Characterizing Uncertainty in Decision Trees through Imprecise Splitting Rules

Malte Nalenz MALTE.NALENZ@STAT.UNI-MUENCHEN.DE
Thomas Augustin THOMAS.AUGUSTIN@STAT.UNI-MUENCHEN.DE
Institut für Statistik, Ludwig-Maximilians-Universität (LMU), Munich, Germany

Motivation Decision trees are one of the most common prediction methods. It's popularity mostly stems from their interpretational and methodological simplicity. Decision tree inducers successively partition the covariate space x into smaller subspaces that are purer with respect to their target values y. Most practical algorithms use a greedy procedure that uses the x-dimension with the largest gain in purity at each step. Decision Trees have been shown to posses decent predictive performance and adaptivity to arbitrary mapping functions $y = f(x)$. A major downside of decision trees is their instability with respect to small perturbations of the training data. Slight changes in the training set can lead to completely different tree structures. This affects the validity of their interpretations as well as their generalizability to unseen data.

Imprecise splitting rules We propose a novel method to account for uncertainty in decision tree learning, when partitioning an *ordered* x-dimension. Instead of choosing a binary split we acknowledge our uncertainty about the optimal cutpoint t_0 and consider a symmetric neighbourhood around the candidate cutpoint as equally likely valid split points. As no prior knowledge about the split's distribution is available we choose a non-parametric approach and use the closest points in the covariate space to construct a neighbourhood $\mathscr{T} = \{t_{-k} = x_{-k}, \cdots, t_0, \cdots, t_k = x_k\}$ where t_0 is the candidate split and t_k and t_{-k} the k'th datapoints with higher and lower ordered covariate values. Our approach can be interpreted in two equivalent ways: 1) How do the predictions change if we choose $x \leq t_0 + \varepsilon$ as a slightly different splitting point. 2) How do the predictions change if the x values are slightly perturbed $x - \varepsilon \leq t_0$. This is in contrast to previous imprecise approaches to decision tree learning, where the IDM model is utilized to find more stable *binary* splitting points e.g. [1]. We utilize the model imprecision in several steps. First, when evaluating a potential split, we consider the whole neighborhood instead of a single candidate and calculate an aggregated meassure e.g. the mean purity $\bar{H}(x, y, \mathscr{T}) = \frac{1}{|\mathscr{T}|} \sum_{t \in \mathscr{T}} H(x, y, t)$, where H is the weighted Shannon-Entropy as used in the C4.5 tree inducer [2]. This approach favours regions in the covariate space, that are stable towards small perturbations. Second, when predicting new test cases the decision at each node is no longer binary. At each node for each split in the set a decision is made and the test case moved to the left *and* right childnode with a weight proportional to the number of votes $w_l = \frac{1}{|\mathscr{T}|} \sum_{t \in \mathscr{T}} I(x \leq t)$ and $w_r = (1 - w_l)$ respectively. As a results a test case does not end up in a single terminal node but in each terminal node with varying weights. The tree prediction becomes a set of predicted probabilities $p(y = 1|x)$ (or real values in e.g. regression) instead of a single predicted value. This probability set can be aggregated to a final prediction e.g. with a weighted average. Additionally this approach allows to quantify the spread of the probability set measured through the variance $var(p(y = 1|x))$ as a measure of how much slight changes in the model affect the trees predictions. Third, we can utilize the imprecision in the tree growing process. At each grown node, training observations are moved into both child nodes with weights calculated the same way as above in the prediction case. When learning the next split this procedure allows to use 'close call' observations in both nodes, leading to much more stable tree structures. Empirical results suggest that our approach outperforms traditional decision trees in terms of accuracy and AUC. Moreover the spread of the probability set is a good indicator for the expected loss. Our classifier performs significantly better on the 'sure' predictions which have a low probability spread. This might be very useful in practical applications or when combining several classifiers into learning ensembles.

References

[1] Carlos J Mantas and Joaquín Abellán. Credal-c4. 5: Decision tree based on imprecise probabilities to classify noisy data. *Expert Systems with Applications*, 41(10):4625–4637, 2014.

[2] J Ross Quinlan. *C4. 5: programs for machine learning*. Elsevier, 2014.

© M. Nalenz & T. Augustin.