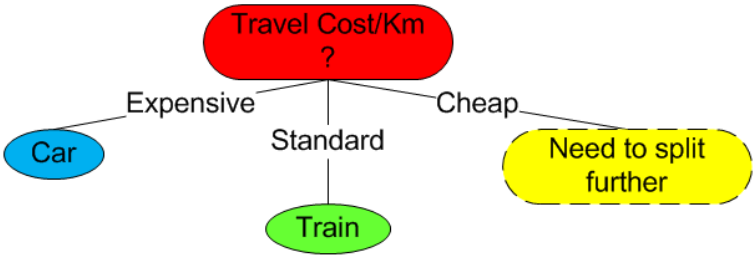
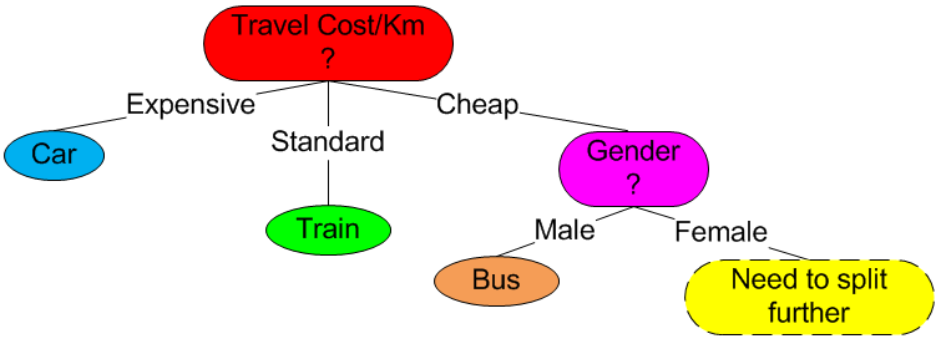
Kardi Teknomo

DECISION TREE TUTORIAL





Revoledu.com

Decision Tree Tutorial by Kardi Teknomo

Published by Revoledu.com
Online edition is available at Revoledu.com
Last Update: October 2012

Table of Contents

Decision tree is a popular classifier that does not require any knowledge or parameter setting. The approach is supervised learning. Given a training data, we can induce a decision tree. From a decision tree we can easily create rules about the data. Using decision tree, we can easily predict the classification of unseen records. In this decision tree tutorial, you will learn how to use, and how to build a decision tree in a very simple explanation.

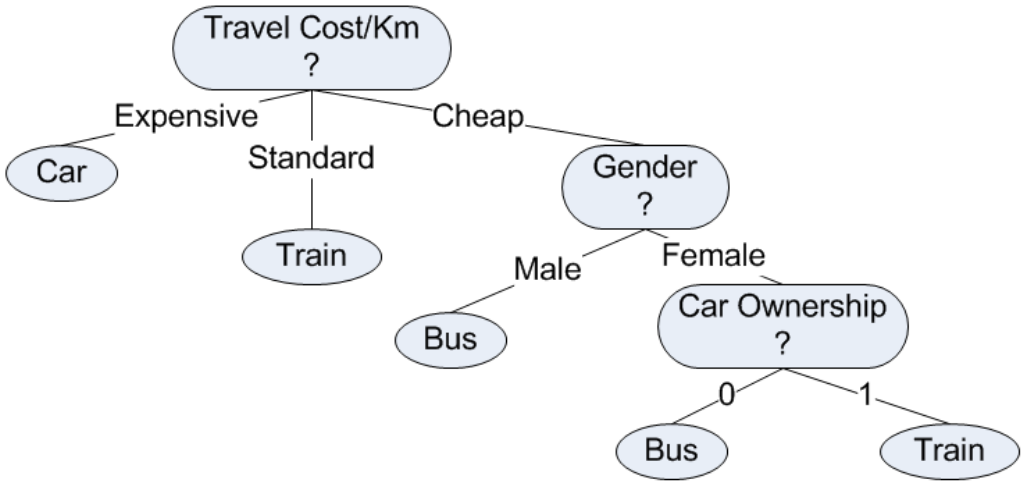
What is Decision Tree?

Decision tree is a hierarchical tree structure that used to classify classes based on a series of questions (or rules) about the attributes of the class. The attributes of the classes can be any type of variables from binary, nominal, ordinal, and quantitative values, while the classes must be qualitative type (categorical or binary, or ordinal). In short, given a data of attributes together with its classes, a decision tree produces a sequence of rules (or series of questions) that can be used to recognize the class.

