Robust Optimal Classification Trees under Noisy Labels: An Analysis and Experimental Reproduction

Efstratios Reppas [1]

[1]National Technical University of Athens

Introduction

The task of this project is to provide an analysis of the proposed OCSTVM algorithm at [1] and replicate its experimental results. The paper is concerned with the problem of binary classification, under the umbrella of supervised learning, which involves learning a separation rule that correctly classifies the instances of a labeled training dataset, while retaining the ability to generalize in observations not present in that dataset. Machine learning models with considerable success in this area have been models following the paradigms of Support Vector Machines (SVMs) [3] and Classification Trees (CTs) [2]. This paper aims to combine the best of both worlds, by introducing a model formulated using Mixed Integer Non-Linear Programming.

Background

The following presents theoretical concepts essential in understanding the proposed model.

- Re-label Support Vector Machines (RE-SVMs) [4] extend the notion of the celebrated original SVM formulation in [3] by allow the relabeling of certain labels during the training process, acknowledging that labels in the training data may not always be reliable. Briefly, SVMs find an optimal hyperplane that maximally separates the data points of two classes in a high-dimensional feature space, by calculating that hyperplane that maximizes the distance between itself and the nearest data points (called support vectors) from either class. The formulation of its extension, the Re-SVM, is as follows:

  $$\min_{w,w_0,e,\xi} \frac{1}{2}\|w\|^2 + c_1\sum_{i=1}^{n} e_i + c_2\sum_{i=1}^{n}\xi_i$$

  subject to:

  $$(1-2\xi_i)y_i(w^\top x_i + w) \geq 1 - e_i, \quad \forall i = 1,\ldots,n,$$

  $$e_i \geq 0, \quad \xi_i \in \{0,1\}, \quad \forall i = 1,\ldots,n.$$

- Optimal Classification Trees (OCTs) [5] Classification Trees are a widely-used hierarchical classification method. They form tree-like structure, where each node represents a classification rule that split the feature space in half. The tree is constructedby recursively partitioning the feature space into smaller, more homogenous subsets, ultimately assigning a label to each terminal leaf node. What is of great interest is the problem of constructing a classification tree with some sense of optimality. In [5] they develop a Mixed Integer Program that aims to maximize the model's accuracy while minimizing its size, coined as the *Optimal Classification Tree (OCT)*. It's objective function is described as follows:

  $$\min \sum_{t\in\mathcal{L}} L_t + \alpha \sum_{t\in\mathcal{B}} d_t$$

  where $\mathcal{L}$ is the set of leaf nodes, L_t is the mis-classification cost at leaf t, $\mathcal{B}$ is the set of branch nodes, d_t is a binary variable indicating whether a split occurs at node t and α is a regularization parameter that penalizes tree complexity. The aforementioned model can be generalized by assigning weights for each feature at each node instead of making the splitting decision based on one feature at each node, forming a hyperplane (*OCT-H* model).
- Classification and Regression Tree (CART) [2] Because the NP-hard complexity of calculating an optimal decision tree - imposed by its formulation as a Mixed Integer Linear Program - can be prohibitive in many applications, algorithms that utilize a greedy approach in selecting the features to be examined have been successful in creating small enough trees very fast. One of the most predominant algorithms is *CART*, which minimizes a heuristic called Gini index to find the tree's nodes at each level. In [?] the authors use this model for comparison purposes.

