1. Introduction

Thuật toán decision tree là phương pháp tiếp cận top-down, bắt đầu với training set và labels của chúng. Training set được partitioned recursively tới những subsets.

Có 3 tham số chính:

- D (data partition). Ở thời điểm ban đầu, D chứa toàn bộ training tuples.

- attribute\_list: danh sách attributes mô tả tuples.

- attribute\_selection\_method: Phương pháp chọn attribute (splitting attribute) cho node

Về cơ bản các thuật toán decision tree khác biệt ở phương pháp chọn attribute cho node

2. Attribute Selection Measure

a) Phương pháp dùng để chọn ra splitting attribute chia data partition D thành những individual classes. Nếu chúng ta chia D thành những partitions theo splitting attribute, lý tưởng là mỗi partition đó có độ thuần khiết cao nhất có thể (ex: tất cả tuples thuộc về 1 partition cùng 1 class). Có thể hiểu Attribute selection measure là splitting rules bởi vì chúng xác định cách mà những tuples tại một node được chia.

Table

Description automatically generated

Diagram

Description automatically generated

b) Gini Index được sử dụng bởi CART.

Gini index đo độ impurity của D:

A picture containing text, watch, clock

Description automatically generated

pi là xác suất một tuple của D thuộc về class Ci, và được tinh bằng: |Ci,D|/|D|. Tổng được tính qua tất cả class

Với mỗi node (attribute), Gini index chia partition tại node đó thành 2 partitions (binary split). Giả sử attribute A là một attribute có giá trị rời rạc (discrete-valued) có v giá trị phân biệt {a1, a2,..., av). Để xác định được điểm chia tốt nhất trên A, chúng ta sẽ phải xem xét qua tất cả subset khả thi. Nếu A có v giá trị phân biệt, thì sẽ có 2^v subset. Ví dụ: nếu income attribute có 3 giá trị {low, medium, hight}, thì chúng ta sẽ có những subset sau: {low, medium, high}, {low, medium}, {low, high}, {medium, high}, {low}, {medium}, {high}, và {}. 2 subset {low, mediumm high} và 1 subset trống có thể bỏ qua vì chúng không chia tuples. Vì thế có tổng cộng 2^v - 2 cách để chia D thành 2 partitions dựa trên binary split trên A.

Với mỗi attribute, chúng ta cần xem xét tất cả binary split (tính tổng impurity cho mỗi kết quả ứng với mỗi binary split).

Ví dụ, giả sử subset S chia D thành 2 phần D1, D2, Gini index sẽ là:

Text

Description automatically generated with low confidence

Subset mang lại Gini index nhỏ nhất (impurity nhỏ nhất) sẽ được chọn làm splitting subset cho attribute đó

Redution in impurition (độ giảm tạp chất) của 1 attribute A nếu sử dụng nó làm splitting attribute đươc tính bằng công thức:

Logo, company name

Description automatically generated

Attribute nào đạt được tối đa redution in impurity sẽ được chọn làm splitting attributes.

3. Feature importance with decision tree

Cách scikitlearn sử dụng decision tree cho bài toán Feature importance đươc mô tả trong document:

“The feature importances. The higher, the more important the feature. The importance of a feature is computed as the (normalized) total reduction of the criterion brought by that feature. It is also known as the Gini importance”

Như vậy importance của 1 featuer được tính bằng tổng độ giảm tạp chất bởi feature đó, hay còn được biết là Gini importance.

The weighted impurity decrease equation is the following:

*N\_t / N \* (impurity - N\_t\_R / N\_t \* right\_impurity*

*- N\_t\_L / N\_t \* left\_impurity)*

Trong đó:

* N là tổng số lượng samples
* N\_t là tổng số lượng samples tại node hiện tại
* N\_t\_L là tổng số lượng samples ở node trái
* N\_t\_R là tổng số lượng samples ở node phải
* Impurity là độ tạp chất của node

Xét ví dụ sau: bộ dữ liệu gồm 4 samples với 3 thuộc tính thuộc về 2 class 1 hoặc 0.

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 |

Decision Tree:

Diagram

Description automatically generated

Theo cách tính feature importance ta đã nói ở trên, ta có:

For X[2] :

feature\_importance = (4 / 4) \* (0.375 - (0.75 \* 0.444)) = 0.042

For X[1] :

feature\_importance = (3 / 4) \* (0.444 - (2/3 \* 0.5)) = 0.083

For X[0] :

feature\_importance = (2 / 4) \* (0.5) = 0.25

Reference:

<https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html#sklearn.tree.DecisionTreeClassifier>

<https://stackoverflow.com/questions/49170296/scikit-learn-feature-importance-calculation-in-decision-trees>

Book: Data mining: Concepts and Techniques by Jiawei Han