Machine Learning (Decision Trees)

Prof K R Chowdhary

CSE Dept., MBM University

December 27, 2024

⇒ The basic idea of data classification: given a training data with known labels or classes (e.g., as shown in Table 1), we would like to learn a model, so that it can be used to predict future data with unknown labels [1].

Table 1: Sample Training database

Record ID	Employment	Age	Salary	Group
1	Self	30	30K	C
2	Industry	35	40K	C
3	Self	35	60K	A
4	Self	30	70K	A
5	Industry	35	40K	C
6	Academia	50	70K	D
7	Self	45	60K	D
8	Academia	30	70K	B
9	Industry	35	60K	B

Introduction to Decision Trees

⇒ As next step, assign each person in mailing list to one of three groups: A, B, C.

⇒ Let the training data with historical information has attributes: $\langle$*Salary, age, employment, group*$\rangle$. Goal is to build a model that takes as input the *predictor attributes* and outputs a value for the *dependent attribute*.

⇒ When the dependent attribute is numerical value, the problem is called *regression*, otherwise it is a *classification* problem.

⇒ In our present discussion, dependent attribute (also called *class labels*) are A, B, C, hence, it is classification problem.

⇒ Let the Table 1 is a sample training database with three predictor attributes: salary, age, and employment, and group as dependent attribute.

Introduction to Decision Trees...

Th classification model Decision trees are popular due to the following reasons:

⇒ Their representation is intuitive, which makes classification model easy to understand,

⇒ An analyst does not need to supply any input parameter for construction of decision trees,

⇒ The accuracy of prediction of decision trees is is better than other classification methods,

⇒ It is possible to construct decision trees from very large training databases using algorithms that are scalable and fast.

Introduction to Decision Trees...

⇒ Decision-trees are tree structures whose leaves are classifications and their branches are conjunctions of features that lead to classifications.

⇒ Their significance is because many data mining methods generate decision trees, which are learned by problem solution methods.

⇒ One approach to learn a decision tree is to split the example set into subsets, based on the value test of some attribute. This is repeated recursively on the subsets, with each split value becoming a sub-tree root.

⇒ Splitting stops when subset becomes so small that further splitting is not possible, i.e., the subset example contains only one classification.

⇒ A split is considered best if it produces minimum number of classification in the subset, i.e., subsequent learning generates smaller sub-trees, which will require less further splitting, in effect reducing the number of steps for solution of the problem.

Decision-tree Algorithm

⇒ A decision tree algorithm comprises two steps: *tree building*, and *tree pruning*.

⇒ In the first step, most decision-tree is grown top-down in a greedy way. Starting with the root node, the database is examined by a method, called, "split selection", it select split condition at each node. Then, database is partitioned and procedure applied recursively.

⇒ In pruning stage, tree constructed in previous phase is pruned to control its size. The pruning methods select the tree in a way that minimizes prediction errors.

⇒ The algorithm 1 is a recursive algorithm that shows a sample tree building phase, and n is the node where tree is to be split. At the output, the algorithm provides decision-tree for data partition D, and node new node value n which is root of the decision tree.

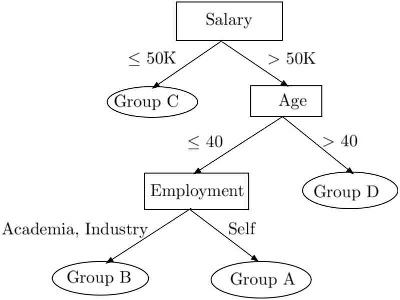
Decision-tree Algorithm

Algorithm 1 Sample Code for building a Decision-Tree

% Input: Data-partition D, Node n, split selection criteria P
% Output: Decision tree for D, with it root as n
Build-Tree(n, D, P)
Apply P to D, and find splitting criteria for node n
if n splits **then**
    Create its child nodes n_1, n_2
    partition D into D_1, D_2 using split criteria
    Build-Tree(n_1, D_1, P)
    Build-tree(n_2, D_2, P)
end if

If the training database does not fit into memory, we need a scalable data access method. Many scalable algorithms have built-in feature, which ensures that only a small set of statistics, are sufficient to implement the split selection.

⇒ The Fig. shows here decision tree for training dataset shown in Table 1. The splitting attributes are: salary, age, and employment, and the class labels are Groups A, B, C, D.



⇒ Every edge that originates from an internal node is labeled with a *splitting predicate*, which involves only the node's splitting attribute. Splitting predicate in this decision tree are: salary $\leq 50K$, $> 50K$, age ≤ 40, > 40, *Self*, and "*Academia, Industry*".

⇒ Property *Splitting predicate*: Any record will take a unique path from the root to exactly one leaf node, which is class label of that record. Combined information at a node about splitting attributes and splitting predicates is *splitting criterion*.

Chowdhary, K.R. (2020). Machine Learning. In: Fundamentals of Artificial Intelligence. Springer, New Delhi. https://doi.org/10.1007/978-81-322-3972-7_13 pp. 393-396.