The results of this case presented the MONDP in 68% of the inventory levels. The levels of inventory that represent the policy were mostly at the extremities of the range in our instant. Limited sensitivity analysis on the costs was performed, without showing any improvements in the policy. A more detailed account of this case study and methodology can be found in [10].

**References:**


[1] L. N. van Wassenhove, "Blackett Memorial Lecture: Humanitarian Aid Logistics: Supply Chain Management in High Gear," *Journal of the Operational Research Society*, 2006.
[2] I. Oxenhaut, *National Director of the Disasters Division. Mexican Red Cross*. 2015.
[3] E. I. Mora-Ochomogo, J. Mora-Vargas, and M. Serrato, "A Qualitative Analysis of Inventory Management Strategies in Humanitarian Logistics Operations," *International Journal of Combinatorial Optimization Problems and Informatics*, vol. 7, no. (1):40, 2016.
[4] B. Balcik, C. D. C. Bozkir, and O. E. Kundakcioglu, "A literature review on inventory management in humanitarian supply chains," *Surveys in Operations Research and Management Science*, vol. 21, no. 2, pp. 101–116, Dec. 2016, doi: 10.1016/j.sorms.2016.10.002.
[5] L. Destro and J. Holguín-Veras, "Material convergence and its determinants: Case of Hurricane Katrina," *Transportation Research Record*, 2011, doi: 10.3141/2234-02.
[6] J. Holguín-Veras, M. Jaller, L. N. Van Wassenhove, N. Pérez, and T. Wachtendorf, "Material Convergence: Important and Understudied Disaster Phenomenon," *Natural Hazards Review*, 2014, doi: 10.1061/(asce)nh.1527-6996.0000113.
[7] M. A. Ülkü, K. M. Bell, and S. G. Wilson, "Modeling the impact of donor behavior on humanitarian aid operations," *Ann Oper Res*, vol. 230, no. 1, pp. 153–168, Jul. 2015, doi: 10.1007/s10479-014-1623-5.
[8] S. Penta, T. Wachtendorf, and M. M. Nelan, "Disaster Relief as Social Action: A Weberian Look at Postdisaster Donation Behavior," *Sociol Forum*, vol. 35, no. 1, pp. 145–166, Mar. 2020, doi: 10.1111/socf.12571.
[9] R. A. Cook and E. J. Lodree, "Dispatching policies for last-mile distribution with stochastic supply and demand," *Transportation Research Part E: Logistics and Transportation Review*, vol. 106, pp. 353–371, Oct. 2017, doi: 10.1016/j.tre.2017.08.008.
[10] Mora-Ochomogo, I., Serrato, M., Mora-Vargas, J., and Akhavan-Tabatabaei, R. (2022). "Application of a Markov Decision Process in Collection Center Operations". In: Regis-Hernández, F., Mora-Vargas, J., Sánchez-Partida, D., Ruiz, A. (eds) Humanitarian Logistics from the Disaster Risk Reduction Perspective . Springer, Cham. https://doi.org/10.1007/978-3-030-90877-5_14