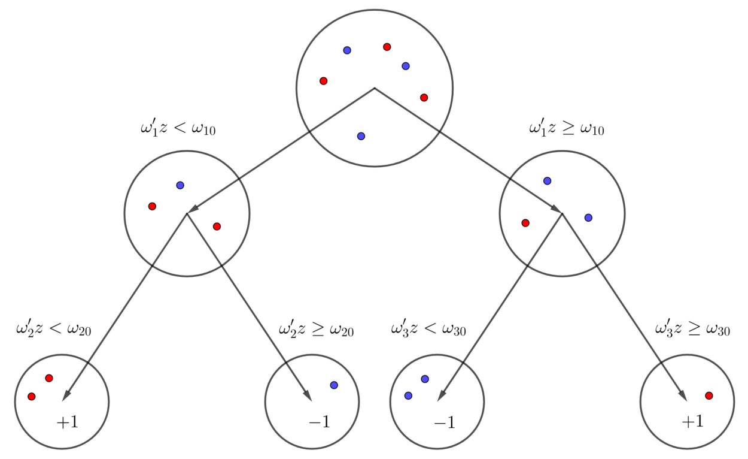
$\omega_1'z < \omega_{10}$
$\omega_1'z \geq \omega_{10}$
$\omega_2'z < \omega_{20}$
$\omega_2'z \geq \omega_{20}$
$\omega_3'z < \omega_{30}$
$\omega_3'z \geq \omega_{30}$
+1
−1
−1
+1

Figure 1. A Classification Tree of depth 2

The OCTSVM

The authors of [1] propose a novel model, combining the OCT-H and RE-SVM paradigms. More specifically, they build upon the OCT-H by including an SVM-like optimality constraint, additionally requiring the model to calculate hyperplanes at each level of the tree that maximize their distance from both classes. Its absence in the original OCT-H formulation allows the calculated hyperplanes to split the feature space in any way that minimizes the misclassification error in the training dataset, potentially missing out on generalizability, as illustrated in the figure below. The integration of a RE-SVM formulation instead of a simple SVM one aims to attribute robustness to the model in noisy datasets. Since we are combining the mixed integer formulation of the OCT-H with the non-linear one of RE-SVM, the final model will have a Mixed Integer Non-Linear Program formulation.

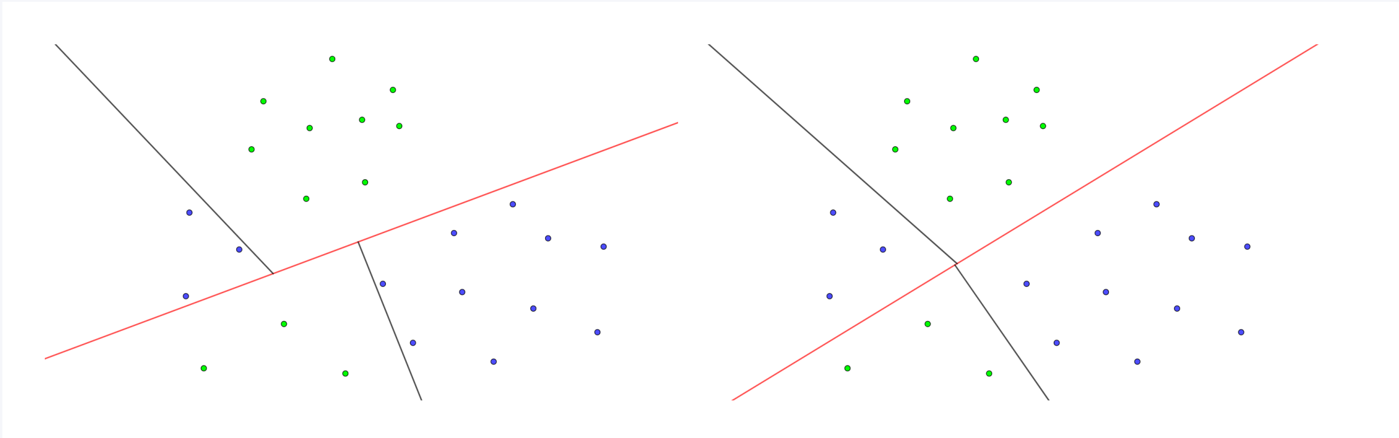

Figure 2. *Representation of the separation rules created with OCT-H (left) and OCTSVM (right). The distances of the hyperplanes from the classes are arbitrary in OCT-H, while in OCTSVM's case they keep a maximal distance from both.*

Model Formulation & Analysis

In this section we will explore the utility each constraint in the OCTSVM's formulation serves. For detailed explanations of how specifically each formula achieves its goal as well as the meaning and definitions of each variable, please refer to the project's report.

