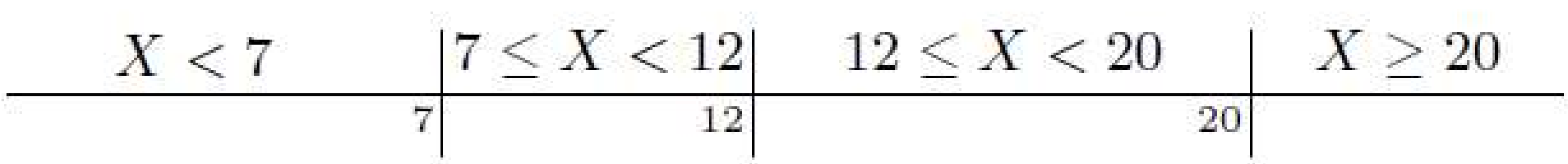
Computer Science 477

Continuous Attributes

Lecture 9

TIDIDT Constraints

- TIDIDT Requires categorical attributes
- Can take individual values 6.3, 7.2, 8.3, 9.2 as categorical
- (For reasons discussed), over splits the training data
  - Large number of subsets, each with few instances.
- More common to separate into non-overlapping subsets:



Discretization - Further Examples

- Convert age to infant, child, young adult, adult, middle-aged, old
- Convert height to very_short, short, medium, tall, very_tall
- Length, ranging from 0.3 to 6.6, inclusive:
- Divide into equal width intervals
  - $0.3 \leq length < 2.4$
  - $2.4 \leq length < 4.5$
  - $4.5 \leq length \leq 6.6$

Equal Width Intervals

- Number of ranges arbitrary
  - 3, not 4, not 12
- Perhaps, many, of the values are in a narrow range such as 2.35 to 2.45
  - A rule involving a test on *length* < 2.4 would include instances where *length* is say 2.39999 and exclude those where *length* is 2.40001.
  - Unlikely that there is any real difference between those values, especially if they were all measured imprecisely by different people at different times.
  - If there were no values between say 2.3 and 2.5, a test such as *length* < 2.4 reasonable.

Equal Frequency Intervals

- Divide *length* into three ranges,
  - Same number of instances in each of the three ranges:
  - $0.3 \leq length < 2.385$
  - $2.385 \leq length < 3.0$
  - $3.0 \leq length \leq 6.6$
- Same problem at cut points,
  - length of 2.99999 treated differently from one of 3.00001

Oversensitivity

- Whichever cut points are chosen there will always be a potential problems with values that fall just below a cut point being treated differently from those that fall just above for no principled reason.
- Ideally, find ‘gaps’ in the range of values.
- If in the *length* there are many values from 0.3 to 0.4 with the next smallest value being 2.2, a test such as *length* < 1.0 would avoid problems around the cut point
  - No instances (in the training set) with values close to 1.0.
  - The value 1.0 is y arbitrary and a different cut point, e.g. 1.5 could be chosen
  - Unfortunately the same gaps may not occur in unseen test data. If there were values such as 0.99, 1.05, 1.49 and 1.51 in the test data
  - Choice of 1.0 or 1.5 could be of critical importance.

Problems

- Equal width intervals and the equal frequency intervals take no account of the classifications when determining where to place the cut points

- One solution:
  - Local versus Global Discretization

Adding Local Discretization to TDIDT

- Convert continuous attributes to a categorical ones at each stage of the process
  - (e.g. at each node of the decision tree).

- Approach 1: convert at each step attributes according to previously noted methods, namely,
  - Equal width
  - Equal frequency

Approach 2

- At each node to convert each continuous attribute to a number of alternative categorical attributes.
- Example: if continuous attribute *A* has values -12.4, -2.4, 3.5, 6.7 and 8.5
  - (each possibly occurring several times)
- Test $A < 3.5$ splits the training data into two parts
- Equivalent to a kind of categorical attribute with two possible values
  - True and false.
- *Pseudo-attribute*

Pseudo-attributes

- Attribute *A* with *n* distinct values $v_1, v_2, \ldots, v_n$ there are $n - 1$ possible corresponding pseudo-attributes
  - $A < v_2, A < v_3, \ldots, A < v_n$
- If one of the pseudo-attributes, *Age* < 27.3, is selected at a node, we can consider the continuous attribute *Age* as having been discretized into two intervals with cut point 27.3.
- This is a local discretization which does not lead to the continuous attribute itself being discarded.
- Hence there may be a further test such as *Age* < 14.1 at a lower level in the 'yes' branch descending from the test *Age* < 27.3.

