Data Mining and Knowledge Discovery
Practice notes – 29.11.2011

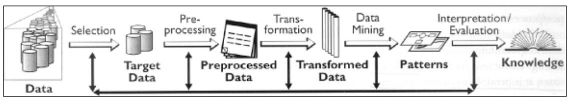
Data Mining and Knowledge Discovery

Petra Kralj Novak
Petra.Kralj.Novak@ijs.si
2011/11/29

Practice plan

- 2011/11/08: Predictive data mining 1
  - Decision trees
  - Evaluating classifiers 1: separate test set, confusion matrix, classification accuracy
  - A taste of Weka
- 2011/11/22: Predictive data mining 2
  - Evaluating classifiers 2: Cross validation
  - Naïve Bayes classifier
  - Numeric prediction
- 2011/11/29: Descriptive data mining
  - Association and classification rules
  - Descriptive data mining in Weka
  - Discussion about seminars and exam
- 2011/12/20: Written exam, Seminar proposal presentations
- 2012/1/24 : Data mining seminar presentations

Keywords



- Data
  - Attribute, example, target variable, class, train set, test set, attribute-value data, market basket data
- Data mining
  - decision tree induction, entropy, information gain, overfitting, Occam's razor, model pruning, naïve Bayes classifier, KNN, association rules, support, confidence, predictive vs. descriptive DM, numeric prediction, regression tree, model tree, heuristics vs. exhaustive search
- Evaluation
  - Accuracy, confusion matrix, cross validation, ROC space, error, leave-one-out

Categorical or numeric?

- Variable with five possible values:
  1. non sufficient
  2. sufficient
  3. good
  4. very good
  5. excellent

Classification or a numeric prediction problem?

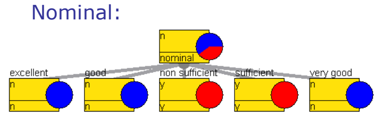
- Target variable with five possible values:
  1. non sufficient
  2. sufficient
  3. good
  4. very good
  5. excellent
- Classification: the **misclassification cost** is the same if "non sufficient" is classified as "sufficient" or if it is classified as "very good"
- Numeric prediction: The error of predicting "2" when it should be "1" is 1, while the error of predicting "5" instead of "1" is 4.
- If we have a variable with ordered values,

it is better to treat it as numeric.

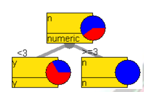
Categorical or numeric attribute?

- A variable with five possible values:
  1. non sufficient
  2. sufficient
  3. good
  4. very good
  5. excellent

Nominal:



Numeric:



- If we have a variable with **ordered** values,

it is better to treat it as numeric.

Information gain of a numeric attribute

Age	Lenses
67	YES
52	YES
63	NO
26	YES
65	NO
23	YES
65	NO
25	YES
26	YES
57	NO
49	NO
23	YES
39	NO
55	NO
53	NO
38	NO
67	YES
54	NO
29	YES
46	NO
44	YES
32	NO
39	NO
45	YES

Information gain of a numeric attribute

Age	Lenses
67	YES
52	YES
63	NO
26	YES
65	NO
23	YES
65	NO
25	YES
26	YES
57	NO
49	NO
23	YES
39	NO
55	NO
53	NO
38	NO
67	YES
54	NO
29	YES
46	NO
44	YES
32	NO
39	NO
45	YES

Sort by Age →

Age	Lenses
23	YES
23	YES
25	YES
26	YES
26	YES
29	YES
32	NO
38	NO
39	NO
39	NO
44	YES
45	YES
46	NO
49	NO
52	YES
53	NO
54	NO
55	NO
57	NO
63	NO
65	NO
65	NO
67	YES
67	YES

Information gain of a numeric attribute

Age	Lenses
67	YES
52	YES
63	NO
26	YES
65	NO
23	YES
65	NO
25	YES
26	YES
57	NO
49	NO
23	YES
39	NO
55	NO
53	NO
38	NO
67	YES
54	NO
29	YES
46	NO
44	YES
32	NO
39	NO
45	YES

