Optimal Decision Trees Model - Code Guidance

Matt Menickelly

March 2, 2017

We begin with some notation. Let the set of all samples be indexed by $I = \{1, 2, \ldots, N\}$, let $I_+ \subset I$ denote the indices of samples with positive labels $+1$ (i.e. $y_i = 1$ for $i \in I_+$), and let $I_- = I \setminus I_+$ denote the indices of the samples with negative labels. Let the set of groups be indexed by $G = \{1, 2, \ldots, |G|\}$ and the set of features be indexed by $J = \{1, 2, \ldots, d\}$. In addition, let $g(j) \in G$ denote the group that contains feature $j \in J$ and let $J(g)$ denote the set of features that are contained in group g. We assume that the topology of the decision tree is given, so let the set of decision nodes be indexed by $K = \{1, 2, \ldots, |K|\}$ and the set of leaf nodes be indexed by $B = \{1, 2, \ldots, |B|\}$. We denote the indices of leaf nodes with positive labels by $B_+ \subset B$ and the indices of leaf nodes with negative labels by $B_- = B \setminus B_+$. For convenience, we let B_+ contain even indices, and B_- contain the odd ones.

We now describe our key decision variables and the constraints on these variables. We use binary variables $v_g^k \in \{0, 1\}$ for $g \in G$ and $k \in K$ to denote if group g is selected for branching at decision node k. Exactly one group has to be be selected for branching at a decision node; consequently, we have the following set of constraints:

$$\sum_{g \in G} v_g^k = 1 \quad \forall k \in K, \tag{1}$$

in the formulation.

The second set of binary variables $z_j^k \in \{0, 1\}$ for $j \in J$ and $k \in K$ are used to denote if feature j is one of the selected features for branching at a decision node k. Clearly, a feature $j \in J$ can be selected only if the group containing it, namely $g(j)$, is selected at that node. Therefore we have the following set of constraints:

$$z_j^k \leq v_{g(j)}^k \quad \forall j \in J, \tag{2}$$

in the formulation. Without loss of generality, we use the convention that if a sample has one of the selected features at a given node, it follows the left branch at that node; otherwise it follows the right branch.

Let

$$S \;=\; \Big\{(v, z) \in \{0,1\}^{|K| \times |G|} \times \{0,1\}^{|K| \times d} : \quad (v, z) \text{ satisfies inequalities (1) and (2)} \Big\},$$

and note that for any $(v, z) \in S$ one can construct a corresponding decision tree in a unique way and vice versa. In other words, for any given $(v, z) \in S$ one can easily decide which leaf node each sample is routed to. We next describe how to relate these variables (and therefore the corresponding decision tree) to the samples.

We use binary variables $c_b^i \in \{0, 1\}$ for $b \in B$ and $i \in I$ to denote if sample i is routed to leaf node b. This means that variable c_b^i should take the value 1 only when sample i exactly follows the unique path in the decision tree that leads to leaf node b. With this in mind, we define the expression

$$L(i, k) = \sum_{j \in J} a_j^i z_j^k \quad \forall k \in K, \ \forall i \in I, \tag{3}$$

and make the following observation:

Proposition 1 *Let $(z, v) \in S$. Then, for all $i \in I$ and $k \in K$ we have $L(i,k) \in \{0,1\}$. Furthermore, $L(i,k) = 1$ if and only if there exists some $j \in J$ such that $a^i_j = 1$ and $z^k_j = 1$.*

Proof 1 *For any $(z, v) \in S$ and $k \in K$, exactly one of the v^k_g variables, say $v^k_{g'}$, takes value 1 and $v^k_g = 0$ for all $g \neq g'$. Therefore, $z^k_j = 0$ for all $j \notin J(g)$. Consequently, the first part of the claim follows for all $i \in I$ as $L(i,k) = \sum_{j \in J} a^i_j z^k_j = \sum_{j \in J(g')} a^i_j z^k_j = z^k_{j_i} \in \{0,1\}$ where $j_i \in J(g')$ is the index of the unique feature for which $a^i_{j_i} = 1$. In addition, $L(i,k) = 1$ if and only if $z^k_{j_i} = 1$ which proves the second part of the claim.*

Consequently, the expression $L(i,k)$ indicates if sample $i \in I$ branches left at node $k \in K$. Similarly, we define the expression

$$R(i,k) = 1 - L(i,k) \quad \forall k \in K,\ \forall i \in I, \tag{4}$$

to indicate if sample i branches right at node k.