Let us start with an example. Throughout this tutorial, we will use the following 10 training data. The training data is supposed to be a part of a transportation study regarding mode choice to select Bus, Car or Train among commuters along a major route in a city, gathered through a questionnaire study. The data have 4 attributes which I selected for the sake of clarity. Attribute gender is binary type, car ownership is quantitative integer (thus behave like nominal). Travel cost/km is quantitative of ratio type but in here I put into ordinal type (at later section of this tutorial, I will discuss how to split quantitative data into qualitative) and income level is also an ordinal type.

Attributes				Classes
Gender	Car ownership	Travel Cost ($)/km	Income Level	Transportation mode
Male	0	Cheap	Low	Bus
Male	1	Cheap	Medium	Bus
Female	1	Cheap	Medium	Train
Female	0	Cheap	Low	Bus
Male	1	Cheap	Medium	Bus
Male	0	Standard	Medium	Train
Female	1	Standard	Medium	Train
Female	1	Expensive	High	Car
Male	2	Expensive	Medium	Car
Female	2	Expensive	High	Car

Based on above training data, we can induce a decision tree as the following:



Notice that attribute “income level” is not included in the decision tree because based on the given data attribute “travel cost per km” would produce better classification than “income level”. We will see later how the decision is generated. In the next section, I will discuss how to use a decision tree to predict unseen record.

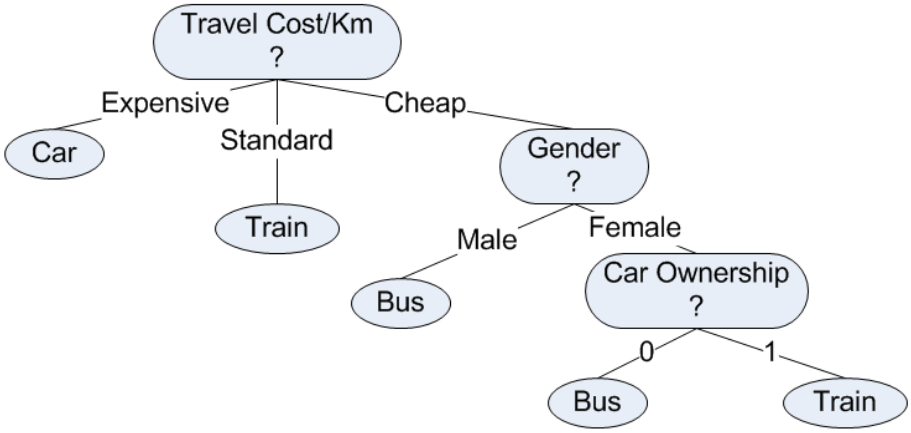
How to use a decision tree?

Decision tree can be used to predict a pattern or to classify the class of a data. Suppose we have new unseen records of a person from the same location where the data sample was taken. The following data are called *test data* (in contrast to *training data*) because we would like to examine the classes of these data.

Person name	Gender	Car ownership	Travel Cost ($)/km	Income Level	Transportation Mode
Alex	Male	1	Standard	High	?
Buddy	Male	0	Cheap	Medium	?
Cherry	Female	1	Cheap	High	?

The question is what transportation mode would Alex, Buddy and Cheery use? Using the decision tree that we have generated in the previous section, we will use deductive approach to classify whether a person will use car, train or bus as his or her mode along a major route in that city, based on the given attributes.

We can start from the root node which contains an attribute of Travel cost per km. If the travel cost per km is expensive, the person uses car. If the travel cost per km is standard price, the person uses train. If the travel cost is cheap, the decision tree needs to ask next question about the gender of the person. If the person is a male, then he uses bus. If the gender is female, the decision tree needs to ask again on how many cars she own in her household. If she has no car, she uses bus, otherwise she uses train.



The rules generated from the decision tree above are mutually exclusive and exhaustive for each class label on the leaf node of the tree:

Rule 1: If Travel cost/km is expensive then mode = car

Rule 2: If Travel cost/km is standard then mode = train

Rule 3: If Travel cost/km is cheap and gender is male then mode = bus

Rule 4: If Travel cost/km is cheap and gender is female and she owns no car then mode = bus

Rule 5: If Travel cost/km is cheap and gender is female and she owns 1 car then mode = train

Based on the rules or decision tree above, the classification is very straightforward. Alex is willing to pay standard travel cost per km, thus regardless his other attributes, his transportation mode must be train. Buddy is only willing to pay cheap travel cost per km, and his gender is male, thus his selection of transportation mode should be bus. Cherry is also willing to pay cheap travel cost per km, and her gender is female and actually she owns a car, thus her transportation mode choice to work is train (probably she uses car only during weekend to shop). Variable Income level never be utilized to classify the transportation mode in this case.