**OR TUTORIAL** Section Editor: **Javier Marenco** <jmarenco@campus.ungs.edu.ar>

# OPTIMIZATION-BASED APPROACHES FOR LEARNING OPTIMAL CLASSIFICATION TREES

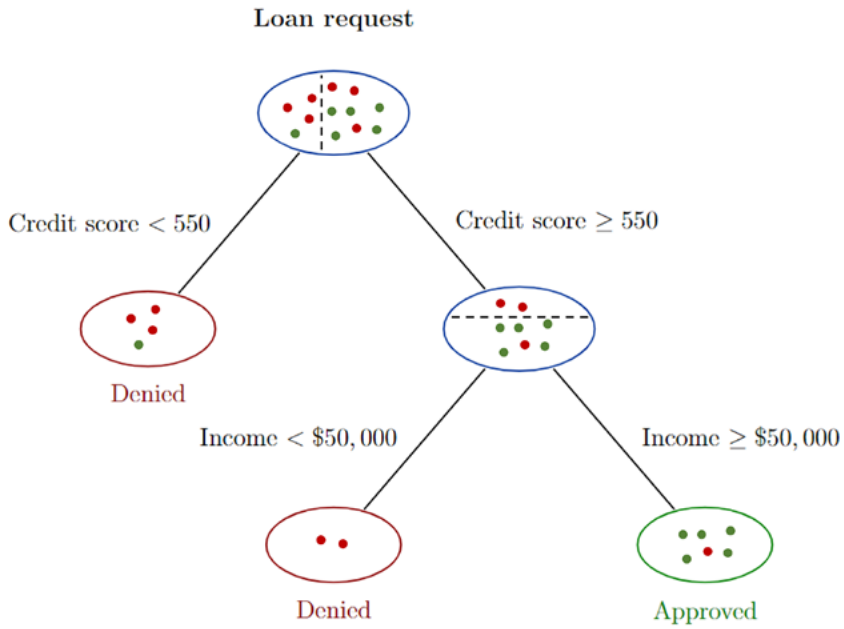

**Federico D'Onofrio:** <federico.donofrio@uniroma1.it>
**Marta Monaci:** <marta.monaci@uniroma1.it>
**Laura Palagi:** <laura.palagi@uniroma1.it>

*Department of Computer, Control and Management Engineering Antonio Ruberti (DIAG), Sapienza University of Rome, Rome, Italy*


**Supervised Classification and Classification Trees.**
In the context of supervised classification, a set of samples belonging to different classes is given, and the goal is to build a Machine Learning (ML) model for classifying new samples into the correct class. Mathematical optimization plays a major role in the training phase, i.e. the process of building such ML models. In this phase, the aim is not solely to identify a model which correctly classifies all the input data, but rather one which is capable to generalize to never seen data. Classification problems are faced in many real-world contexts, including medical diagnosis (to diagnose a patient based on symptoms, medical history, and other factors), fraud detection (to identify fraudulent activities by analyzing patterns in financial transactions) and credit scoring (to assess the creditworthiness of borrowers and make informed lending decisions), etc. In such high stakes domains, it is crucial to use interpretable ML models [12], which can provide explanatory insights on their decision-making process. Decision trees are among the most popular Supervised ML tools for solving classification tasks. They are renowned for their ease of use, transparent structure, and, most of all, for their interpretability. Indeed, the logic of a classification tree is easily understandable by humans and it is straightforward to extract decision rules from the model as a conjunction of predicates, in contrast to other machine learning methods that are perceived as opaque "black boxes".

Fig. 1 reports a toy example of a classification tree trained to classify customers for the approval or denial of a loan request. According to the decision rule defined by the tree, first if the applicant has a credit score above 550, the loan request is approved; otherwise if the applicant has a stable source of income above $50,000, the loan request would be approved too; otherwise it is denied.

More formally, let us consider a training dataset composed of $P$ samples $(x^i, y^i)$, each with input *features* $x^i \in R^n$ and a *class label* $y^i \in \{1, \ldots, K\}$, indicating which of the $K$ possible classes the sample $i$ belongs to. During the training phase, a classification tree method builds up a binary tree structure of a maximum predefined depth. A decision tree is composed of *branch nodes* and *leaf nodes*. Each branch node $t$ applies a splitting rule on the feature space, routing samples to its left or right child node. Each splitting rule is defined by a separating hyperplane $H_t(x) := \{x : h_t(x) = 0\}$, where $h_t(x) = a_t^T x - b_t$ is the hyperplane function and $a_t \in R^n$ and $b_t \in R^n$. If $h_t(x^i) \geq 0$, sample $i$ will follow the right branch of node $t$, otherwise it will follow the left one. Leaf nodes are the terminal nodes of the tree and they act as collectors of samples. In particular, each leaf is assigned a class label according to some simple predefined rule, usually the most common label among the samples in the node.