To complete the model, we relate the expressions $L(i,k)$ and $R(i,k)$ to the c^i_b variables. Given that the topology of the tree is fixed, there is a unique path leading to each leaf node $b \in B$ from the root of the tree. This path visits a subset of the nodes $K(b) \subset K$ and for each $k \in K(b)$ either the left branch or the right branch is followed. Let $K^L(b) \subseteq K(b)$ denote the decision nodes where the left branch is followed to reach leaf node b and let $K^R(b) = K(b) \setminus K^L(b)$ denote the decision nodes where the right branch is followed. Sample i is routed to b only if it satisfies all the conditions at the nodes leading to that leaf node. Consequently, we define the constraints

$$c^i_b \leq L(i,k) \quad \text{for all} \quad \forall b \in B,\ \forall i \in I,\ k \in K^L(b), \tag{5}$$

$$c^i_b \leq R(i,k) \quad \text{for all} \quad \forall b \in B,\ \forall i \in I,\ k \in K^R(b), \tag{6}$$

for all $i \in I$ and $b \in B$. Combining these with the equations

$$\sum_{b \in B} c^i_b = 1 \quad \forall i \in I \tag{7}$$

gives a complete formulation. Let

$$Q(z, v) = \left\{ c \in \{0,1\}^{N \times |B|} : \text{ such that (5)-(7) hold} \right\}.$$

We next formally show that combining the constraints in S and $Q(z, v)$ gives a correct formulation.

Proposition 2 *Let $(z, v) \in S$, and let $c \in Q(z, v)$. Then, $c^i_b \in \{0,1\}$ for all $i \in I$ and $b \in B$. Furthermore, if $c^i_b = 1$ for some $i \in I$ and $b \in B$, then sample i is routed to leaf node b.*

Proof 2 *Given $(z, v) \in S$ and $i \in I$, assume that the correct leaf node sample i should be routed to in the decision tree defined by (z, v) is the leaf node b'. For all other leaf nodes $b \in B \setminus \{b'\}$, sample i either has $L(i,k) = 0$ for some $k \in K^L(b)$ or $R(i,k) = 0$ for some $k \in K^R(b)$. Consequently, $c^i_b = 0$ for all $b \neq b'$. Equation (7) then implies that $c^i_{b'} = 1$ and therefore $c^i_b \in \{0,1\}$ for all $b \in B$. Conversely, if $c^i_{b'} = 1$ for some $b' \in B$, then $L(i,k) = 1$ for all $k \in K^L(b)$ and $R(i,k) = 1$ for all $k \in K^R(b)$.*

We therefore have the following integer programming (IP) formulation:

$$\max \quad \sum_{i \in I_+} \sum_{b \in B_+} c^i_b + C \sum_{i \in I_-} \sum_{b \in B_-} c^i_b \tag{8a}$$

$$\text{s. t.} \quad (z, v) \in S \tag{8b}$$

$$c \in Q(z, v) \tag{8c}$$

where C in the objective (8a) is a constant weight chosen in case of class imbalance. For instance, if a training set has twice as many good examples as bad examples, it may be worth considering setting $C = 2$, so that every correct classification of a bad data point is equal to two correct classifications of good data points.

Observe in the provided code that the order of the variables is

$$z_1^1, \ldots, z_{|J|}^1, \ldots, z_1^{|K|}, \ldots, z_{|J|}^{|K|}, v_1^1, \ldots, v_{|G|}^1, \ldots, v_1^{|K|}, \ldots, v_{|G|}^{|K|}, c_1^1, \ldots, c_1^{|I|}, \ldots, c_{|B|}^1, \ldots, c_{|B|}^{|I|}.$$

When constructing the inequality matrix A, the coefficients are applied in this way, as can be seen by reading the code and comparing to the described constraints.

