Missing Values

M.M. Pedram
pedram@khu.ac.ir
Kharazmi University
(Fall 2009)

1

Unknown Attribute Values

- In a data set, often some attribute values for some samples can be missing, because :
  - the value is not relevant to a particular sample, or
  - it was not recorded when the data was collected, or
  - an error was made by the person the entering data into a database.
- To solve the problem of missing values, there are two choices:
  1. Discard all samples in a database with missing data, or
  2. Define a new algorithm or modify an existing algorithm that will work with missing data.
- The first solution is simple but unacceptable when large amounts of missing values exist in a set of samples.

2

Unknown Attribute Values

- To address the second alternative, several questions must be answered:
  1. How does one compare two samples with different numbers of unknown values?
  2. Training samples with unknown values cannot be associated with a particular value of the test, and so they cannot be assigned to any subsets of cases. How should these samples be treated in the partitioning?
  3. In a testing phase of classification, how does one treat a missing value if the test is on the attribute with the missing value?

- Several classification algorithms that work with missing data are usually based on filling in a missing value with the most probable value, or on looking at the probability distribution of all values for the given attribute. None of these approaches is uniformly superior.

  ⇒ *Data Quality Studies* → *Data Quality Mining*

3

Unknown Attribute Values

C4.5 modification to handle missing values:

- In C4.5, *it is an accepted principle that samples with unknown values are distributed probabilistically according to the relative frequency of known values*.
  i. Calculate *Info*(I) as before, except that only samples with known values of attributes are taken into account.
  ii. Let *Info*(I, *x*) be calculated as before, except that only samples with known values of attribute *x* are taken into account.
  iii. Then the gain parameter can reasonably be corrected with a factor *F*:

$$F = \frac{\text{number of samples in the database with a known value for a given attribute}}{\text{total number of samples in a data set}}$$

The new gain criterion will have the form:

$$Gain\ (x) = F\ .\ [Info\ (I)\ -\ Info\ (I, x)]$$

4

Unknown Attribute Values

iv. *Split-info*(*x*) can be altered by regarding the samples with unknown values as an additional group in splitting., i.e., If attribute *x* has n outcomes, its *Split-info*(*x*) is computed as if *x* divided the data set into n+1 subsets.

v. Repeat steps i to iv for other attributes, and select the attribute with the largest *Gain_Ratio* for the current splitting point (node),

vi. After splitting the set D into subsets using the selected attribute, the record with the missing value will be represented (counted) in all subsets with the relative frequency of each subset.

vii. Repeat the steps ii to vi for each branches, till meet the stop condition.

5

Unknown Attribute Values

Example: there is one value missing for Attributel denoted by "?".

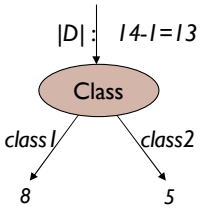
Attribute1	Attribute2	Attribute3	Class
A	70	True	CLASS1
A	90	True	CLASS2
A	85	False	CLASS2
A	95	False	CLASS2
A	70	False	CLASS1
?	90	True	CLASS1
B	78	False	CLASS1
B	65	True	CLASS1
B	75	False	CLASS1
C	80	True	CLASS2
C	70	True	CLASS2
C	80	False	CLASS1
C	80	False	CLASS1
C	96	False	CLASS1

6

Unknown Attribute Values

i.

$$Info(I) = -\frac{8}{13}\cdot\log_2\left(\frac{8}{13}\right) - \frac{5}{13}\cdot\log_2\left(\frac{5}{13}\right) = 0.961 \ bits$$



ii. Suppose x = Attribute1

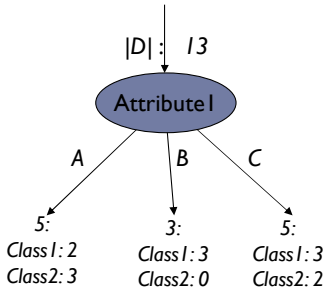
$$Info(I,x) = \frac{5}{13}H(A) + \frac{3}{13}H(B) + \frac{5}{13}H(C)$$

$$H(A) = -\frac{2}{5}\cdot\log_2\left(\frac{2}{5}\right) - \frac{3}{5}\cdot\log_2\left(\frac{3}{5}\right)$$

$$H(B) = 0$$

$$H(C) = -\frac{3}{5}\cdot\log_2\left(\frac{3}{5}\right) - \frac{2}{5}\cdot\log_2\left(\frac{2}{5}\right)$$

$$Info(I,x) = 0.747 \ bits$$



7

Unknown Attribute Values

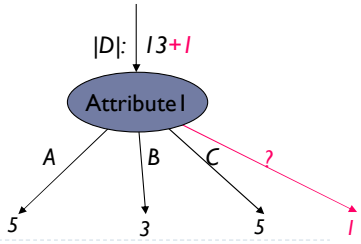
iii. $F = \frac{13}{14}$

$$Gain(x) = F.\left(Info(I) - Info(I,x)\right)$$

$$= \frac{13}{14}\cdot\left(0.961 - 0.747\right) = 0.199 \ bits$$

iv. $$Split\_Info(x = Attribute1) = -\frac{5}{14}\cdot\log_2\left(\frac{5}{14}\right) - \frac{3}{14}\cdot\log_2\left(\frac{3}{14}\right) - \frac{5}{14}\cdot\log_2\left(\frac{5}{14}\right) - \frac{1}{14}\cdot\log_2\left(\frac{1}{14}\right)$$

$$= 1.809 \ bits$$



8

Unknown Attribute Values

- If the missing value for *Attribute1* was B, then
  - *Gain*(*Attribute1*) would be 0.216. Thus, *Gain* is slightly lower when there is missing value, i.e. 0.199, as there is a factor *F* less than 1.
  - *Split_info* is still determined from the entire training set and is larger, since there is an extra category for unknown values.