Sort by Age →

Age	Lenses
23	YES
23	YES
25	YES
26	YES
26	YES
29	YES
32	NO
38	NO
39	NO
39	NO
44	YES
45	YES
46	NO
49	NO
52	YES
53	NO
54	NO
55	NO
57	NO
63	NO
65	NO
65	NO
67	YES
67	YES

Define possible splitting points

Age	Lenses
23	YES
23	YES
25	YES
26	YES
26	YES
29	YES
32	NO
38	NO
39	NO
39	NO
44	YES
45	YES
46	NO
49	NO
52	YES
53	NO
54	NO
55	NO
57	NO
63	NO
65	NO
65	NO
67	YES
67	YES

Information gain of a numeric attribute

Age	Lenses	
23	YES	
23	YES	
25	YES	
26	YES	
26	YES	
29	YES	→ **30.5**
32	NO	
38	NO	
39	NO	
39	NO	→ **41.5**
44	YES	
45	YES	→ **45.5**
46	NO	
49	NO	→ **50.5**
52	YES	→ **52.5**
53	NO	
54	NO	
55	NO	
57	NO	
63	NO	
65	NO	
65	NO	→ **66**
67	YES	
67	YES	

Information gain of a numeric attribute

Age	Lenses	
23	YES	
23	YES	
25	YES	
26	YES	
26	YES	
29	YES	→ **30.5**
32	NO	
38	NO	
39	NO	
39	NO	→ **41.5**
44	YES	
45	YES	→ **45.5**
46	NO	
49	NO	→ **50.5**
52	YES	→ **52.5**
53	NO	
54	NO	
55	NO	
57	NO	
63	NO	
65	NO	
65	NO	→ **66**
67	YES	
67	YES	

Age

<30.5, 6/24: E(6/6 , 0/6) = 0

>=30.5, 18/24: E(5/18 , 13/18) = 0.85

Information gain of a numeric attribute

Age	Lenses	
23	YES	
23	YES	
25	YES	
26	YES	
26	YES	
29	YES	→ **30.5**
32	NO	
38	NO	
39	NO	
39	NO	→ **41.5**
44	YES	
45	YES	→ **45.5**
46	NO	
49	NO	→ **50.5**
52	YES	→ **52.5**
53	NO	
54	NO	
55	NO	
57	NO	
63	NO	
65	NO	
65	NO	→ **66**
67	YES	
67	YES	

E(S) = E(11/24 , 13/24) = 0.99

Age

<30.5, 6/24: E(6/6 , 0/6) = 0

>=30.5, 18/24: E(5/18 , 13/18) = 0.85

InfoGain (S, $Age_{30.5}$)=

$= E(S) - \sum p_v E(p_v)$

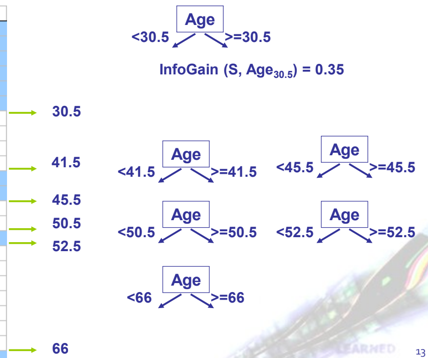
= 0.99 – (6/24*0 + 18/24*0.85)

= 0.35

Data Mining and Knowledge Discovery
Practice notes – 29.11.2011

Information gain of a numeric attribute

Age	Lenses
23	YES
23	YES
25	YES
26	YES
26	YES
29	YES
32	NO
38	NO
39	NO
39	NO
44	YES
45	YES
46	NO
49	NO
52	YES
53	NO
54	NO
55	NO
57	NO
63	NO
65	NO
65	NO
67	YES
67	YES



Classification rules

Covering algorithm (e.g. Ripper by Cohen, 1995):

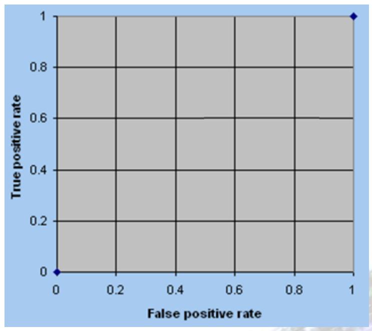
- We have an empty rule base
- Add "the best" rule to the rule base
- Remove the positive examples that are covered by "the best" rule from the training dataset
- Until there are no more positive examples in the training dataset