0.1 Computational Tractability

While (8) is a correct formulation, it can be improved to enhance computational performance. We next discuss some ideas that help reduce the size of the problem, break symmetry and strengthen the linear programming relaxation.

0.1.1 Strengthening the model

In our code, setting strengthen = True will perform the following. Consider inequalities (5)

$$c_b^i \leq L(i,k)$$

for $i \in I$, $b \in B$ and $k \in K^L(b)$ where $K^L(b)$ denotes the decision nodes where the left branch is followed to reach the leaf node b. Also remember that $\sum_{b \in B} c_b^i = 1$ for $i \in I$ due to equation (7).

Now consider a fixed $i \in I$ and $k \in K$. If $L(i,k) = 0$, then $c_b^i = 0$ for all b such that $k \in K^L(b)$. On the other hand, if $L(i,k) = 1$ then at most one $c_b^i = 1$ for b such that $k \in K^L(b)$. Therefore,

$$\sum_{b \in B : K^L(b) \ni k} c_b^i \leq L(i,k) \tag{9}$$

is a valid inequality for all $i \in I$ and $k \in K$. While this inequality is satisfied by all integral solutions to the set $Q(z,v)$, it is violated by some of the solutions to its continuous relaxation. We replace the inequalities (5) in the formulation with (9) to obtain a tighter formulation. We also replace inequalities (6) in the formulation with the following valid inequality:

$$\sum_{b \in B : K^R(b) \ni k} c_b^i \leq R(i,k) \tag{10}$$

for all $i \in I$ and $k \in K$.

0.1.2 Deleting unnecessary variables

In our code, setting deleted=True will perform the following. Notice that the objective function (8a) uses variables c_b^i only if it corresponds to a correct classification of the sample (i.e., $i \in I_+$ and $b \in B_+$, or $i \in I_-$ and $b \in B_-$). Consequently, remaining c_b^i variables can be projected out of the formulation without changing the value of the optimal solution. We therefore only define c_b^i variables for

$$\left\{(i,b) : i \in I_+, b \in B_+, \text{ or, } i \in I_-, b \in B_-\right\} \tag{11}$$

and write constraints (5) and (6) for these variables only. This reduces the number of c variables and the associated constraints in the formulation by a half. In addition, we delete equation (7).

Also note that the objective function (8a) is maximizing a (weighted) sum of c_b^i variables and the only constraints that restrict the values of these variables are inequalities (5) and (6) which have a right hand

side of 0 or 1. Consequently, replacing the integrality constraints $c_b^i \in \{0,1\}$ with simple bound constraints $1 \geq c_b^i \geq 0$, still yields optimal solutions that satisfy $c_b^i \in \{0,1\}$. Consequently, we do not require c_b^i to be integral in the formulation and therefore reduce the number of integer variables significantly.

0.1.3 Breaking symmetry: Anchor features

In our code, setting anchor=True will perform the following. If the variables of an integer program can be permuted without changing the structure of the problem, the integer program is called *symmetric*. This poses a problem for MILP solvers (such as IBM ILOG CPLEX) since the search space increases exponentially, see Margot (2009). The formulation (8) falls into this category as there may be multiple alternate solutions that represent the same decision tree. In particular, consider a decision node that is not adjacent to leaf nodes and assume that the subtrees associated with the left and right branches of this node are symmetric (i.e. they have the same topology). In this case, if the branching condition is reversed at this decision node (in the sense that the values of the v variables associated with the chosen group are flipped), and, at the same time, the subtrees associated with the left and right branches of this node are switched, one obtains an alternate solution to the formulation corresponding to the same decision tree. To avoid this, we designate one particular feature $j(g) \in J(g)$ of each group $g \in G$ to be the *anchor feature* of that group and enforce that if a group is selected for branching at such a node, samples with the anchor feature follow the left branch. More precisely, we add the following equations to the formulation:

$$z_{j(g)}^k = v_{j(g)}^k \tag{12}$$

for all $g \in G$, and all $k \in K$ that is not adjacent to a leaf node and has symmetric subtrees hanging on the right and left branches. While equations (12) lead to better computational performance, they do not exclude any decision trees from the feasible set of solutions.