Person name	Travel Cost ($)/km	Gender	Car ownership	Transportation Mode
Alex	Standard	Male	1	Train
Buddy	Cheap	Male	0	Bus
Cherry	Cheap	Female	1	Train

Though decision tree is very powerful method, at this point, I shall give several notes to the readers in decision tree utilization. First, it must be noted, however, that with limited number of training data (only 10) that induce the decision tree, we cannot generalize the rules of the decision tree above to be applicable for other cases in your city. The decision

tree above is only true for the cases on the given data, which is only for the particular major route in that city where the data was gathered.

The sequence of rules generated by the decision tree is based on priority of the attributes. For example, there is no rule for people who own more than 1 car because based on the data it is already covered by attribute travel cost/km. For those who own 2 cars the travel cost/km are always expensive, thus the mode is car.

Due to the limitation of decision algorithm (most algorithms of decision tree employ greedy strategy with no backtracking thus it is not exhaustive search), these sequences of priority in general is not optimum. We cannot say that the rules generated by decision tree are the best rules.

In the next section, you will learn more detail on how to generate a decision tree.

How to generate a decision tree?

In this section, you will learn how to generate a decision tree. This approach is sometimes called *decision tree inductive* because the decision tree are build based on data. I will show the manual computation step by step such that you can check using calculator or spreadsheet.

Before I discuss about decision tree algorithm, it would be better if you familiar yourself with several measures of impurity. Therefore, the topics in this section are:

- How to measure impurity?
  - Entropy
  - Gini Index
  - Classification error
- How a decision tree algorithm work?
- Improvement through gain ratio

How to measure impurity?

Given a data table that contains attributes and class of the attributes, we can measure homogeneity (or heterogeneity) of the table based on the classes. We say a table is pure or homogenous if it contains only a single class. If a data table contains several classes, then we say that the table is impure or heterogeneous. There are several indices to measure degree of impurity quantitatively. Most well known indices to measure degree of impurity are entropy, gini index, and classification error. The formulas are given below

$$Entropy = \sum_j -p_j log_2 p_j$$

$$Gini\ Index = 1 - \sum_j p_j^2$$

$$Classification\ Error = 1 - max\{p_j\}$$

All above formulas contain values of probability p_j of a class j.

In our example, the classes of Transportation mode below consist of three groups of Bus, Car and Train. In this case, we have 4 buses, 3 cars and 3 trains (in short we write as 4B, 3C, 3T). The total data is 10 rows.

Attributes				Classes
Gender	**Car ownership**	**Travel Cost ($)/km**	**Income Level**	**Transportation mode**
Male	0	Cheap	Low	Bus
Male	1	Cheap	Medium	Bus
Female	0	Cheap	Low	Bus
Male	1	Cheap	Medium	Bus
Female	1	Expensive	High	Car
Male	2	Expensive	Medium	Car
Female	2	Expensive	High	Car
Female	1	Cheap	Medium	Train
Male	0	Standard	Medium	Train
Female	1	Standard	Medium	Train

Based on these data, we can compute probability of each class. Since probability is equal to frequency relative, we have

Prob (Bus) = 4 / 10 = 0.4
Prob (Car) = 3 / 10 = 0.3
Prob (Train) = 3 / 10 = 0.3

Observe that when to compute probability, we only focus on the *classes*, not on the *attributes*. Having the probability of each class, now we are ready to compute the quantitative indices of impurity degrees.

Entropy

One way to measure impurity degree is using entropy.

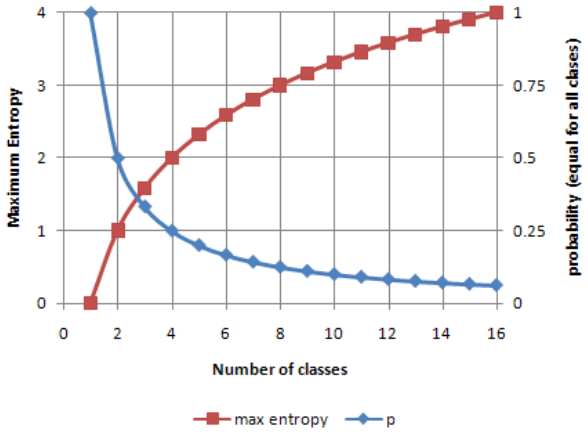
$$Entropy = \sum_j -p_j log_2 p_j$$

Example: Given that Prob (Bus) = 0.4, Prob (Car) = 0.3 and Prob (Train) = 0.3, we can now compute entropy as

Entropy = – 0.4 log (0.4) – 0.3 log (0.3) – 0.3 log (0.3) = 1.571

The logarithm is base 2.