Find the best rule:

- Start with an empty rule condition
- add one condition at a time to the current rule and evaluate the rule (information gain, Laplace estimate)

ROC space and AUC

ROC ... Reciever Operator Charachteristics



Simple mushroom dataset

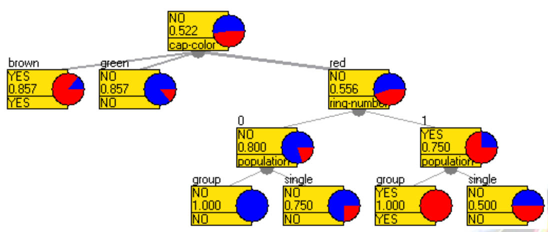
Train set

cap-color	ring-number	population	EDIBLE
red	1	single	YES
green	1	group	NO
brown	1	single	YES
brown	1	single	YES
brown	1	single	YES
brown	1	single	YES
red	1	single	NO
red	0	group	NO
green	0	group	NO
green	0	single	NO
green	0	single	NO
red	1	group	YES
red	1	group	YES
brown	1	group	YES
brown	0	single	YES
brown	0	single	NO
green	0	group	NO
green	0	group	NO
red	0	single	NO
red	0	single	YES
red	0	single	NO
green	0	group	YES
red	0	single	NO

Test set

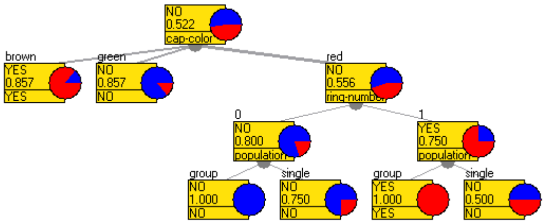
cap-color	ring-number	population	EDIBLE
brown	1	single	NO
green	0	group	NO
red	1	single	YES
red	0	group	NO
red	1	group	YES

Decision tree induced on the train set



Data Mining and Knowledge Discovery
Practice notes – 29.11.2011

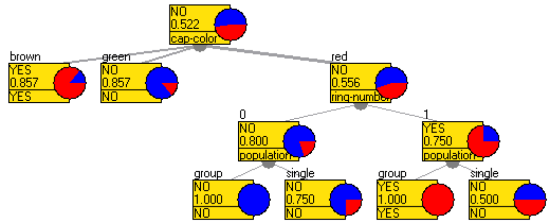
Confusion matrix



cap-color	ring-number	population	EDIBLE	DT1
brown	1	single	NO	
green	0	group	NO	
red	1	single	YES	
red	0	group	NO	
red	1	group	YES	

	Predicted YES	Predicted NO
Actual YES		
Actual NO		

Confusion matrix



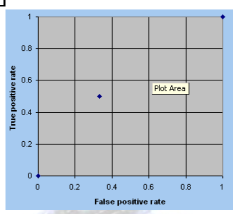
cap-color	ring-number	population	EDIBLE	DT1
brown	1	single	NO	YES
green	0	group	NO	NO
red	1	single	YES	NO
red	0	group	NO	NO
red	1	group	YES	YES

	Predicted YES	Predicted NO
Actual YES	1	1
Actual NO	1	2

ROC space

	Predicted YES	Predicted NO
Actual YES	1	1
Actual NO	1	2

- True positive rate =
  = # true positives / # all positives =
  = TPr = 1/2
- False positive rate =
  = # false positives / # all negatives =
  = FPr = 1/3



ROC space 2

- Classifier "always YES"

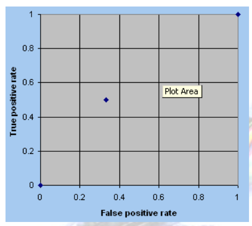
	Predicted YES	Predicted NO
Actual YES	2	0
Actual NO	3	0

- TPr = 1
- FPr = 1

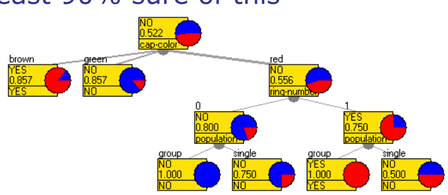
- Classifier "always NO"