Algorithmically

- For each continuous attribute *A*
  - a) Sort the instances into ascending numerical order.
  - b) If there are *n* distinct values $v_1, v_2, \ldots, v_n$, calculate the values of information gain (or of GINI index or other measure) for each of the $n-1$ corresponding pseudo-attributes $A < v_2, A < v_3, \ldots, A < v_n$.
  - c) Find which of the $n-1$ attribute values gives the largest value of information gain (or optimizes some other measure).
  - If this is v_i return the pseudo-attribute $A < vi$, and the value of the corresponding measure.
- Calculate the value of information gain (or other measure) for any categorical attributes.
- Select the attribute or pseudo-attribute with the largest value of information gain (or which optimizes some other measure).

Pseudo-attributes - Information Gain

- Three stages
- First: Count the number of instances with each of the possible classifications in the part of the training set under consideration at the node.
- Values do not depend on which attribute is subsequently processed and so only have to be counted once at each node of the tree.

Pseudo-attributes – Stage Two

- Work through the continuous attributes one by one.
  - Call attribute Var
- Consider all possible pseudo-attributes $Var < X$ where X is one of the values of Var
  - In the part of the training set under consideration at the given node.
- Call the values of attribute Var candidate cut points.
- Call the largest value of measure maxmeasure and the value of X that gives that largest value the cut point for attribute Var.

Pseudo-Attributes – Stage Three

- Having found the value of *maxmeasure* (and the corresponding cut points)

- Find the largest and then compare it with the values of the measure obtained for any categorical attributes to determine which attribute or pseudo-attribute to split on at the node.

Example – Golf Data Set

- Count the number of instances with each of the possible classifications.
  - 9 *play* and 5 *don't play*, a total of 14.

Outlook	Temp (°F)	Humidity (%)	Windy	Class
sunny	75	70	true	play
sunny	80	90	true	don't play
sunny	85	85	false	don't play
sunny	72	95	false	don't play
sunny	69	70	false	play
overcast	72	90	true	play
overcast	83	78	false	play
overcast	64	65	true	play
overcast	81	75	false	play
rain	71	80	true	don't play
rain	65	70	true	don't play
rain	75	80	false	play
rain	68	80	false	play
rain	70	96	false	play

- Process each of the continuous attributes in turn (Stage 2).
  - Two: *temperature* and *humidity*.
  - Illustrate Stage 2 using attribute *temperature*

Example – Stage 2

- Sort attribute value in ascending order
  - Construct a two-column table
  - Attribute value and classification
  - *Sorted instances table*
- Twelve distinct values

Temperature	Class
64	play
65	don't play
68	play
69	play
70	play
71	don't play
72	play
72	don't play
75	play
75	play
80	don't play
81	play
83	play
85	don't play

Processing Sorted Instance Table

- n instances and rows in sorted instances numbered 1 to n
- Work through the table from bottom to top
  - Accumulate a count of the number of instances of each classification
- As each row is processed it attribute value is compared with the value for the row below
  - If larger, treat as candidate cut point
  - Value of measure is computed using the "frequency table method"
- Algorithm returns *maxmeasure* and *cutvalue*
  - *Maxmeasure is the information gain or gain raito (or whatever).*
  - *Cutvalue is the value is the attribute value that currently maximizes the maxmeasure*