	Attribute1		
	Without missing value	*With missing value*	
Gain (bits)	*0.216*	*0.199*	↓
Split_info (bits)	*1.577*	*1.809*	↑
Gain_ratio	*0.156*	*0.110*	↓

Unknown Attribute Values

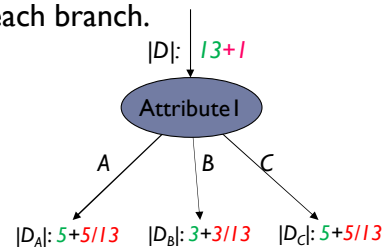
v. *Gain_ratio* is largest for Attribute1 (compute for other attributes!).

vi. Enumerate the record with the missing values in each branch with the relative frequency of each branch.

relative frequency(*A*)= 5/13

relative frequency(*B*)= 3/13

relative frequency(*C*)= 5/13



T_1: (*Attribute1 = A*)

Att.2	Att.3	Class	w
70	True	CLASS1	1
90	True	CLASS2	1
85	False	CLASS2	1
95	False	CLASS2	1
70	False	CLASS1	1
90	***True***	***CLASS1***	*5/13*

T_2: (*Attribute1 = B*)

Att.2	Att.3	Class	w
90	***True***	***CLASS1***	**3/13**
78	False	CLASS1	1
65	True	CLASS1	1
75	False	CLASS1	1

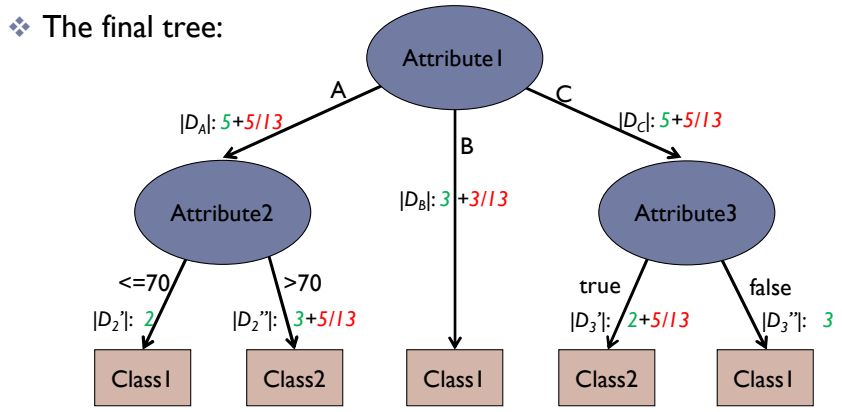
T_3: (*Attribute1 = C*)

Att.2	Att.3	Class	w
80	True	CLASS2	1
70	True	CLASS2	1
80	False	CLASS1	1
80	False	CLASS1	1
96	False	CLASS1	1
90	***True***	***CLASS1***	*5/13*

Unknown Attribute Values

vii. Repeat the steps ii to vi for each branches, till meet the stop condition.

- The final tree:



11

Unknown Attribute Values

```
If    Attribute1 = A
      Then
            If    Attribute2 <= 70
                  Then
                        Classification = CLASS1   (2.0 / 0);
                  Else
                        Classification = CLASS2   (3.4 / 0.4);
      Elseif Attribute1 = B
      Then
                        Classification = CLASS1   (3.2 / 0);
      Elseif Attribute1 = C
      Then
            If    Attribute3 = True
                  Then
                        Classification = CLASS2   (2.4 / 0.4)
                  Else
                        Classification = CLASS1   (3.0 / 0).
```

$|D_i|$ / E

$|D_A|$: 5+5/13

$|D_2'|$: 2

$|D_2''|$: 3+5/13,

$|D_B|$: 3 +3/13

$|D_3'|$: 2+5/13

$|D_3''|$: 3

12

Unknown Attribute Values

- (3.4 / 0.4) means that 3.4 (or 3 + 5/13) fractional training samples reached the leaf, of which 0.4 (or 5/13) did not belong to the class assigned to the leaf.

- It is possible to express the $|D_i|$ and E parameters in percentages. For example: $|D_i|/E$ = (3.4 / 0.4) $\rightarrow$
  - Accuracy = $(|D_i| - E)/|D_i|$ = 3/3.4 ×100% = 88% of cases at a given leaf would be classified as CLASS2.
  - Error = $E/|D_i|$ = 0.4/3.4 ×100% = 12% of cases at a given leaf would be classified as CLASS1.

Unknown Attribute Values Classification

Using Decision Tree to classify new samples (*test* & *recall*):

- A similar approach is taken in C4.5 when the decision tree is used to classify a sample previously not present in a database.
  - Starting with a root node in a decision tree, tests on attribute values will determine traversal through the tree, and at the end, the algorithm will finish in one of leaf nodes that uniquely defines the class of a testing example or with probabilities, if the training set had missing values.
    - If the value for a relevant testing attribute is unknown, the system explores all possible outcomes from the test the class with the highest probability is assigned as the predicted class.

Unknown Attribute Values