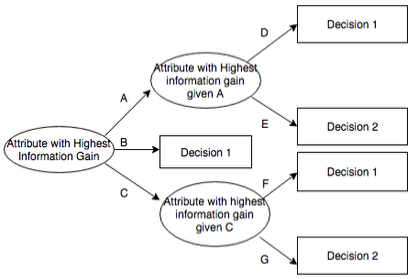
ID3 algorithm

In decision tree learning, **ID3** (**Iterative Dichotomiser 3**) is an algorithm invented by Ross Quinlan[1] used to generate a decision tree from a dataset. ID3 is the precursor to the C4.5 algorithm, and is typically used in the machine learning and natural language processing domains.



Potential ID3-generated decision tree. Attributes are arranged as nodes by ability to classify examples. Values of attributes are represented by branches.

Contents

Algorithm

The ID3 algorithm begins with the original set S as the root node. On each iteration of the algorithm, it iterates through every unused attribute of the set S and calculates the entropy $\mathrm{H}(S)$ or the information gain $IG(S)$ of that attribute. It then selects the attribute which has the smallest entropy (or largest information gain) value. The set S is then split or partitioned by the selected attribute to produce subsets of the data. (For example, a node can be split into child nodes based upon the subsets of the population whose ages are less than 50, between 50 and 100, and greater than 100.) The algorithm continues to recurse on each subset, considering only attributes never selected before.

Recursion on a subset may stop in one of these cases:

- every element in the subset belongs to the same class; in which case the node is turned into a leaf node and labelled with the class of the examples.
- there are no more attributes to be selected, but the examples still do not belong to the same class. In this case, the node is made a leaf node and labelled with the most common class of the examples in the subset.
- there are no examples in the subset, which happens when no example in the parent set was found to match a specific value of the selected attribute. An example could be the absence of a person among the population with age over 100 years. Then a leaf node is

created and labelled with the most common class of the examples in the parent node's set.

Throughout the algorithm, the decision tree is constructed with each non-terminal node (internal node) representing the selected attribute on which the data was split, and terminal nodes (leaf nodes) representing the class label of the final subset of this branch.

Summary

1. Calculate the entropy of every attribute a of the data set S.
2. Partition ("split") the set S into subsets using the attribute for which the resulting entropy after splitting is minimized; or, equivalently, information gain is maximum.
3. Make a decision tree node containing that attribute.
4. Recurse on subsets using the remaining attributes.

Pseudocode

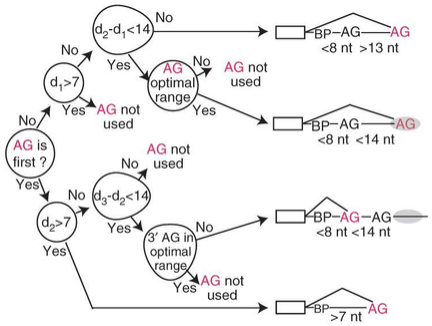
```
ID3 (Examples, Target_Attribute, Attributes)
    Create a root node for the tree
    If all examples are positive, Return the single-node tree Root, with label = +.
    If all examples are negative, Return the single-node tree Root, with label = -.
    If number of predicting attributes is empty, then Return the single node tree Root,
    with label = most common value of the target attribute in the examples.
    Otherwise Begin
        A ← The Attribute that best classifies examples.
        Decision Tree attribute for Root = A.
        For each possible value, v_i, of A,
            Add a new tree branch below Root, corresponding to the test A = v_i.
            Let Examples(v_i) be the subset of examples that have the value v_i for A
            If Examples(v_i) is empty
                Then below this new branch add a leaf node with label = most common target value in the examples
            Else below this new branch add the subtree ID3 (Examples(v_i), Target_Attribute, Attributes - {A})
    End
    Return Root
```

Properties

ID3 does not guarantee an optimal solution. It can converge upon local optima. It uses a greedy strategy by selecting the locally best attribute to split the dataset on each iteration. The algorithm's optimality can be improved by using backtracking during the search for the optimal decision tree at the cost of possibly taking longer.

ID3 can overfit the training data. To avoid overfitting, smaller decision trees should be preferred over larger ones. This algorithm usually produces small trees, but it does not always produce the smallest possible decision tree.

ID3 is harder to use on continuous data than on factored data (factored data has a discrete number of possible values, thus reducing the possible branch points). If the values of any given



ID3-generated decision tree used to determine whether a particular nucleotide pair within a pre-mRNA sequence corresponds to an mRNA splice site. This tree has been shown to have a 95% correct prediction rate.[2]

attribute are continuous, then there are many more places to split the data on this attribute, and searching for the best value to split by can be time consuming.

Usage

The ID3 algorithm is used by training on a data set S to produce a decision tree which is stored in memory. At runtime, this decision tree is used to classify new test cases (feature vectors) by traversing the decision tree using the features of the datum to arrive at a leaf node. The class of this terminal node is the class the test case is classified as.