Algorithm - Processing Sorted Instance Table

Algorithm for Processing a Sorted Instances Table

Set count of all classes to zero

Set maxmeasure to a value less than the smallest possible value of the measure used

```
for i=1 to n − 1 {
   increase count of class(i) by 1
   if value(i) < value(i + 1){
      (a) Construct a frequency table for pseudo-attribute
            Var < value(i + 1)
      (b) Calculate the value of measure
      (c) If measure > maxmeasure {
            maxmeasure=measure
            cutvalue=value(i + 1)
      }
   }
}
```

Processing the sorted instance table

- *golf* training set and continuous attribute temperature
- Temperature 64 and class *play*.
- Increase the count for class *play* to 1.
- Count for class *don't play* is zero.
- Temperature is less than that for the next instance
- So proceed to construct a frequency table for the pseudo-attribute *temperature* < 65

Temperature	Class
64	play
65	don't play
68	play

Class	$temperature < 65$	$temperature \geq 65$	Class total
play	1 *	8	**9**
don't play	0 *	5	**5**
Column sum	1	13	**14**

Processing second row of sorted instance table

- Temperature of 68, class don't play
- Create new frequency table
  - Update class count(s)
  - Column totals

Temperature	Class
64	play
65	don't play
68	play

Class	$temperature < 68$	$temperature \geq 68$	Class total
play	1 *	8	**9**
don't play	1 *	4	**5**
Column sum	2	12	**14**

ChiMerge Algorithm for Global Discretization

- Sort items according to continuous attribute values into ascending numerical order
- Construct a frequency table giving the number of occurrences of each distinct value of the attribute for each possible classification

Value of A	Observed frequency for class			Total
	$c1$	$c2$	$c3$	
1.3	1	0	4	5
1.4	0	1	0	1
1.8	1	1	1	3
2.4	6	0	2	8
6.5	3	2	4	9
8.7	6	0	1	7
12.1	7	2	3	12
29.4	0	0	1	1
56.2	2	4	0	6
87.1	0	1	3	4
89.0	1	1	2	4

ChiMerge Algorithm for Global Discretization

- Interpret each row not just as a single attribute
  - As an *interval*, i.e. a range of values
  - starting at the value, continuing up to but excluding the value given in the row below.
  - Row labelled 1.3 corresponds to the interval 1.3 ≤ *A* < 1.4. indicate the lowest number in the range of values included in that interval. The

Value of A	Observed frequency for class			Total
	$c1$	$c2$	$c3$	
1.3	1	0	4	5
1.4	0	1	0	1
1.8	1	1	1	3
2.4	6	0	2	8
6.5	3	2	4	9
8.7	6	0	1	7
12.1	7	2	3	12
29.4	0	0	1	1
56.2	2	4	0	6
87.1	0	1	3	4
89.0	1	1	2	4

ChiMerge Algorithm for Global Discretization

- Frequency table could be augmented by an additional column showing the interval corresponding to each classification

Value of A	Interval	Observed frequency for class			Total
		$c1$	$c2$	$c3$	
1.3	$1.3 \leq A < 1.4$	1	0	4	5
1.4	$1.4 \leq A < 1.8$	0	1	0	1
1.8	$1.8 \leq A < 2.4$	1	1	1	3
2.4	$2.4 \leq A < 6.5$	6	0	2	8
6.5	$6.5 \leq A < 8.7$	3	2	4	9
8.7	$8.7 \leq A < 12.1$	6	0	1	7
12.1	$12.1 \leq A < 29.4$	7	2	3	12
29.4	$29.4 \leq A < 56.2$	0	0	1	1
56.2	$56.2 \leq A < 87.1$	2	4	0	6
87.1	$87.1 \leq A < 89.0$	0	1	3	4
89.0	$89.0 \leq A$	1	1	2	4

ChiMerge Algorithm for Global Discretization

- ChiMerge systematically applies statistical tests to combine pairs
- Does not merge intervals that are statistically different
- Implicitly, if a pair is merged if it doesn’t modify outcome, classification
- For each pair, tests the hypothesis

<u>Hypothesis</u>
The class is independent of which of the two adjacent intervals an instance belongs to.