Entropy of a pure table (consist of single class) is zero because the probability is 1 and log (1) = 0. Entropy reaches maximum value when all classes in the table have equal probability. Figure below plots the values of maximum entropy for different number of classes n, where probability is equal to p=1/n. I this case, maximum entropy is equal to -n*p*log p. Notice that the value of entropy is larger than 1 if the number of classes is more than 2.



Gini Index

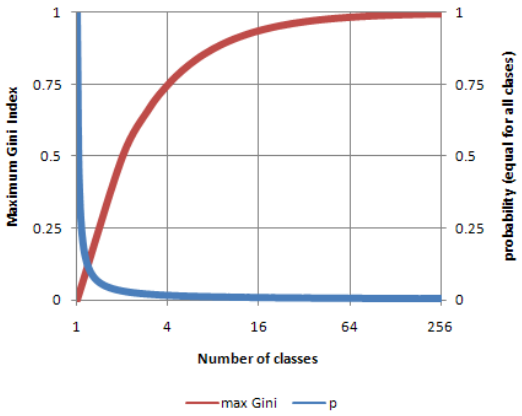
Another way to measure impurity degree is using Gini index.

$$Gini\ Index = 1 - \sum_j p_j^2$$

Example: Given that Prob (Bus) = 0.4, Prob (Car) = 0.3 and Prob (Train) = 0.3, we can now compute Gini index as

Gini Index = 1 – (0.4^2 + 0.3^2 + 0.3^2) = 0.660

Gini index of a pure table (consist of single class) is zero because the probability is 1 and 1-(1)^2 = 0. Similar to Entropy, Gini index also reaches maximum value when all classes in the table have equal probability. Figure below plots the values of maximum gini index for different number of classes n, where probability is equal to p=1/n. Notice that the value of Gini index is always between 0 and 1 regardless the number of classes.



Classification error

Still another way to measure impurity degree is using index of classification error

$$Classification\ Error = 1 - max\{p_j\}$$

Example: Given that Prob (Bus) = 0.4, Prob (Car) = 0.3 and Prob (Train) = 0.3, index of classification error is given as

Classification Error Index = 1 – Max{0.4, 0.3, 0.3} = 1 - 0.4 = 0.60

Similar to Entropy and Gini Index, Classification error index of a pure table (consist of single class) is zero because the probability is 1 and 1-max(1) = 0. The value of classification error index is always between 0 and 1. In fact the maximum Gini index for a given number of classes is always equal to the maximum of classification error index because for a number of classes n, we set probability is equal to p=1/n and maximum Gini index happens at 1-n*(1/n)^2 = 1-1/n, while maximum classification error index also happens at 1-max{1/n} =1-1/n.

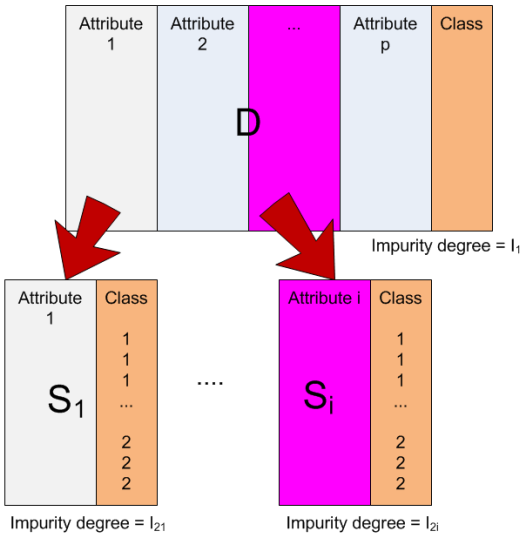
Knowing how to compute degree of impurity, now we are ready to proceed with decision tree algorithm that I will explain in the next section.