$$\min(\delta + c_1\sum_{i=1}^{n}\sum_{t=1}^{T} e_{it} + c_2\sum_{i=1}^{n}\sum_{t=1}^{T}\xi_{it} + c_3\sum_{t=1}^{T} d_t) \tag{1}$$

The **objective function** of the model. It is a linear combination of the objective functions of the RE-SVM and OCT models. the δ variable represents the SVM distance maximization, as illustrated below. The c_1, c_2, c_3 parameters define the prioritization of each objective.

$$\frac{1}{2}\|\omega_t\|_2 \leq \delta \quad \forall t = 1,\ldots,T \tag{2}$$

This is the **SVM constraint** as seen in the objective function of the RE-SVM formulation. Since δ will be minimized in the objective function, it indirectly minimizes the left side of the inequality that we are interested in. This constraint also provides our model's formulation with its non-linear characterization.

$$y_i(\omega_t x_i + \omega_{t0}) - 2y_i(\beta_t x_i + \beta_{it0}) \geq 1 - e_{it} - M(1 - z_{it}), \forall i = 1,\ldots,N, t = 1,\ldots,T. \tag{3}$$

$$\beta_{itj} = \xi_{it}\omega_{tj}, \quad \forall i = 1,\ldots,N,\ t = 0,\ldots,T,\ j = 0,\ldots,p \tag{4}$$

This set of constraints translates the **relabeling capability** of the RE-SVM into the tree structure of the OCTSVM. It characterizes our problem as Mixed Integer for the first time.

$$\|\omega_t\|_2 \leq Md_t \ \ \forall t = 1,\ldots,T. \tag{5}$$

$$d_t \leq d_p(t) \ \forall t = 1,\ldots,T \tag{6}$$

This set **defines the d variable** present in the objective function, as an **indicator that a node is used** and contains a splitting rule. It nullifies the weights of any node not in use and states that its descendant nodes cannot be used either, maintaining the hierarchy of the tree.

$$\sum_{t\in u} z_{it} = 1 \ \ \forall i = 1,\ldots,N, \forall u \in U \tag{7}$$

$$z_{it} \leq z_{ip(t)} \ \ \forall i = 1,\ldots,N, t = 2,\ldots,T \tag{8}$$

These are **sanity constraints**. The first states that each instance in the training dataset must be used in each level of the tree, in exactly one node per level. The second prohibits an instance from being present in a node if it wasn't present in its parent node, ensuring the hierarchical information processing of the tree.

$$\omega_t x_i + \omega_{t0} \geq -M(1 - \theta_{it}) \ \ \forall i = 1,\ldots,N, t = 1,\ldots,T \tag{9}$$

$$\omega_t x_i + \omega_{t0} \leq M\theta_{it} \ \ \forall i = 1,\ldots,N, t = 1,\ldots,T \tag{10}$$

$$z_{ip(t)} - z_{it} \leq \theta_{ip(t)} \ \ \forall i = 1, ..., N, t \in \tau_{bl} \tag{11}$$

$$z_{ip(t)} - z_{it} \leq 1 - \theta_{ip(t)} \ \ \forall i = 1, ..., N, t \in \tau_{br} \tag{12}$$

This set defines the **inheritance scheme** of the tree. The first two define the θ variable as a binary indicator of whether an instance lies on the left (negative) or the right (positive) side of the separation hyperplane. Now, with the help of this variable, the next two constraints define whether this instance will be inherited to the left or to the right child of the node.

$$e_{it} \in \mathbb{R}^+, \quad \xi_{it} \in \{0,1\}, \quad \beta_{it0} \in \mathbb{R}, \tag{13}$$

$$\theta_{it} \in \{0,1\}, \quad \forall i = 1,\ldots,N,\ t = 1,\ldots,T, \tag{14}$$

$$\omega_t \in \mathbb{R}^p, \quad \omega_t \in \mathbb{R}, \quad d_t \in \{0,1\}, \quad \forall t = 1,\ldots,T. \tag{15}$$

Finally, the lower bounds and definitions (real/binary) for the **variables** of the formulation.

