
Pruning is a data compression technique in machine learning and search algorithms that reduces the size of decision trees by removing sections of the tree that are non-critical and redundant to classify instances. Pruning reduces the complexity of the final classifier , and hence improves predictive accuracy by the reduction of overfitting .

One of the questions that arises in a decision tree algorithm is the optimal size of the final tree. A tree that is too large risks overfitting the training data and poorly generalizing to new samples. A small tree might not capture important structural information about the sample space. However, it is hard to tell when a tree algorithm should stop because it is impossible to tell if the addition of a single extra node will dramatically decrease error. This problem is known as the horizon effect . A common strategy is to grow the tree until each node contains a small number of instances then use pruning to remove nodes that do not provide additional information.

Pruning should reduce the size of a learning tree without reducing predictive accuracy as measured by a cross-validation set. There are many techniques for tree pruning that differ in the measurement that is used to optimize performance.

Techniques

Pruning processes can be divided into two types (pre- and post-pruning).

Pre-pruning procedures prevent a complete induction of the training set by replacing a stop () criterion in the induction algorithm (e.g. max. Tree depth or information gain (Attr)> minGain). Pre-pruning methods are considered to be more efficient because they do not induce an entire set, but rather trees remain small from the start. Prepruning methods share a common problem, the horizon effect. This is to be understood as the undesired premature termination of the induction by the stop () criterion.

Post-pruning (or just pruning) is the most common way of simplifying trees. Here, nodes and subtrees are replaced with leaves to reduce complexity. Pruning can not only significantly reduce the size but also improve the classification accuracy of unseen objects. It may be the case that the accuracy of the assignment on the train set deteriorates, but the accuracy of the classification properties of the tree increases overall.

The procedures are differentiated on the basis of their approach in the tree (top-down or bottom-up).

Bottom-up pruning

These procedures start at the last node in the tree (the lowest point). Following recursively upwards, they determine the relevance of each individual node. If the relevance for the classification is not given, the node is dropped or replaced by a leaf. The advantage is that no relevant sub-trees can be lost with this method.

These methods include Reduced Error Pruning (REP), Minimum Cost Complexity Pruning (MCCP), or Minimum Error Pruning (MEP).

Top-down pruning

In contrast to the bottom-up method, this method starts at the root of the tree. Following the structure below, a relevance check is carried out which decides whether a node is relevant for the classification of all n items or not. By pruning the tree at an inner node, it can happen that an entire sub-tree (regardless of its relevance) is dropped. One of these representatives is pessimistic error pruning (PEP), which brings quite good results with unseen items.

Pruning algorithms

Reduced error pruning

One of the simplest forms of pruning is reduced error pruning. Starting at the leaves, each node is replaced with its most popular class. If the prediction accuracy is not affected then the change is kept. While somewhat naive, reduced error pruning has the advantage of simplicity and speed .

Cost complexity pruning

Cost complexity pruning generates a series of trees ■ T 0 … T m {\displaystyle T_{0}\dots T_{m}} ■ where ■ T 0 {\displaystyle T_{0}} ■ is the initial tree and ■ T m {\displaystyle T_{m}} ■ is the root alone. At step ■ i {\displaystyle i} ■ , the tree is created by removing a subtree from tree ■ i − 1 {\displaystyle i-1} ■ and replacing it with a leaf node with value chosen as in the tree building algorithm. The subtree that is removed is chosen as follows:

Define the error rate of tree ■ T {\displaystyle T} ■ over data set ■ S {\displaystyle S} ■ as ■ err ■ (T , S) {\displaystyle \operatorname {err} (T,S)} ■ .

The subtree t {\displaystyle t} that minimizes err ■ (prune ■ (T , t) , S) − err ■ (T , S) | leaves ■ (T) | − | leaves ■ (prune ■ (T , t)) | {\displaystyle {\frac {\operatorname {err} (\operatorname {prune} (T,t),S)-\operatorname {err} (T,S)}{\left\vert \operatorname {leaves} (T)\right\vert -\left\vert \operatorname {leaves} (\operatorname {prune} (T,t))\right\vert }}} is chosen for removal.

The function ■ prune ■ (T , t) {\displaystyle \operatorname {prune} (T,t)} ■ defines the tree obtained by pruning the subtrees ■ t {\displaystyle t} ■ from the tree ■ T {\displaystyle T} ■ . Once the series of trees has been created, the best tree is chosen by generalized accuracy as measured by a training set or cross-validation.

Examples

Pruning could be applied in a compression scheme of a learning algorithm to remove the redundant details without compromising the model's performances. In neural networks, pruning removes entire neurons or layers of neurons.

See also

Alpha–beta pruning

Artificial neural network

Null-move heuristic

Pruning (artificial neural network)

References

Pearl, Judea (1984). Heuristics: Intelligent Search Strategies for Computer Problem Solving . Addison-Wesley. ISBN 978-0-201-05594-8 .

Mansour, Y. (1997). "Pessimistic decision tree pruning based on tree size" . Proc. 14th International Conference on Machine Learning . pp. 195– 201.

Breslow, L. A.; Aha, D. W. (1997). "Simplifying Decision Trees: A Survey". The Knowledge Engineering Review . 12 (1): 1– 47. doi : 10.1017/S0269888997000015 . S2CID 18782652 .

Quinlan, J. R. (1986). "Induction of Decision Trees" . Machine Learning . 1 . Kluwer: 81– 106. doi : 10.1007/BF00116251 .

Further reading

MDL based decision tree pruning Archived 2017-08-29 at the Wayback Machine

Decision tree pruning using backpropagation neural networks

External links

Fast, Bottom-Up Decision Tree Pruning Algorithm

Introduction to Decision tree pruning