Decision Tree Algorithm

There are several most popular decision tree algorithms such as ID3, C4.5 and CART (classification and regression trees). In general, the actual decision tree algorithms are recursive. (For example, it is based on a greedy recursive algorithm called Hunt algorithm that uses only local optimum on each call without backtracking. The result is not optimum

but very fast). For clarity, however, in this tutorial, I will describe as if the algorithm is iterative.

Here is an explanation on how a decision tree algorithm work. We have a data record which contains attributes and the associated classes. Let us call this data as table D. From table D, we take out each attribute and its associate classes. If we have p attributes, then we will take out p subset of D. Let us call these subsets as S_i. Table D is the parent of table S_i.



From table D and for each associated subset S_i, we compute degree of impurity. We have discussed about how to compute these indices in the previous section.

To compute the degree of impurity, we must distinguish whether it is come from the parent table D or it come from a subset table S_i with attribute i.

If the table is a parent table D, we simply compute the number of records of each class. For example, in the parent table below, we can compute degree of impurity based on transportation mode. In this case we have 4 Busses, 3 Cars and 3 Trains (in short 4B, 3C, 3T):

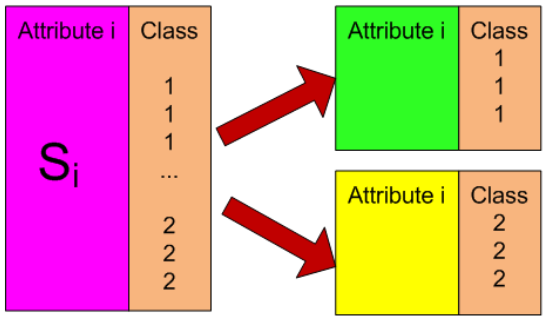
Data

Attributes				Classes
Gender	Car ownership	Travel Cost ($)/km	Income Level	Transportation mode
Male	0	Cheap	Low	Bus
Male	1	Cheap	Medium	Bus
Female	1	Cheap	Medium	Train
Female	0	Cheap	Low	Bus
Male	1	Cheap	Medium	Bus
Male	0	Standard	Medium	Train
Female	1	Standard	Medium	Train
Female	1	Expensive	High	Car
Male	2	Expensive	Medium	Car
Female	2	Expensive	High	Car

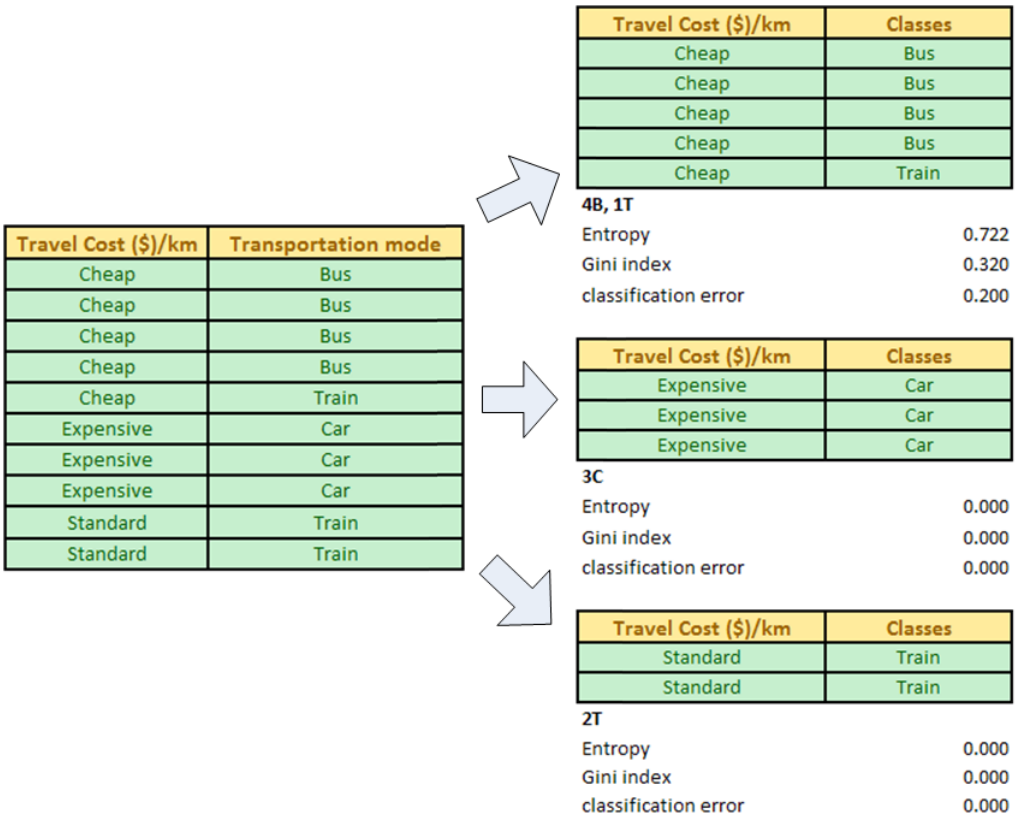
4B, 3C, 3T

Entropy	1.571
Gini index	0.660
Classification error	0.600

If the table is a subset of attribute table S_i, we need to separate the computation of impurity degree for each value of the attribute i.