▲ Figure 1: Classification tree example.

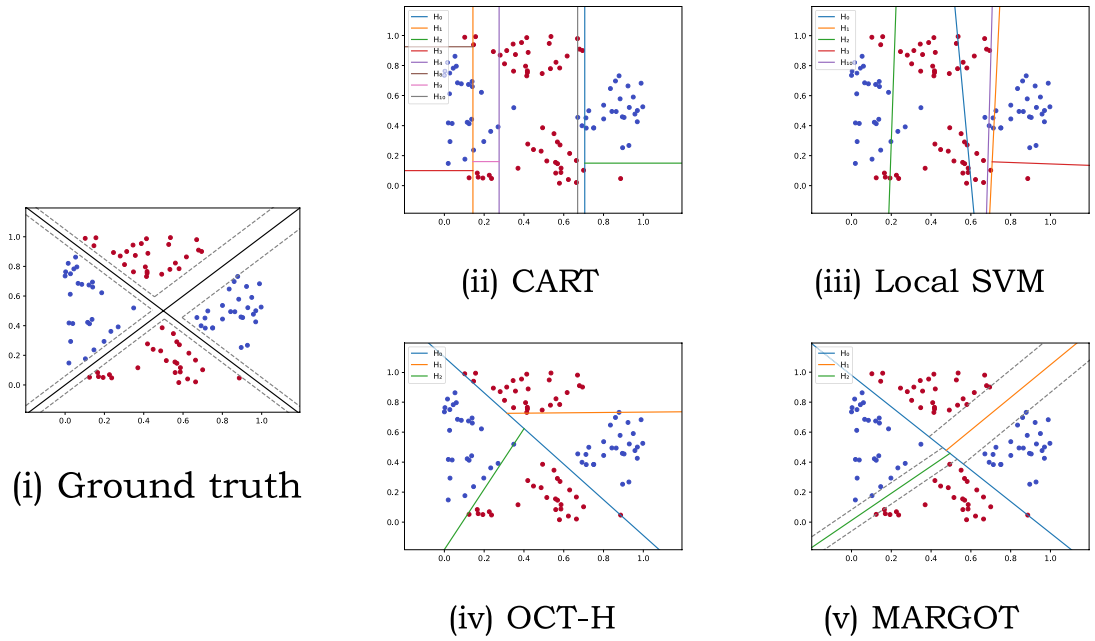
Decision trees can be divided into *univariate* and *multivariate* trees depending on the type of hyperplane splits employed. In a univariate tree, hyperplanes are axis-aligned involving one single feature per split. Thus, the branching rule simply checks if the value of a single feature $x_t$ is above or under a given threshold $b_t$. Multivariate trees, instead, apply oblique hyperplanes which may involve several features. Consequently, multivariate splits allow for more flexibility yielding shallow trees with less branching levels than univarate ones, even though they are less interpretable. According to the hierarchical tree structure, the feature space will be recursively partitioned into disjoint regions and each final partition corresponds to a leaf node. Each sample in the leaf will be classified with the same class label, the one assigned to the leaf. The obtained tree is then used to classify out-of-sample data: every new sample will follow a unique path within the tree ending up in a leaf node. The training phase aims at finding coefficients $a_t$ and $b_t$ and at assigning class labels to leaves optimizing some measure of performance.