	Predicted YES	Predicted NO
Actual YES	0	2
Actual NO	0	3

- TPr = 0
- FPr = 0



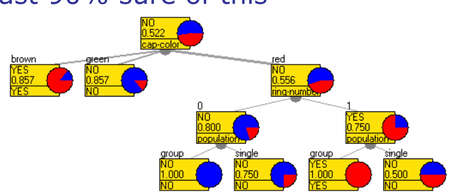
Confusion matrix 2: A mushroom is edible if the model is at least 90% sure of this



cap-color	ring-number	population	EDIBLE	DT2
brown	1	single	NO	
green	0	group	NO	
red	1	single	YES	
red	0	group	NO	
red	1	group	YES	

	Predicted YES	Predicted NO
Actual YES		
Actual NO		

Confusion matrix 2: A mushroom is edible if the model is at least 90% sure of this



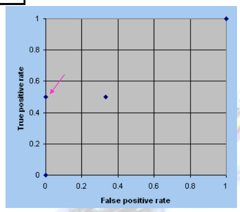
cap-color	ring-number	population	EDIBLE	DT2
brown	1	single	NO	NO
green	0	group	NO	NO
red	1	single	YES	NO
red	0	group	NO	NO
red	1	group	YES	YES

	Predicted YES	Predicted NO
Actual YES	1	1
Actual NO	0	3

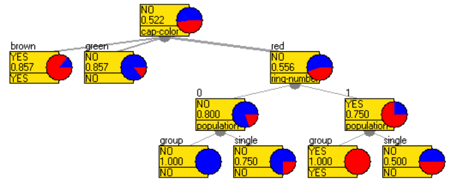
ROC space

	Predicted YES	Predicted NO
Actual YES	1	1
Actual NO	0	3

- True positive rate TPr = 1/2
- False positive rate FPr = 0



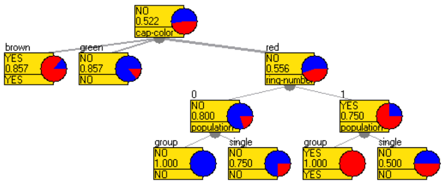
Confusion matrix 3: A mushroom is edible if the model is at least 20% sure of this



cap-color	ring-number	population	EDIBLE	DT3
brown	1	single	NO	
green	0	group	NO	
red	1	single	YES	
red	0	group	NO	
red	1	group	YES	

	Predicted YES	Predicted NO
Actual YES		
Actual NO		

Confusion matrix 3: A mushroom is edible if the model is at least 20% sure of this



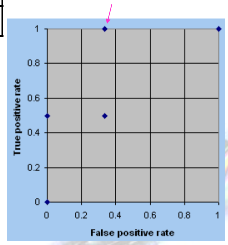
cap-color	ring-number	population	EDIBLE	DT3 (20%)
brown	1	single	NO	YES
green	0	group	NO	NO
red	1	single	YES	YES
red	0	group	NO	NO
red	1	group	YES	YES

	Predicted YES	Predicted NO
Actual YES	2	0
Actual NO	1	2

ROC space

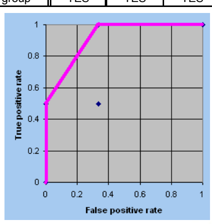
	Predicted YES	Predicted NO
Actual YES	2	0
Actual NO	1	2

- True positive rate TPr = 1
- False positive rate FPr = 1/3



ROC convex hull

cap-color	ring-number	population	EDIBLE	DT1 (50%)	DT2 (90%)	DT3 (20%)	YES	NO
brown	1	single	NO	YES	NO	YES	YES	NO
green	0	group	NO	NO	NO	NO	YES	NO
red	1	single	YES	NO	NO	YES	YES	NO
red	0	group	NO	NO	NO	NO	YES	NO
red	1	group	YES	YES	YES	YES	YES	NO

