



WIKIPEDIA  
The Free Encyclopedia

[Main page](#)

[Contents](#)

[Featured content](#)

[Current events](#)

[Random article](#)

[Donate to Wikipedia](#)

[Wikipedia store](#)

Interaction

[Help](#)

[About Wikipedia](#)

[Community portal](#)

[Recent changes](#)

[Contact page](#)

Tools

[What links here](#)

[Related changes](#)

[Upload file](#)

[Special pages](#)

[Permanent link](#)

[Page information](#)

[Wikidata item](#)

[Cite this page](#)

Print/export

[Create a book](#)

[Download as PDF](#)

[Printable version](#)

Languages

[Deutsch](#)

[Ελληνικά](#)

[Español](#)

[Français](#)

[Italiano](#)

[Nederlands](#)

[日本語](#)

[Русский](#)

[中文](#)

 [Edit links](#)

[Create account](#) [Log in](#)

Article

[Talk](#)

[Read](#)

[Edit](#)

[View history](#)



# ID3 algorithm

From Wikipedia, the free encyclopedia

In [decision tree learning](#), **ID3 (Iterative Dichotomiser 3)** is an [algorithm](#) invented by [Ross Quinlan](#)<sup>[1]</sup> 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.

## Contents [hide]

### 1 Algorithm

#### 1.1 Summary

#### 1.2 Properties

#### 1.3 Usage

### 2 The ID3 metrics

#### 2.1 Entropy

#### 2.2 Information Gain

### 3 See also

### 4 References

### 5 Further Reading

### 6 External links

## Algorithm [edit]

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](#) *H(S)* (or [information gain](#) *IG(A)*) of that attribute. It then selects the attribute which has the smallest entropy (or largest information gain) value. The set *S* is then split by the selected attribute (e.g. age < 50, 50 ≤ age < 100, age ≥ 100) to produce subsets of the data. The algorithm continues to recur 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 (+ or -), then the node is turned into a leaf 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 (some are + and some are -), then the node is turned into a leaf and labelled with the most common class of the examples in the subset
- there are no examples in the subset, this happens when no example in the parent set was found to be matching a specific value of the selected attribute, for example if there was no example with age ≥ 100. Then a leaf is created, and labelled with the most common class of the examples in the parent set.

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

## Summary [edit]

- Calculate the entropy of every attribute using the data set *S*
- Split the set *S* into subsets using the attribute for which entropy is minimum (or, equivalently, information gain is maximum)
- Make a decision tree node containing that attribute
- Recur on subsets using remaining attributes.

## Properties [edit]

ID3 does not guarantee an optimal solution; it can get stuck in local optimums. It uses a greedy approach by selecting the best attribute to split the dataset on each iteration. One improvement that can be made on the algorithm can be to use [backtracking](#) during the search for the optimal decision tree.

ID3 can [overfit](#) to 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 tree.

ID3 is harder to use on continuous data. If the values of any given attribute is 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 [\[edit\]](#)

The ID3 algorithm is used by training on a dataset  $S$  to produce a [decision tree](#) which is stored in memory. At runtime, this decision tree is used to classify new unseen test cases by working down the decision tree using the values of this test case to arrive at a terminal node that tells you what class this test case belongs to.

## The ID3 metrics [\[edit\]](#)

### Entropy [\[edit\]](#)

[Entropy](#)  $H(S)$  is a measure of the amount of uncertainty in the (data) set  $S$  (i.e. entropy characterizes the (data) set  $S$ ).

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

Where,

- $S$  - The current (data) set for which entropy is being calculated (changes every iteration of the ID3 algorithm)
- $X$  - 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  $H(S) = 0$ , 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. The higher the entropy, the higher the potential to improve the classification here.

### Information Gain [\[edit\]](#)

[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  $A$ .

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

Where,

- $H(S)$  - 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  $S$
- $H(t)$  - 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 [\[edit\]](#)

- [CART](#)
- [C4.5 algorithm](#)




## References [\[edit\]](#)

- ↑ Quinlan, J. R. 1986. Induction of Decision Trees. Mach. Learn. 1, 1 (Mar. 1986), 81-106
- ↑ Mitchell, Tom M. *Machine Learning*. McGraw-Hill, 1997. pp. 55–58.
- ↑ Grzymala-Busse, Jerzy W. "Selected Algorithms of Machine Learning from Examples." *Fundamenta Informaticae* 18, (1993): 193–207.

## Further Reading [\[edit\]](#)

- [Decision Trees and Political Party Classification](#) [↗](#)

## External links [\[edit\]](#)

- Seminars - <http://www2.cs.uregina.ca/> 
- Description and examples - <http://www.cise.ufl.edu/> 
- Description and examples - <http://www.cis.temple.edu/> 
- An implementation of ID3 in Python 
- An implementation of ID3 in Ruby 
- An implementation of ID3 in Common Lisp 
- An implementation of ID3 algorithm in C# 
- An implementation of ID3 in Perl 
- An implementation of ID3 in Prolog 
- An implementation of ID3 in C (This code is commented in Italian) 
- An implementation of ID3 in R 

Categories: [Decision trees](#) | [Classification algorithms](#)

This page was last modified on 26 July 2015, at 23:19.

Text is available under the [Creative Commons Attribution-ShareAlike License](#); additional terms may apply. By using this site, you agree to the [Terms of Use](#) and [Privacy Policy](#). Wikipedia® is a registered trademark of the [Wikimedia Foundation, Inc.](#), a non-profit organization.

[Privacy policy](#) [About Wikipedia](#) [Disclaimers](#) [Contact Wikipedia](#) [Developers](#) [Mobile view](#)