0.1.4 Relaxing some binary variables

In our code, setting relaxed=True and/or relaxedobj=True will do the following. Obviously, relaxed=True corresponds to the z variables while relaxedobj=True corresponds to the v variables. The computational difficulty of a MILP typically increases with the number of integer variables in the formulation and therefore it is desirable to impose integrality on as few variables as possible. Let $j \in J$ and consider a feature selection variable z_j^k, where $k \in K$ is a decision node that is adjacent to leave nodes. We next show that these variables take values $\{0,1\}$ in an optimal solution even when they are not explicitly constrained to be integral.

Proposition 3 *Every extreme point solution to* (8) *is integral even if variables z_j^k are not declared integral for $j \in J$ and decision nodes $k \in K$ that are adjacent to a leaf nodes.*

Proof 3 *Assume the claim does not hold and let (v, z, c) be an extreme point solution to the relaxed formulation such that z_j^k is fractional for some $j \in J$ and $k \in K$ and node k is adjacent to leaf nodes b^+ and b^-. Due to integrality, $v_{g(j)}^k = 1$ as $v_{g(j)}^k \geq z_j^k$ and $z_j^k > 0$. Therefore $v_g^k = 0$ for all other $g \in G \setminus \{g(j)\}$. Let $I_0 \subset I$ be the collection of samples that satisfy the conditions to follow the path leading to node k. Clearly, $c_{b^+}^i$ and $c_{b^-}^i$ variables associated with all $i \in I \setminus I_0$ are zero in the solution.*

We will now construct two solutions $(v, \bar{z}, \bar{c})$ and $(v, \hat{z}, \hat{c})$ by increasing and decreasing z_j^k with some small $\epsilon > 0$, respectively, and modifying the c variables accordingly. We will then argue that (v, z, c) is a convex combination of these new solutions and therefore it is not an extreme point solution. Remember that in the formulation we have $c_{b^+}^i$ variables associated with $i \in I_+$ and $c_{b^-}^i$ variables associated with $i \in I_-$ only. In

addition,

$$c^i_{b^+} \leq R(i,k) \quad = \quad 1 - \sum_{j \in J} a^i_j z^k_j \quad \forall i \in I_0 \cap I_+ \tag{13}$$

$$c^i_{b^-} \leq L(i,k) \quad = \quad \sum_{j \in J} a^i_j z^k_j \quad \forall i \in I_0 \cap I_-. \tag{14}$$

Now consider $i \in I_0$. If $a^i_j = 0$, then we do not change the value of the associated c variable in the perturbed solutions $(v, \bar{z}, \bar{c})$ and $(v, \hat{z}, \hat{c})$. If $a^i_j = 1$, we consider two cases. If $i \in I_+$ and $c^i_{b^+} = 0$, or, $i \in I_-$ and $c^i_{b^-} = 0$, then we again do not change the value of the associated c variable in the perturbed solutions. Finally, for the remaining $i \in I_+$ we set $\bar{c}^i_{b^+} = c^i_{b^+} + \epsilon$ and $\hat{c}^i_{b^+} = c^i_{b^+} - \epsilon$. Similarly, for the remaining $i \in I_-$ we set $\bar{c}^i_{b^-} = c^i_{b^-} - \epsilon$ and $\hat{c}^i_{b^-} = c^i_{b^-} + \epsilon$. It is easy to check that $(v, \bar{z}, \bar{c})$ and $(v, \hat{z}, \hat{c})$ are feasible solutions and contain (v, z, c) in their convex hull.