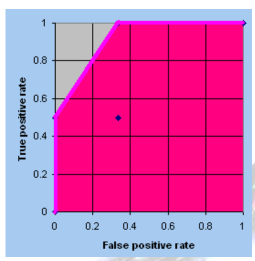


AUC – Area Under Curve

AUC =

= (0.5+1)/2*1/3+2/3

= 0.917



Association Rules

Association rules

- Rules **X → Y**, X, Y conjunction of items
- Task: Find **all** association rules that satisfy minimum support and minimum confidence constraints

- **Support**:
  Sup(X → Y) = #XY/#D ≅ p(XY)
- **Confidence**:
  Conf(X → Y) = #XY/#X ≅ p(XY)/p(X) = p(Y|X)

Association rules - algorithm

1. generate frequent itemsets with a minimum support constraint
2. generate rules from frequent itemsets with a minimum confidence constraint

* Data are in a transaction database

Association rules – transaction database

Items: **A**=apple, **B**=banana,
C=coca-cola, **D**=doughnut

- Client 1 bought: A, B, C, D
- Client 2 bought: B, C
- Client 3 bought: B, D
- Client 4 bought: A, C
- Client 5 bought: A, B, D
- Client 6 bought: A, B, C

Frequent itemsets

- Generate frequent itemsets with support at least 2/6

A	B	C	D
1	1	1	1
	1	1	
	1		1
1		1	
1	1		1
1	1	1	

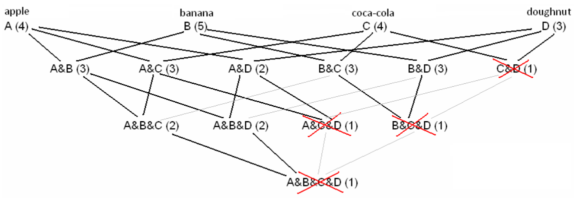
Frequent itemsets algorithm

Items in an itemset should be sorted alphabetically.

- Generate all 1-itemsets with the given minimum support.
- Use 1-itemsets to generate 2-itemsets with the given minimum support.
- From 2-itemsets generate 3-itemsets with the given minimum support as unions of 2-itemsets with the same item at the beginning.
- …
- From n-itemsets generate (n+1)-itemsets as unions of n-itemsets with the same (n-1) items at the beginning.

Data Mining and Knowledge Discovery
Practice notes – 29.11.2011

Frequent itemsets lattice



Frequent itemsets:
- A&B, A&C, A&D, B&C, B&D
- A&B&C, A&B&D

Rules from itemsets

- A&B is a frequent itemset with support 3/6
- Two possible rules
  - A→B confidence = #(A&B)/#A = 3/4
  - B→A confidence = #(A&B)/#B = 3/5
- All the counts are in the itemset lattice!

Quality of association rules

Support(X) = #X / #D .... P(X)
Support(X→Y) = Support (XY) = #XY / #D .... P(XY)
Confidence(X→Y) = #XY / #X .... P(Y|X)

Lift(X→Y) = Support(X→Y) / (Support (X)*Support(Y))

Leverage(X→Y) = Support(X→Y) – Support(X)*Support(Y)

Conviction(X → Y) = 1-Support(Y)/(1-Confidence(X→Y))

Quality of association rules

Support(X) = #X / #D .... P(X)
Support(X→Y) = Support (XY) = #XY / #D .... P(XY)
Confidence(X→Y) = #XY / #X .... P(Y|X)

Lift(X→Y) = Support(X→Y) / (Support (X)*Support(Y))
How many more times the items in X and Y occur together then it would be expected if the itemsets were statistically independent.

Leverage(X→Y) = Support(X→Y) – Support(X)*Support(Y)
Similar to lift, difference instead of ratio.

Conviction(X → Y) = 1-Support(Y)/(1-Confidence(X→Y))
Degree of implication of a rule.
Sensitive to rule direction.

Discussion

- Transformation of an attribute-value dataset to a transaction dataset.
- What would be the association rules for a dataset with two items A and B, each of them with support 80% and appearing in the same transactions as rarely as possible?
  - minSupport = 50%, min conf = 70%
  - minSupport = 20%, min conf = 70%
- What if we had 4 items: A, ¬A, B, ¬ B
- Compare decision trees and association rules regarding handling an attribute like “PersonID”. What about attributes that have many values (eg. Month of year)