**Overview on Classification Trees**

It is well-known that learning an optimal binary decision tree is an *NP*-complete problem [11]. For this reason, traditional approaches build univariate decision trees based on simple iterative heuristics. In general, they rely on a top-down greedy strategy for growing the tree by generating splits at each node, and, once the tree is built, a bottom-up pruning procedure is applied to handle the complexity of the tree, i.e. the number of splits. Breiman et al. [6] developed a heuristic algorithm known as CART (Classification and Regression Trees), for finding univariate decision trees. Starting from the root node, each hyperplane split is generated by minimizing a local impurity function, e.g. the Gini impurity for classification tasks. Other univariate approaches employing different impurity functions were later proposed by Quinlan (ID3, C4.5). The main drawback of these approaches lies on their greedy nature which may lead to myopic decisions. Indeed, each split is determined in isolation in the tree, yielding tree classifiers not able to capture well the underlying truth of the dataset. Thus, these heuristics lead to short computational times, but may result in poor generalization performances. In order to overcome these shortcomings, tree ensemble methods, such as Random Forests and XGBoost, have been proposed. These approaches aggregate together decision trees achieving better predictive performances at the expense of lower interpretability, resulting in "black-box" models.

Another way to improve prediction quality which retains interpretability is to use multivariate decision trees which employ oblique hyperplane splits. In this case, top-down greedy approaches are not efficient and cannot be used anymore. In the last years, there has been a growing interest in the defintion of exact optimization approaches to find Optimal Classification Trees (OCTs) using mathematical programming tools and, in particular Mixed Integer Programming (MIP). Thanks to the great improvement of both algorithms for MIP and hardware, MIP approaches became viable in the construction of OCT models. Such approaches adopt an holistic view of the decision tree to define a single optimization model accounting for the tree hierarchical structure. Indeed, the MIP framework is perfectly suitable to express the combinatorial nature of the decisions involved in the construction process of a tree. Discrete decisions can be related to the tree topology and the branching rules, e.g. whether to split in a node and which features to select in a split. Other choices may regard the discrete outcomes, e.g. which leaf a sample is assigned to and whether a point is well classified. Beyond this expressive power, the mixed-integer framework also lends itself to handle global objectives and constraints to embed desirable properties such as fairness, sparsity, cost-sensitivity, robustness.

In their seminal paper [3], Bertsimas and Dunn proposed for the first time mixed-integer linear models to build optimal trees with a fixed maximum depth both with univariate splits and with multivariate ones. The objective is to seek a trade-off between the minimization of the misclassification loss and either the complexity of the tree or the sparsity of the hyperplanes. In both models, each sample is forced to end up in a single leaf (assignment constraints) and a class label for each leaf node is chosen according to the most common label rule. The classification error in the objective function is computed according to the assignment of each sample to a leaf. Routing constraints enforce each sample to follow a unique path, while other constraints control the complexity of the tree by imposing a minimum number of points accepted by each leaf. Along these lines, several other formulations have been proposed. Some of the most recent works are: [13], where the authors presented an integer linear formulation whose size is largely independent from the training data size; [10], where a mixed-integer model is derived by exploiting the special structure of categorical features for binary classification tasks; [1], where a flow-based mixed-integer linear model with a stronger linear relaxation is proposed for learning optimal trees with binary features. Alongside integer optimization approaches, continuous optimization ones have also been investigated in the optimal trees context. In [5], Blanquero et al. proposed a nonlinear programming model to find an optimal "randomized" tree with oblique splits. At each node, a random decision is made and a sample is not assigned to a class in a deterministic way but only with a given probability. For the interested reader who wishes to further investigate the topic, we suggest taking a closer look at the survey by Carrizosa et al. [7], which provides an extensive analysis of optimization approaches for constructing optimal classification trees.




▲ Figure 2: Comparison of heuristic and optimal approaches on a 2D synthetic dataset.

**Maximum Margin Optimal Trees.**

Following a different view point, approaches using Support Vector Machines (SVMs) [8] for each split in the tree have been investigated (e.g. [2, 4]). In this context, in [9] a novel mixed-integer quadratic formulation for training optimal trees for solving binary classification tasks is proposed. The resulting model, Margin Optimal Classification Tree (MARGOT), exploits the generalization properties of SVMs and defines branching rules as maximum margin hyperplanes by following a linear SVM paradigm in a hierarchical tree structure. The maximum depth of the tree is predetermined, and each branch node of the model defines an SVM-based problem. The overall objective function is a trade-off between minimization of the misclassification cost and the maximization of the margin of each splitting hyperplane. Routing and assignment constraints are used to nest the "local" SVM problems together. In MARGOT model, it is possible to induce sparsity of the hyperplanes by limiting the number of features used at each split. Indeed, sparsity is a core component of interpretability [12] and having fewer features selected at each branch node allows the end user to identify the key factors influencing the outcome. Two alternative versions of MARGOT are proposed which train the optimal tree performing a feature selection either by adding budget constraints on the number of used features or by penalizing the number of used features in the objective function.