- If the hypothesis is confirmed, intervals are merged

Statistical test: χ^2

$$6.5 \leq A < 8.7$$
$$8.7 \leq A < 12.1$$

- χ^2 test for independence
- For each pair of adjacent rows, construct a contingency table:

Value of A	Observed frequency for class			Total observed
	$c1$	$c2$	$c3$	
8.7	6	0	1	7
12.1	7	2	3	12
Total	13	2	4	19

- The 'row sum' (right-hand column) and the 'column sum' (bottom row) - 'marginal totals'.
- Correspond (respectively) to
  - Number of instances for each value of *A* (i.e. with their value of attribute in the corresponding interval)
  - Number of instances in each class for both intervals combined.

Use of the χ^2 statistic

- χ^2 value is then compared with a *threshold value T*
  - Depends on the number of classes and
  - The level of statistical significance required.
- (Here) use a significance level of 90%
  - Gives a threshold value of 4.61.
- If we assume that the classification is independent of which of the two adjacent intervals an instance belongs to, there is a 90% probability that χ^2 will be less than 4.61.
- If χ^2 is less than 4.61 the hypothesis of independence is supported at the *90% significance level*
  - The two intervals are merged.

Calculating the Expected Values and χ^2

- For a given pair of adjacent rows (intervals) the value of χ^2 is calculated using
  - The ‘observed’ and ‘expected’ frequency values
  - For each combination of class and row.
- There are three classes so there are six such combinations.
- Observed frequency value, denoted by *O*, is the frequency that actually occurred.
- Expected value *E* is the frequency value that would be expected to occur by chance
  - Given the assumption of independence

Calculating the Expected Values and χ^2

- Row is *i* and the class is *j*, then let the total number of instances in row *i* be $rowsum_i$ and let the total number of occurrences of class *j* be $colsum_j$.
- The grand total number of instances for the two rows combined be *sum*.
- Assuming the hypothesis that the class is independent of which of the two rows an instance belongs, calculate the expected number of instances in row *i* for class *j* thus: follows.
- There are a total of $colsum_j$ occurrences of class *j* in the two intervals combined,
- So class *j* occurs a proportion of $\frac{colsum_j}{sum}$ the time.

Calculating the Expected Values and χ^2

- There are a total of $colsum_j$ occurrences of class j in the two intervals combined,
- So class j occurs a proportion of $\frac{colsum_j}{sum}$ the time.
- There are $rowsum_i$ instances in row i ; expect $rowsum_i \frac{colsum_j}{sum}$ occurrences of class j in row i.
- To calculate this value for any combination of row and class,
  - Take the product of the corresponding row sum and column sum
  - Divide by the grand total of the observed values for the two rows.

Expected Values and χ^2 - Example

- To calculate expected value for any combination of row and class,
  - Take the product of the corresponding row sum and column sum
  - Divided by the grand total of the observed values for the two rows.
- For the adjacent intervals labelled 8.7 and 12.1 the six values of *O* and *E* are:
  - Expected value for C1 at value 8.7 is $\frac{7 \times 13}{19} = 4.789 \sim 4.79$

Value of A	Observed frequency for class			Total
	$c1$	$c2$	$c3$	
1.3	1	0	4	5
1.4	0	1	0	1
1.8	1	1	1	3
2.4	6	0	2	8
6.5	3	2	4	9
8.7	6	0	1	7
12.1	7	2	3	12
29.4	0	0	1	1
56.2	2	4	0	6
87.1	0	1	3	4
89.0	1	1	2	4

Value of A	Frequency of class						Total Observed
	C1		C2		C3		
	O	E	O	E	O	E	
8.7	6	4.79	0	0.74	1	1.47	7
12.1	7	8.21	2	1.26	3	2.53	12
Total	13		2		4		19

