Interpretable Image Classification Using Sparse Oblique Decision Trees

Suryabhan Singh Hada   Miguel Á. Carreira-Perpiñán
{shada,mcarreira-perpinan}@ucmerced.edu

Dept. of Computer Science and Engineering,
University of California, Merced
http://eecs.ucmerced.edu

April 20, 2022

Introduction

- Interpreting the image datasets is a difficult task, as each image contains a lot of irrelevant data. This makes it hard to understand what part of the image is important or what common concept defines a particular category of the class.
- 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 sub-groups 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?
- We address these issues by using sparse oblique trees as a tool to understand the given image dataset.

Sparse 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.
- Unlike axis-aligned trees that operate only on a single feature at each node, the sparse oblique tree operates on a small, learnable subset of features.
- Sparse oblique trees are not only accurate but also very interpretable.

Tree alternating optimization (TAO)

- Traditionally, decision trees have been trained with a recursive partition procedure, such as CART and C4.5. However, this produces sub-optimal trees and does not work well with oblique trees.
- Tree alternating optimization (TAO) [1] is a recently proposed algorithm that can achieve highly accurate oblique or axis-aligned trees.
- TAO can learn far more accurate oblique trees that remain small and very interpretable.
- TAO has been shown to outperform existing tree algorithms by a large margin [2], and to improve forests [3].

Our approach

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

Extracting features using weights of decision nodes

- Consider a decision node i in T with decision rule:
  - "if $\mathbf{w}_i^T\mathbf{x} + b_i \geq 0$ then go to right child, else go to left child"
    - $\mathbf{w}_i \in \mathbb{R}^D$ is the weight vector.
    - $b_i \in \mathbb{R}$ is the bias.
    - $\mathbf{x} \in \mathbb{R}^D$ is the input.
- Write $\mathbf{w}$ as: $\mathbf{w} = \begin{pmatrix}\mathbf{w}_0 & \mathbf{w}_- & \mathbf{w}_+\end{pmatrix}$
  - $\mathbf{w}_0 = 0$.
  - $\mathbf{w}_- < 0$.
  - $\mathbf{w}_+ > 0$.
- Call $\mathcal{S}_0$, $\mathcal{S}_-$ and $\mathcal{S}_+$ the corresponding sets of indices in $\mathbf{w}$.

Extracting features using weights of decision nodes

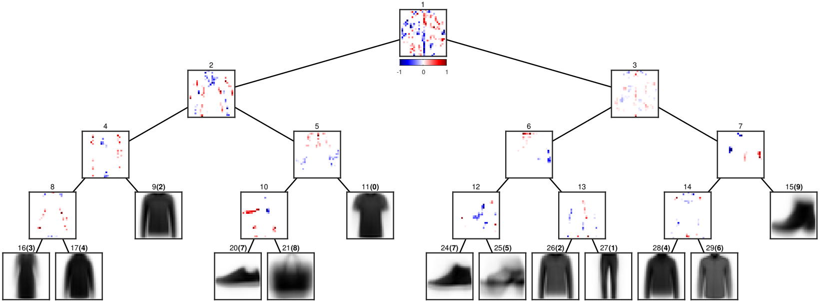
- Consider input $\mathbf{x} \in \mathbb{R}^D$:
  - If $\mathbf{x}$ goes right, we represent the feature selected as a binary vector $\mu_+ \in \{0,1\}^d$, containing ones only at $\mathcal{S}_+$.
  - If $\mathbf{x}$ goes left, we represent the feature selected as a binary vector $\mu_- \in \{0,1\}^d$, containing ones only at $\mathcal{S}_-$.
- We call μ^+ and μ^- the $\textsc{Node-Features}$, where location of one represents features selected by $\mathbf{w}$.

Interpret dataset using NODE-FEATURES

- 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.
- 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.
- 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.

Fashion-MNIST dataset



Difference between pair of classes

Decision node # 8 | dress class leaf # 16 | coat class leaf # 17

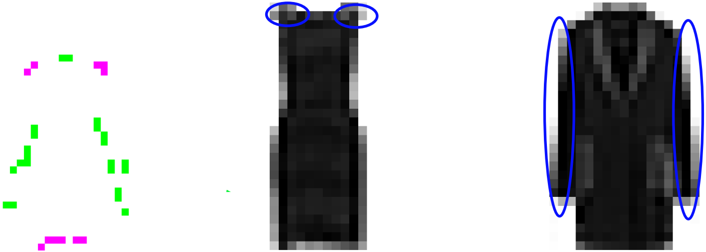
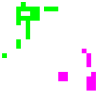
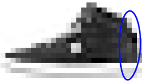
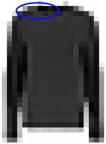
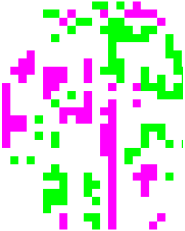
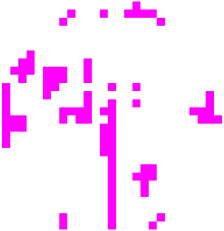
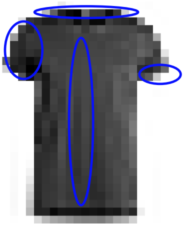


Figure: Magenta color represents negative weights (left child) and green color represents positive weights (right child).

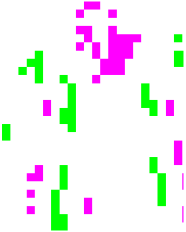
	left subtree		right subtree	
Decision node # 6	sneaker class leaf # 24	sandal class leaf # 25	pullover leaf # 26	trouser leaf # 27
				

Feature selection for a given instance

Decision node weight	weights along path node # 1, left child	image part selected
		

Feature selection for a given instance

Decision node
weight



weights along
path
node # 2, right child

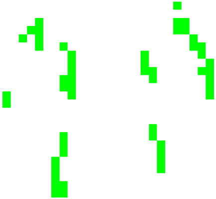
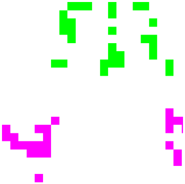


image part
selected

Feature selection for a given instance

Decision node weight



weights along path
node # 5, right child

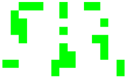


image part selected

Conclusion

- Sparse oblique trees can be used as an accurate yet interpretable model.
- Weights of the decision nodes can 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.

References

- Miguel Á. Carreira-Perpiñán and Pooya Tavallali, "Alternating optimization of decision trees, with application to learning sparse oblique trees," in *NEURIPS 2018*.
- 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*.
- Miguel Á. Carreira-Perpiñán and Arman Zharmagambetov, "Ensembles of bagged TAO trees consistently improve over random forests, AdaBoost and gradient boosting," in *FODS 2020*.