Experimental Reproduction

Following the paper's example, we will be comparing the OCTSVM, CART and OCT models (OCT-H is omitted) in 8 different datasets. Since we are interested in testing the model's robustness under noisy labels, we will be applying different (0%, 20%, 30% and 40%) in the training set's labels. For more details regarding the methodology, please refer to the paper.

Dataset	% Flipped	CART		OCT		OCTSVM	
		Paper	Ours	Paper	Ours	Paper	Ours
Australian	0	82.99	**85.51**	85.44	83.48	**86.34**	84.5
	20	74.85	83.47	83.45	**85.5**	**84.55**	78
	30	66.87	**82.03**	79.15	70.72	**80.24**	72.08
	40	56.93	**76.67**	71.34	68.68	**73.89**	55.94
BreastCancer	0	92.22	94.56	93.18	94.71	**96.25**	**95.57**
	20	90.47	93	90.92	88.13	**91.57**	**95.28**
	30	83.29	93.42	**90.87**	89.56	87.98	**92.41**
	40	77.75	87.69	**86.50**	86.82	81.92	**92.27**
Heart	0	73.66	77.37	73.88	77.01	**84.13**	**84.84**
	20	72.98	67.78	71.63	71.09	**82.61**	**81.1**
	30	68.22	65.57	70.18	68.87	**80.51**	**79.98**
	40	62.36	66.62	64.52	63.77	**76.19**	**68.54**
Ionosphere	0	83.08	**89.45**	81.80	82.62	**85.51**	89.16
	20	75.65	**81.2**	79.59	78.91	**80.51**	74.95
	30	70.02	73.53	75.20	**76.08**	**77.79**	50.12
	40	60.17	64.1	70.65	**75.81**	**75.71**	43.29
MONK's	0	60.67	78.01	59.03	79.63	**61.26**	**85.18**
	20	57.11	75	59.59	75.46	**60.48**	**78.7**
	30	57.19	76.15	57.87	**77.55**	**60.09**	73.14
	40	54.01	64.81	55.25	**75**	**60.12**	74.07
Sonar	0	65.36	71.63	66.90	69.23	**74.76**	**79.33**
	20	57.41	60.58	60.38	**67.3**	**69.92**	65.38
	30	54.64	59.13	58.97	**62.98**	**67.30**	62.02
	40	55.16	55.77	57.77	**62.01**	**63.84**	56.25
Wholesale	0	**90.28**	**91.59**	90.01	91.14	89.75	88.41
	20	83.79	88.86	87.84	**90.23**	82.06	87.72
	30	78.54	84.09	**84.58**	84.77	78.13	**85.91**
	40	69.52	67.72	**75.79**	76.14	71.72	**85.23**

Table 1. Accuracy (%) comparison of the models between [1] and our experiments (excl. averages - bold is best).

The main observation from the paper still stands: OCTSVM demonstrates superior performance in most datasets. However, the robustness in noisy datasets isn't prevalent as in [1], probably due to additional constraints we had in the parameter search space. An interesting observation was that in many cases the OCTSVM used ultimately only one layer, essentially reducing itself into an RE-SVM. Also, CART's speed difference (in a scale of seconds instead of hours) probably renders complex mathematical programs like OCT and OCTSVM impractical in many applications.

References

[1] V. Blanco, A. Japón, J. Puerto, "Robust optimal classification trees under noisy labels," arXiv:2012.08560 [cs.LG], Dec. 2020.

[2] L. Breiman, J. H. Friedman, R. A. Olshen, C. J. Stone, *Classification and Regression Trees*, 1st ed. Belmont, CA, USA: Wadsworth, 1984.

[3] V. Vapnik, *The Nature of Statistical Learning Theory*, 1st ed. New York, NY, USA: Springer, 1995.

[4] V. Blanco, A. Japón, J. Puerto, "A mathematical programming approach to SVM-based classification with label noise," *Computers & Industrial Engineering*, vol. 172, p. 108611, 2022.

[5] D. Bertsimas, J. Dunn, "Optimal classification trees," *Machine Learning*, vol. 106, no. 7, pp. 1039–1082, 2017.