COMS30035, Machine learning: Combining Models 3, Trees

Edwin Simpson
edwin.simpson@bristol.ac.uk

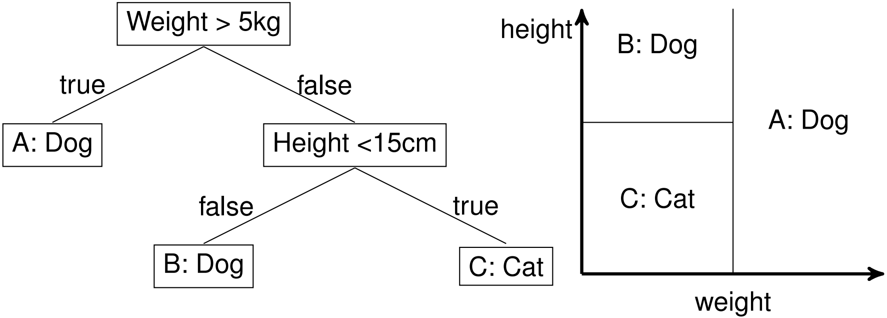
Department of Computer Science, SCEEM
University of Bristol

November 12, 2020

Agenda

- Model Selection
- Model Averaging
- Ensembles: Bagging and Boosting
- Tree-based Models
- Conditional Mixture Models
- Ensembles of Humans

Decision Trees



Decision Trees as Partitioning Input Space

- One model is responsible for assigning a decision for each region of input space;
- The correct model for an input $\boldsymbol{x}$ is chosen by traversing the binary decision tree, following the path from the top to a leaf.
- Leaf node is responsible for assigning a decision, such as a:
  - Class label;
  - Probability distribution over class labels;
  - Scalar value (for regression tasks).
- Mixtures of Experts assign points to regions of input space with soft borders by weighting models probabilistically.

Learning the Tree Structure

- Which input variable to use at each node?
- What threshold to set for the split at each node?
- Classification and Regression Trees (CART): one of many possible learning algorithms
- Objective: minimise the error (regression: sum-of-squares; classification: cross-entropy as used in neural networks or Gini index)

Learning the Tree Structure

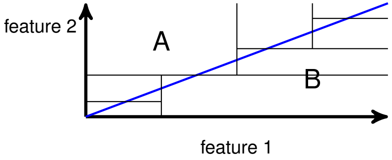
- Number of possible solutions grows combinatorially with the number of input variables
- Greedy algorithm: add nodes one-at-a-time, choosing the best split at each point
    1. Start from the root node
    2. Run *exhaustive search* over each possible variable and threshold for a new node. For each variable and threshold:
        - Compute the average value of the target variable for each leaf of the proposed new node
        - Compute the error if we stop adding nodes here
        - Choose the variable & threshold that minimise the error
        - Add a new node for the chosen variable and threshold.
    3. Repeat step 2 until there are only n data points associated with each leaf node.
    4. Prune back the tree to remove branches that do not reduce error by more than a small tolerance value, ϵ.

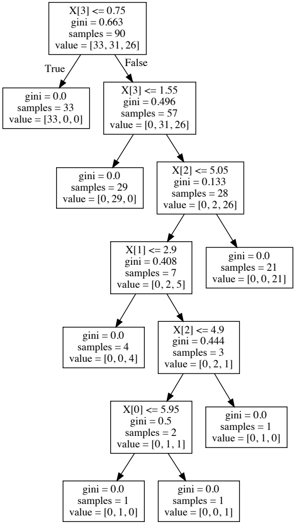
Pruning

- Balance residual training-set error against model complexity
- Start with a tree T_0
- Consider pruning each node in T_0 by combining the branches to obtain tree T
- Compute a criterion $C(T) = \sum_{\tau=1}^{|T|} e_\tau(T) + \lambda|T|$ for each proposed pruning step
- If $C(T) \leq C(T_0)$ keep the pruned tree, else reinstate the pruned node.

Interpretability

- The sequence of decisions is often easier to interpret than other methods (think of neural networks);
- However, sometimes small changes to the dataset cause big changes to the tree;
- If the optimal decision boundary is not aligned with the axes of an input variable, we need a lot of splits.





Random Forest – Adapting Bagging to Trees

- With bagging, base models make similar splits on the same features – the strongest predictors – meaning their errors become correlated
- Random forest modifies the training procedure for each tree, m:
  1. Randomly sample N data points with replacement from a training set with N data points.
  2. Learn the tree using the greedy CART algorithm but when determining each split, consider only $d \ll D$ randomly-chosen features.
- As with bagging, combine predictions by taking mean/majority vote.
- Extremely effective in many applications (see Murphy (2012), Machine Learning: A Probabilistic Perspective, Section 16.2.5)

Now do the quiz!

Please do the quiz for this lecture on Blackboard.