The ID3 metrics

Entropy

Entropy $\mathrm{H}(S)$ is a measure of the amount of uncertainty in the (data) set S (i.e. entropy characterizes the (data) set S).

$$\mathrm{H}(S) = \sum_{x \in X} -p(x) \log_2 p(x)$$

Where,

- S – The current dataset for which entropy is being calculated
  - This changes at each step of the ID3 algorithm, either to a subset of the previous set in the case of splitting on an attribute or to a "sibling" partition of the parent in case the recursion terminated previously.
- X – The set of classes in S
- $p(x)$ – The proportion of the number of elements in class x to the number of elements in set S

When $\mathrm{H}(S) = 0$, the set S is perfectly classified (i.e. all elements in S are of the same class).

In ID3, entropy is calculated for each remaining attribute. The attribute with the **smallest** entropy is used to split the set S on this iteration. Entropy in information theory measures how much information is expected to be gained upon measuring a random variable; as such, it can also be used to quantify the amount to which the distribution of the quantity's values is unknown. A constant quantity has zero entropy, as its distribution is perfectly known. In contrast, a uniformly distributed random variable (discretely or continuously uniform) maximizes entropy. Therefore, the greater the entropy at a node, the less information is known about the classification of data at this stage of the tree; and therefore, the greater the potential to improve the classification here.

As such, ID3 is a greedy heuristic performing a best-first search for locally optimal entropy values. Its accuracy can be improved by preprocessing the data.

Information gain

Information gain $IG(A)$ is the measure of the difference in entropy from before to after the set S is split on an attribute A. In other words, how much uncertainty in S was reduced after splitting set S on attribute A.

$$IG(S, A) = \mathrm{H}(S) - \sum_{t \in T} p(t)\mathrm{H}(t) = \mathrm{H}(S) - \mathrm{H}(S|A).$$

Where,

- $\mathrm{H}(S)$ – Entropy of set S
- T – The subsets created from splitting set S by attribute A such that $S = \bigcup_{t \in T} t$
- $p(t)$ – The proportion of the number of elements in t to the number of elements in set S
- $\mathrm{H}(t)$ – Entropy of subset t

In ID3, information gain can be calculated (instead of entropy) for each remaining attribute. The attribute with the **largest** information gain is used to split the set S on this iteration.

See also

- Classification and regression tree (CART)
- C4.5 algorithm
- Decision tree learning
  - Decision tree model

References

1. Quinlan, J. R. 1986. Induction of Decision Trees. Mach. Learn. 1, 1 (Mar. 1986), 81–106
2. Taggart, Allison J; DeSimone, Alec M; Shih, Janice S; Filloux, Madeleine E; Fairbrother, William G (2012-06-17). "Large-scale mapping of branchpoints in human pre-mRNA transcripts in vivo" (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3465671). *Nature Structural & Molecular Biology*. **19** (7): 719–721. doi:10.1038/nsmb.2327 (https://doi.org/10.1038%2Fnsmb.2327). ISSN 1545-9993 (https://www.worldcat.org/issn/1545-9993). PMC 3465671 (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3465671). PMID 22705790 (https://pubmed.ncbi.nlm.nih.gov/22705790).

Further reading

- Mitchell, Tom Michael (1997). *Machine Learning*. New York, NY: McGraw-Hill. pp. 55–58. ISBN 0070428077. OCLC 36417892 (https://www.worldcat.org/oclc/36417892).
- Grzymala-Busse, Jerzy W. (February 1993). "Selected Algorithms of Machine Learning from Examples" (https://people.eecs.ku.edu/~jerzygb/j24-sel.pdf) (PDF). *Fundamenta Informaticae*. **18** (2): 193–207 – via ResearchGate.

External links

- Seminars – http://www2.cs.uregina.ca/ (http://www2.cs.uregina.ca/~hamilton/courses/831/notes/ml/dtrees/4_dtrees1.html)
- Description and examples – http://www.cise.ufl.edu/ (http://www.cise.ufl.edu/~ddd/cap6635/Fall-97/Short-papers/2.htm)
- Description and examples – http://www.cis.temple.edu/ (http://www.cis.temple.edu/~ingargio/cis587/readings/id3-c45.html)

- Decision Trees and Political Party Classification (http://jeremykun.com/2012/10/08/decision-trees-and-political-party-classification/)
- An implementation of ID3 in Python (https://github.com/tofti/python-id3-trees)
- An implementation of ID3 in Ruby (http://ai4r.org/machineLearning.html)
- An implementation of ID3 in Common Lisp (https://web.archive.org/web/20171225121804/http://www.pvv.ntnu.no/~oyvinht/static/OSS/cl-id3/)
- An implementation of ID3 algorithm in C# (https://web.archive.org/web/20061109050158/http://www.codeproject.com/cs/algorithms/id3.asp)
- An implementation of ID3 in Perl (https://metacpan.org/module/AI::DecisionTree)
- An implementation of ID3 in Prolog (http://ftp.cs.stanford.edu/cs/robotics/shoham/prolog.tar.Z)
- An implementation of ID3 in C (This code is commented in Italian) (http://id3alg.altervista.org)
- An implementation of ID3 in R (https://ipub.com/id3-with-data-tree/)
- An implementation of ID3 in JavaScript (https://github.com/willkurt/ID3-Decision-Tree)