Fig. 2 shows a synthetic dataset (i) used to compare two heuristic and two optimal approaches for constructing a classification tree. CART (ii) uses axis aligned splits, and it needs higher depths in order to achieve good classification performances; it is prone to overfitting. Local SVM (iii) [9] is a simple top-down approach which, for each branch node, solves an SVM problem defined only on the data routed to that node. Even though it creates oblique splits which are more flexible than ortogonal ones, the overall tree lacks of generalization capabilities. OCT-H (iv) [3] creates hyperplanes which correctly classify all samples but do not take into account their distance from the cluster of points of the same color. Finally, MARGOT (v) builds a more robust tree which mostly resembles the ground truth. The good performance of MARGOT are confirmed on a benchmark of datasets from the UCI Repository. More details about the formulations and the computational results can be found in [9]. The source code of the experiments is available at: https: //github.com/m-monaci/MARGOT.

**References**

[1] Sina Aghaei, Andrés Gómez, and Phebe Vayanos. Strong optimal classification trees. *CoRR*, abs/2103.15965, 2021.

[2] Kristin P. Bennett and Jennifer A. Blue. A support vector machine approach to decision trees. *1998 IEEE International Joint Conference on Neural Networks Proceedings. IEEE World Congress on Computational Intelligence (Cat. No.98CH36227)*, 3:2396–2401 vol.3, 1998.

[3] Dimitris Bertsimas and Jack Dunn. Optimal classification trees. *Machine Learning*, 106(7):1039–1082, July 2017.

[4] Víctor Blanco, Alberto Japón, and Justo Puerto. Robust optimal classification trees under noisy labels. *Advances in Data Analysis and Classification*, 16(1):155–179, 2022.

[5] Rafael Blanquero, Emilio Carrizosa, Cristina Molero-Río, and Dolores Romero Morales. Optimal random- ized classification trees. *Computers & Operations Research*, 132:105281, Aug 2021.

[6] Leo Breiman, Jerome Friedman, Charles J. Stone, and R.A. Olshen. *Classification and Regression Trees*. Chapman and Hall/CRC, 1984.

[7] Emilio Carrizosa, Cristina Molero del Rio, and Dolores Romero Morales. Mathematical optimization in classification and regression trees. *TOP*, 29(1):5–33, April 2021. Published online: 17. Marts 2021.

[8] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, sep 1995.

[9] Federico D'Onofrio, Giorgio Grani, Marta Monaci, and Laura Palagi. Margin optimal classification trees. *preprint arXiv:2210.10567*, 2022.

[10] Oktay Gũnluữk, Jayant Kalagnanam, Minhan Li, Matt Menickelly, and Katya Scheinberg. Optimal decision trees for categorical data via integer programming. *Journal of Global Optimization*, 81:233–260, 2021.

[11] Laurent Hyafil and Ronald L. Rivest. Constructing optimal binary decision trees is NP-complete. *Inf. Process. Lett.*, 5:15–17, 1976.

[12] Cynthia Rudin, Chaofan Chen, Zhi Chen, Haiyang Huang, Lesia Semenova, and Chudi Zhong. Interpretable machine learning: Fundamental principles and 10 grand challenges. *Statistics Surveys*, 16:1–85, 2022.

[13] Sicco Verwer and Yingqian Zhang. Learning optimal classification trees using a binary linear program formulation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):1625–1632, Jul. 2019.