For example, attribute Travel cost per km has three values: Cheap, Standard and Expensive. Now we sort the table Si = [Travel cost/km, Transportation mode] based on the values of Travel cost per km. Then we separate each value of the travel cost and compute the degree of impurity (either using entropy, gini index or classification error).



Information gain

The reason for different ways of computation of impurity degrees between data table D and subset table S_i is because we would like to compare the difference of impurity degrees *before* we split the table (i.e. data table D) and *after* we split the table according to the values of an attribute i (i.e. subset table Si) . The measure to compare the difference of impurity degrees is called **information gain**. We would like to know what our gain is if we split the data table based on some attribute values.

Information gain is computed as impurity degrees of the parent table and weighted summation of impurity degrees of the subset table. The weight is based on the number of records for each attribute values. Suppose we will use entropy as measurement of impurity degree, then we have:

Information gain (i) = Entropy of parent table D – Sum (n_k/n * Entropy of each value k of subset table S_i)

For example, our data table D has classes of 4B, 3C, 3T which produce entropy of 1.571. Now we try the attribute Travel cost per km which we split into three: Cheap that has classes of 4B, 1T (thus entropy of 0.722), Standard that has classes of 2T (thus entropy =

0 because pure single class) and Expensive with single class of 3C (thus entropy also zero).

The information gain of attribute Travel cost per km is computed as 1.571 – (5/10 * 0.722+2/10*0+3/10*0) = 1.210

You can also compute information gain based on Gini index or classification error in the same method. The results are given below.

Gain of Travel Cost/km (multiway) based on	
Entropy	1.210
Gini index	0.500
classification error	0.500

For each attribute in our data, we try to compute the information gain. The illustration below shows the computation of information gain for the first iteration (based on the data table) for other three attributes of Gender, Car ownership and Income level.

Subset

Gender	Classes
Female	Bus
Female	Car
Female	Car
Female	Train
Female	Train

1B, 2C, 2T

Entropy	1.522
Gini index	0.640
classification error	0.600

Gender	Classes
Male	Bus
Male	Bus
Male	Bus
Male	Car
Male	Train

3B, 1C, 1T

Entropy	1.371
Gini index	0.560
classification error	0.400

Gain of Gender based on

Entropy	0.125
Gini index	0.060
classification error	0.100

Car ownership	Classes
0	Bus
0	Bus
0	Train

2B, 1T

Entropy	0.918
Gini index	0.444
classification error	0.333

Car ownership	Classes
1	Bus
1	Bus
1	Car
1	Train
1	Train

2B, 1C, 2T

Entropy	1.522
Gini index	0.640
classification error	0.600

Car ownership	Classes
2	Car
2	Car

2C

Entropy	0.000
Gini index	0.000
classification error	0.000

Gain of Car ownership (multiway) based on

Entropy	0.534
Gini index	0.207
classification error	0.200

Income Level	Classes
High	Car
High	Car

2C

Entropy	0.000
Gini index	0.000
classification error	0.000

Income Level	Classes
Low	Bus
Low	Bus

2B

Entropy	0.000
Gini index	0.000
classification error	0.000

Income Level	Classes
Medium	Bus
Medium	Bus
Medium	Car
Medium	Train
Medium	Train
Medium	Train

2B, 1C, 3T

Entropy	1.459
Gini index	0.611
classification error	0.500

Gain of Income Level (multiway) based on

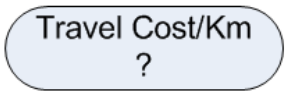
Entropy	0.695
Gini index	0.293
classification error	0.300

Table below summarizes the information gain for all four attributes. In practice, you don't need to compute the impurity degree based on three methods. You can use either one of Entropy or Gini index or index of classification error.

