C4.5 algorithm

From 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
Español
Français
日本語
Русский
Српски / srpski
中文
Edit links

This article includes a list of references, but **its sources remain unclear** because it has **insufficient inline citations**. Please help to improve this article by introducing more precise citations. *(July 2008)*

C4.5 is an algorithm used to generate a decision tree developed by Ross Quinlan.[1] C4.5 is an extension of Quinlan's earlier ID3 algorithm. The decision trees generated by C4.5 can be used for classification, and for this reason, C4.5 is often referred to as a statistical classifier.

It became quite popular after ranking #1 in the *Top 10 Algorithms in Data Mining* pre-eminent paper published by Springer LNCS in 2008.[2]

Contents [hide]

Algorithm [edit]

C4.5 builds decision trees from a set of training data in the same way as ID3, using the concept of information entropy. The training data is a set $S = s_1, s_2, ...$ of already classified samples. Each sample s_i consists of a p-dimensional vector $(x_{1,i}, x_{2,i}, ..., x_{p,i})$, where the x_j represent attributes or features of the sample, as well as the class in which s_i falls.

At each node of the tree, C4.5 chooses the attribute of the data that most effectively splits its set of samples into subsets enriched in one class or the other. The splitting criterion is the normalized information gain (difference in entropy). The attribute with the highest normalized information gain is chosen to make the decision. The C4.5 algorithm then recurs on the smaller sublists.

This algorithm has a few base cases.

- All the samples in the list belong to the same class. When this happens, it simply creates a leaf node for the decision tree saying to choose that class.
- None of the features provide any information gain. In this case, C4.5 creates a decision node higher up the tree using the expected value of the class.
- Instance of previously-unseen class encountered. Again, C4.5 creates a decision node higher up the tree using the expected value.

Pseudocode [edit]

In pseudocode, the general algorithm for building decision trees is:[3]

1. Check for base cases
2. For each attribute *a*
    1. Find the normalized information gain ratio from splitting on *a*
3. Let *a_best* be the attribute with the highest normalized information gain
4. Create a decision *node* that splits on *a_best*
5. Recur on the sublists obtained by splitting on *a_best*, and add those nodes as children of *node*

Implementations [edit]

J48 is an open source Java implementation of the C4.5 algorithm in the weka data mining tool.

Improvements from ID.3 algorithm [edit]

C4.5 made a number of improvements to ID3. Some of these are:

- Handling both continuous and discrete attributes - In order to handle continuous attributes, C4.5 creates a threshold and then splits the list into those whose attribute value is above the threshold and those that are less than or equal to it.[4]
- Handling training data with missing attribute values - C4.5 allows attribute values to be marked as ? for missing. Missing attribute values are simply not used in gain and entropy calculations.
- Handling attributes with differing costs.
- Pruning trees after creation - C4.5 goes back through the tree once it's been created and attempts to remove branches that do not help by replacing them with leaf nodes.

Improvements in C5.0/See5 algorithm [edit]

The **neutrality of this section is disputed**. Relevant discussion may be found on the talk page. Please do not remove this message until the dispute is resolved. *(August 2011)*

Quinlan went on to create C5.0 and See5 (C5.0 for Unix/Linux, See5 for Windows) which he markets commercially. C5.0 offers a number of improvements on C4.5. Some of these are:[5][6]

- Speed - C5.0 is significantly faster than C4.5 (several orders of magnitude)
- Memory usage - C5.0 is more memory efficient than C4.5
- Smaller decision trees - C5.0 gets similar results to C4.5 with considerably smaller decision trees.
- Support for boosting - Boosting improves the trees and gives them more accuracy.
- Weighting - C5.0 allows you to weight different cases and misclassification types.
- Winnowing - a C5.0 option automatically winnows the attributes to remove those that may be unhelpful.

Source for a single-threaded Linux version of C5.0 is available under the GPL.

See also [edit]

- ID3 algorithm

References [edit]

1. ^ Quinlan, J. R. C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, 1993.
2. ^ Umd.edu - Top 10 Algorithms in Data Mining
3. ^ S.B. Kotsiantis, Supervised Machine Learning: A Review of Classification Techniques, Informatica 31(2007) 249-268, 2007
4. ^ J. R. Quinlan. Improved use of continuous attributes in c4.5. Journal of Artificial Intelligence Research, 4:77-90, 1996.
5. ^ Is See5/C5.0 Better Than C4.5?
6. ^ M. Kuhn and K. Johnson, Applied Predictive Modeling, Springer 2013

External links [edit]

- Original implementation on Ross Quinlan's homepage: http://www.rulequest.com/Personal/
- See5 and C5.0

Categories: Classification algorithms | Decision trees