Finally calculating χ^2

- Using observed and expected values, calculate $\frac{(0-E)^2}{E}$ for each of the six combinations
- Value of χ^2 is the sum of the six values for $\frac{(0-E)^2}{E}$
- If the independence hypothesis is correct O and E values would be the same and χ^2 is zero
  - Small value for χ^2 supports hypothesis
  - Larger value militates against it
- When χ^2 exceeds threshold, hypothesis is rejected
- Important adjustment, when $E < 0.5$ replace it with 0.5

Final Calculation

- χ^2 at each row is the value for the pair of adjacent row
  - That row
  - The row below
- Original table has 11 rows, so $10 \times \chi^2$ calculations, values
- Each value represents is the value for that row and the one below it

Value of A	Frequency for class $c1$	$c2$	$c3$	Total	Value of χ^2
1.3	1	0	4	5	3.11
1.4	0	1	0	1	1.08
1.8	1	1	1	3	2.44
2.4	6	0	2	8	3.62
6.5	3	2	4	9	4.62
8.7	6	0	1	7	1.89
12.1	7	2	3	12	1.73
29.4	0	0	1	1	3.20
56.2	2	4	0	6	6.67
87.1	0	1	3	4	1.20
89.0	1	1	2	4	
Total	27	12	21	60	

Final Step

- Select the smallest χ^2 value
- Compare it to the threshold
- If it falls below the threshold, merge it with the row immediately below it
  - The independence of which the χ^2 represents
- Smallest value is 1.08 for row

  1.4
- New resulting interval is

  $1.4 \leq x < 2.4$

Value of A	Frequency for class			Total	Value of χ^2
	$c1$	$c2$	$c3$		
1.3	1	0	4	5	3.11
1.4	0	1	0	1	1.08
1.8	1	1	1	3	2.44
2.4	6	0	2	8	3.62
6.5	3	2	4	9	4.62
8.7	6	0	1	7	1.89
12.1	7	2	3	12	1.73
29.4	0	0	1	1	3.20
56.2	2	4	0	6	6.67
87.1	0	1	3	4	1.20
89.0	1	1	2	4	
Total	27	12	21	60	

New Table

- Revised frequency table

Value of A	Frequency for class			Total
	$c1$	$c2$	$c3$	
1.3	1	0	4	5
1.4	1	2	1	4
2.4	6	0	2	8
6.5	3	2	4	9
8.7	6	0	1	7
12.1	7	2	3	12
29.4	0	0	1	1
56.2	2	4	0	6
87.1	0	1	3	4
89.0	1	1	2	4

Merged

Final Step

- χ^2 values calculated for the new frequency table
- Only need to do this for rows adjacent to the recently merged one

Value of A	Frequency for class			Total	Value of χ^2
	$c1$	$c2$	$c3$		
1.3	1	0	4	5	**3.74**
1.4	1	2	1	4	**5.14**
2.4	6	0	2	8	3.62
6.5	3	2	4	9	4.62
8.7	6	0	1	7	1.89
12.1	7	2	3	12	1.73
29.4	0	0	1	1	3.20
56.2	2	4	0	6	6.67
87.1	0	1	3	4	1.20
89.0	1	1	2	4	
Total	27	12	21	60	

- Smallest χ^2 is 1.20
  - Below the threshold
- Intervals 87.1 and 89.0 merged
- Continue until one reaches a fixed point:
  - Smallest χ^2 is above the threshold

Final Table

- All possible merging complete

Value of A	Frequency for class			Total	Value of χ^2
	$c1$	$c2$	$c3$		
1.3	24	6	16	46	10.40
56.2	2	4	0	6	5.83
87.1	1	2	5	8	
Total	27	12	21	60	

- $1.3 \leq x < 56.2$
- $56.2 \leq x < 87.1$
- $x \geq 87.1$

minIntervals and maxIntervals

- Two extrema:
  - Large number of intervals
  - Just one interval
- Large numbers of intervals does little to solve the problem of discretization
- Just one interval cannot contribute to a decision making process
  - Attribute value is independent of classification.
- Two solutions
  - Modify significance level hypothesis of independence must pass, triggering interval merge.
  - Set a minimum and a maximum number of intervals
- minInterval and maxInteval difficult to justify by statistical theory