Results of first Iteration

Gain	Gender	Car ownership	Travel Cost/KM	Income Level
Entropy	0.125	0.534	**1.210**	0.695
Gini index	0.060	0.207	**0.500**	0.293
Classification error	0.100	0.200	**0.500**	0.300

Once you get the information gain for all attributes, then we find the optimum attribute that produce the maximum information gain (i* = argmax {information gain of attribute i}). In our case, travel cost per km produces the maximum information gain. We put this optimum attribute into the node of our decision tree. As it is the first node, then it is the root node of the decision tree. Our decision tree now consists of a single root node.



Once we obtain the optimum attribute, we can split the data table according to that optimum attribute. In our example, we split the data table based on the value of travel cost per km.

Data

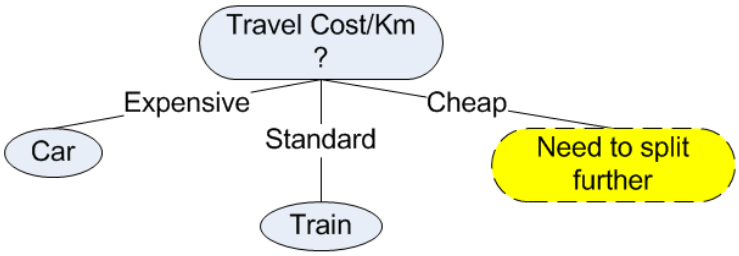
Attributes				Classes
Gender	Car	Travel Cost	Income Level	Transportation
Male	0	Cheap	Low	Bus
Male	1	Cheap	Medium	Bus
Female	0	Cheap	Low	Bus
Male	1	Cheap	Medium	Bus
Female	1	Cheap	Medium	Train
Female	1	Expensive	High	Car
Male	2	Expensive	Medium	Car
Female	2	Expensive	High	Car
Male	0	Standard	Medium	Train
Female	1	Standard	Medium	Train

Attributes				Classes
Gender	Car ownership	Travel Cost /km	Income Level	Transportation mode
Female	0	Cheap	Low	Bus
Male	0	Cheap	Low	Bus
Male	1	Cheap	Medium	Bus
Male	1	Cheap	Medium	Bus
Female	1	Cheap	Medium	Train

Attributes				Classes
Gender	Car ownership	Travel Cost /km	Income Level	Transportation mode
Female	1	Expensive	High	Car
Female	2	Expensive	High	Car
Male	2	Expensive	Medium	Car

Attributes				Classes
Gender	Car ownership	Travel Cost /km	Income Level	Transportation mode
Female	1	Standard	Medium	Train
Male	0	Standard	Medium	Train

After the split of the data, we can see clearly that value of Expensive travel cost/km is associated only with pure class of Car while Standard travel cost/km is only related to pure class of Train. Pure class is always assigned into leaf node of a decision tree. We can use this information to update our decision tree in our first iteration into the following.



For Cheap travel cost/km, the classes are not pure, thus we need to split further in the next iteration.

Second Iteration

In the second iteration, we need to update our data table. Since Expensive and Standard travel cost/km have been associated with pure class, we do not need these data any longer. For second iteration, our data table D is only come from the Cheap Travel cost/km. We remove attribute travel cost/km from the data because they are equal and redundant.

Attributes				Classes
Gender	Car ownership	Travel Cost /km	Income Level	Transportation mode
Female	0	Cheap	Low	Bus
Male	0	[illegible]	Low	Bus
Male	1	[illegible]	Medium	Bus
Male	1	[illegible]	Medium	Bus
Female	1	Cheap	Medium	Train

Data

Attributes			Classes
Gender	Car ownership	Income Level	Transportation mode
Female	0	Low	Bus
Male	0	Low	Bus
Male	1	Medium	Bus
Male	1	Medium	Bus
Female	1	Medium	Train

Now we have only three attributes: Gender, car ownership and Income level. The degree of impurity of the data table D is shown in the picture below.

Data second iteration

Attributes			Classes
Gender	Car ownership	Income Level	Transportation mode
Female	0	Low	Bus
Male	0	Low	Bus
Male	1	Medium	Bus
Male	1	Medium	Bus
Female	1	Medium	Train

4B, 1T

Entropy	0.722
Gini index	0.320
classification error	0.200

Then, we repeat the procedure of computing degree of impurity and information gain for the three attributes. The results of computation are exhibited below.

