, "Comparative analysis of dna microarray data through the use of feature selection techniques," in *ICMLA 2010*.

[13] Chid Apte, Fred Damerau, and Sholom Weiss, *Text mining with decision rules and decision trees*, Citeseer, 1998.

[14] Alya Al Nasseri, Allan Tucker, and Sergio de Cesare, "Quantifying StockTwits semantic terms' trading behavior in financial markets: An effective application of decision tree algorithms," *Expert systems with applications*, vol. 42, no. 23, pp. 9192–9210, 2015.

[15] Leo Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, Oct. 2001.

[16] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi, "A survey of methods for explaining black box models," *ACM Computing Surveys*, vol. 51, no. 5, pp. 93, May 2018.

[17] Miguel Á. Carreira-Perpiñán and Pooya Tavallali, "Alternating optimization of decision trees, with application to learning sparse oblique trees," in *NEURIPS 2018*.

[18] Leo J. Breiman, Jerome H. Friedman, R. A. Olshen, and Charles J. Stone, *Classification and Regression Trees*, Wadsworth, Belmont, Calif., 1984.

[19] J. Ross Quinlan, *C4.5: Programs for Machine Learning*, Morgan Kaufmann, 1993.

[20] Arman Zharmagambetov, Suryabhan Singh Hada, Magzhan Gabidolla, and Miguel Á. Carreira-Perpiñán, "Non-greedy algorithms for decision tree optimization: An experimental comparison," in *IJCNN 2021*.

[21] Miguel Á. Carreira-Perpiñán and Arman Zharmagambetov, "Ensembles of bagged TAO trees consistently improve over random forests, AdaBoost and gradient boosting," in *FODS 2020*.

[22] Arman Zharmagambetov and Miguel Á. Carreira-Perpiñán, "Smaller, more accurate regression forests using tree alternating optimization," in *ICML 2020*.

[23] Arman Zharmagambetov, Magzhan Gabidolla, and Miguel Á. Carreira-Perpiñán, "Improved boosted regression forests through non-greedy tree optimization," in *IJCNN 2021*.

[24] Arman Zharmagambetov, Magzhan Gabidolla, and Miguel Á. Carreira-Perpiñán, "Improved multiclass AdaBoost for image classification: The role of tree optimization," in *ICIP 2021*.

[25] Miguel Á. Carreira-Perpiñán and Suryabhan Singh Hada, "Counterfactual explanations for oblique decision trees: Exact, efficient algorithms," in *AAAI 2021*.

[26] Miguel Á. Carreira-Perpiñán and Suryabhan Singh Hada, "Counterfactual explanations for oblique decision trees: Exact, efficient algorithms," arXiv:2103.01096, Mar. 1 2021.

[27] Suryabhan Singh Hada and Miguel Á. Carreira-Perpiñán, "Exploring counterfactual explanations for classification and regression trees," in *XKDD 2021*.

[28] Suryabhan Singh Hada, Miguel Á. Carreira-Perpiñán, and Arman Zharmagambetov, "Sparse oblique decision trees: A tool to understand and manipulate neural net features," arXiv:2104.02922, Apr. 7 2021.

[29] Suryabhan Singh Hada, Miguel Á. Carreira-Perpiñán, and Arman Zharmagambetov, "Understanding and manipulating neural net features using sparse oblique classification trees," in *ICIP 2021*.

[30] Yerlan Idelbayev, Arman Zharmagambetov, Magzhan Gabidolla, and Miguel Á. Carreira-Perpiñán, "Faster neural net inference via forests of sparse oblique decision trees," arXiv, 2022.

[31] Han Xiao, Kashif Rasula, and Roland Vollgraf, "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms," arXiv:1708.07747, 15 2017.

[32] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[33] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," in *ICLR 2015*.

[34] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *CVPR 2009*.

[35] C. L. Blake and C. J. Merz, "UCI repository of machine learning databases," 1998.