Subsets of second iterations

Gender	Classes
Female	Bus
Female	Train

1B, 1T

Entropy	1.000
Gini index	0.500
classification	0.500

Car ownership	Classes
0	Bus
0	Bus

2B

Entropy	0.000
Gini index	0.000
classification error	0.000

Income Level	Classes
Low	Bus
Low	Bus

2B

Entropy	0.000
Gini index	0.000
classification error	0.000

Gender	Classes
Male	Bus
Male	Bus
Male	Bus

3B

Entropy	0.000
Gini index	0.000
classification	0.000

Car ownership	Classes
1	Bus
1	Bus
1	Train

2B, 1T

Entropy	0.918
Gini index	0.444
classification error	0.333

Income Level	Classes
Medium	Bus
Medium	Bus
Medium	Train

2B, 1T

Entropy	0.918
Gini index	0.444
classification error	0.333

Gain of Gender based on

Entropy	0.322
Gini index	0.120
classification	0.000

Gain of Car ownership based on

Entropy	0.171
Gini index	0.053
classification error	0.000

Gain of Income Level based on

Entropy	0.171
Gini index	0.053
classification error	0.000

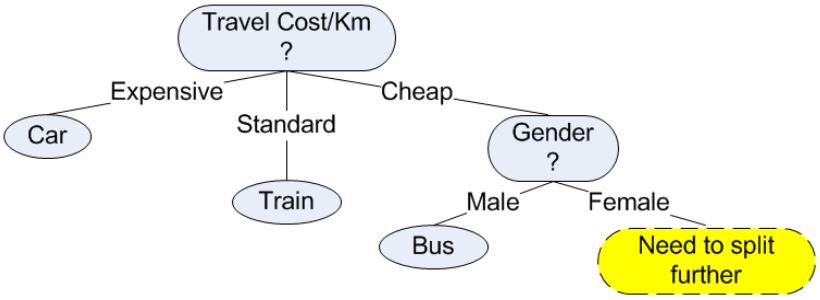
The maximum gain is obtained for the optimum attribute Gender. Once we obtain the optimum attribute, the data table is split according to that optimum attribute. In our case, Male Gender is only associated with pure class Bus, while Female still need further split of attribute.

Attributes			Classes
Gender	Car ownership	Income Level	Transportation mode
Female	0	Low	Bus
Female	1	Medium	Train
Male	0	Low	Bus
Male	1	Medium	Bus
Male	1	Medium	Bus

Attributes			Classes
Gender	Car ownership	Income Level	Transportation mode
Female	0	Low	Bus
Female	1	Medium	Train

Attributes			Classes
Gender	Car ownership	Income Level	Transportation mode
Male	0	Low	Bus
Male	1	Medium	Bus
Male	1	Medium	Bus

Using this information, we can now update our decision tree. We can add node Gender which has two values of male and female. The pure class is related to leaf node, thus Male gender has leaf node of Bus. For Female gender, we need to split further the attributes in the next iteration.



Third iteration

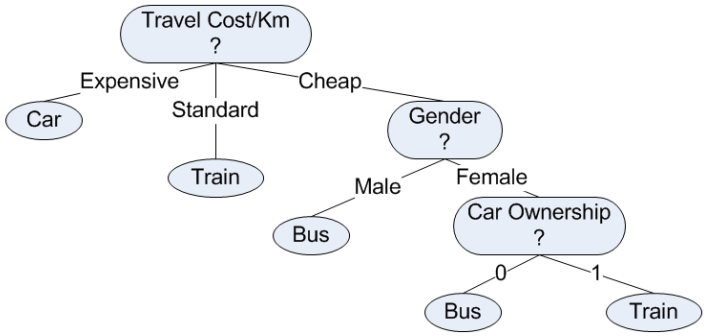
Data table of the third iteration comes only from part of the data table of the second iteration with male gender removed (thus only female part). Since attribute Gender has been used in the decision tree, we can remove the attribute and focus only on the remaining two attributes: Car ownership and Income level.

Attributes			Classes
Gender	Car ownership	Income Level	Transportation mode
~~Female~~	0	Low	Bus
~~Female~~	1	Medium	Train

Data third iteration

Attributes		Classes
Car ownership	Income Level	Transportation mode
0	Low	Bus
1	Medium	Train

If you observed the data table of the third iteration, it consists only two rows. Each row has distinct values. If we use attribute car ownership, we will get pure class for each of its value. Similarly, attribute income level will also give pure class for each value. Therefore, we can use either one of the two attributes. Suppose we select attribute car ownership, we can update our decision tree into the final version.



Now we have grown the full decision tree based on the data.