Tertius: first-order rule discovery

- Common distinction between predictive (supervised) and descriptive (non-supervised) induction
- Descriptive approaches include clustering, association rule learning, learning of attribute dependencies, subgroup discovery, multiple predicate learning, …
- ***Rule discovery*** is about learning descriptive rather than predictive rules

Association rules

- Examples of association rules $X \rightarrow Y$:
  - Students taking courses *X* tend to take course *Y*
  - Clients purchasing products *X* tend to purchase product *Y*
  - Papers referring to papers *X* tend to refer to paper *Y*
- Heuristics:
  - ***Frequency***: fraction of all examples that satisfy *X* and *Y*
  - ***Confidence***: fraction of examples satisfying *X* that also satisfy *Y*
- Algorithm:
  - find boundary of itemsets exceeding frequency threshold
  - partition frequent sets to find rules with sufficient confidence

Association rules: example

A	*B*	*C*	*D*
+	+	–	+
–	–	+	–
+	+	+	–
–	+	–	+
–	+	+	+

- Itemsets with frequency ≥.4 are *AB*, *BC*, *BD* and their subsets
- Association rules with confidence ≥.75 are $\varnothing \rightarrow B$, $A \rightarrow B$, $D \rightarrow B$, and $B \rightarrow D$

Clausal discovery

- Different from classification rule learning, because clauses can have different heads
- Different from association rule learning, because heads are disjunctive
- Requirements for confirmation heuristic:
  - needs to take the whole contingency table into account
  - counter-instances are those that satisfy the head and falsify the body, all other instances confirm the rule
  - shouldn't be symmetric in head and body, otherwise it learns equivalences rather than implications

The Tertius approach

- Suppose the formula to be evaluated is an implication **H←B**:
  - determine the truthvalues of **H** and **B** for each example, and organise the observed frequencies in a contingency table

	B	¬B	
H	n_{HB}	$n_{H\neg B}$	n_H
¬H	$n_{\neg HB}$	$n_{\neg H\neg B}$	$n_{\neg H}$
	n_B	$n_{\neg B}$	N

The Tertius approach

- Suppose the formula to be evaluated is an implication **H←B**:
  - obtain expected frequencies from the marginals under some null hypothesis of independence

	B	$\neg B$	
H	n_{HB} (μ_{HB})	$n_{H\neg B}$ $(\mu_{H\neg B})$	n_H
$\neg H$	$n_{\neg HB}$ $(\mu_{\neg HB})$	$n_{\neg H\neg B}$ $(\mu_{\neg H\neg B})$	$n_{\neg H}$
	n_B	$n_{\neg B}$	N

The Tertius approach

- Suppose the formula to be evaluated is an implication **H←B**:
  - define confirmation in terms of the difference between the expected frequency of ***counter-instances*** $\mu_{\neg HB}$ and the observed frequency $n_{\neg HB}$

	B		¬B		
H	n_{HB}	(μ_{HB})	$n_{H\neg B}$	$(\mu_{H\neg B})$	n_H
¬H	$n_{\neg HB}$	$(\mu_{\neg HB})$	$n_{\neg H\neg B}$	$(\mu_{\neg H\neg B})$	$n_{\neg H}$
	n_B		$n_{\neg B}$		N

Which null hypotheses?

- ***Simple expected frequencies*** assume **H** and **B** are statistically independent:
  - $\mu_{\neg HB} = n_{\neg H} * n_B / N$
- This can be adapted to take logical background knowledge into account
  - multiway contingency tables (computationally expensive!)

Novelty and satisfaction

- ***Novelty*** is defined as the relative decrease in counter-instances from expected to observed:
  - $\Delta_{\neg HB} = (\mu_{\neg HB} - n_{\neg HB}) / N$ $\quad -.25 \leq \Delta_{\neg HB} \leq .25$
    - $= [p(H \mid B) - p(H)]\, p(B)$
  - Also called ***weighted relative accuracy***
    - note: symmetric in H and B

- ***Satisfaction*** is defined as the fraction of expected but non-observed counter-instances:
  - $\sigma_{\neg HB} = (\mu_{\neg HB} - n_{\neg HB}) / \mu_{\neg HB}$ $\quad 0 \leq \sigma_{\neg HB} \leq 1$ if $\Delta_{\neg HB} \geq 0$
    - $= [p(H \mid B) - p(H)] / [1 - p(H)]$

Confirmation

- ***Confirmation*** trades off novelty and satisfaction
  - $\Delta_{\neg HB} * \sigma_{\neg HB} = (\mu_{\neg HB} - n_{\neg HB})^2 / N \mu_{\neg HB}$
    is the contribution of ¬HB to $\Phi^2 = \chi^2 / N$

- Theorem:

$$\Phi^2 \geq \left(\frac{\mu_{\neg HB} - n_{\neg HB}}{\sqrt{N\mu_{\neg HB} - \mu_{\neg HB}}} \right)^2 = \left[conf(H \leftarrow B) \right]^2$$

  - *conf*(H←B) is lowest $\sqrt{\Phi^2}$ given $n_{\neg HB}$, $\mu_{\neg HB}$ and N

Virtual contingency table

	B	¬B	
H	$\sqrt{N}\mu_{\neg HB}-n_{\neg HB}$	$N-2\sqrt{N}\mu_{\neg HB}+n_{\neg HB}$	$N-\sqrt{N}\mu_{\neg HB}$
¬H	$n_{\neg HB}$	$\sqrt{N}\mu_{\neg HB}-n_{\neg HB}$	$\sqrt{N}\mu_{\neg HB}$
	$\sqrt{N}\mu_{\neg HB}$	$N-\sqrt{N}\mu_{\neg HB}$	N

Example: contact lenses

Age	*Spectacles*	*Astigmatic*	*Tear production*	***Lenses***
young	myope	no	reduced	**none**
young	myope	no	normal	**soft**
young	myope	yes	reduced	**none**
young	myope	yes	normal	**hard**
young	hypermetrope	no	reduced	**none**
young	hypermetrope	no	normal	**soft**
young	hypermetrope	yes	reduced	**none**
young	hypermetrope	yes	normal	**hard**
pre-presbyopic	myope	no	reduced	**none**
pre-presbyopic	myope	no	normal	**soft**
pre-presbyopic	myope	yes	reduced	**none**
pre-presbyopic	myope	yes	normal	**hard**
pre-presbyopic	hypermetrope	no	reduced	**none**
pre-presbyopic	hypermetrope	no	normal	**soft**
pre-presbyopic	hypermetrope	yes	reduced	**none**
pre-presbyopic	hypermetrope	yes	normal	**none**
presbyopic	myope	no	reduced	**none**
presbyopic	myope	no	normal	**none**
presbyopic	myope	yes	reduced	**none**
presbyopic	myope	yes	normal	**hard**
presbyopic	hypermetrope	no	reduced	**none**
presbyopic	hypermetrope	no	normal	**soft**
presbyopic	hypermetrope	yes	reduced	**none**
presbyopic	hypermetrope	yes	normal	**none**

Contact lense decision list

```
IF Tear-production = reduced THEN Lenses = none (12)
ELSE /* Tear-production = normal */
     IF Astigmatic = no THEN Lenses = soft (6/1)
     ELSE /* Astigmatic = yes */
          IF Spectacles = myope THEN Lenses = hard (3)
          ELSE /* Spectacles = hypermetrope */
               Lenses = none (3/1)
```

```
=== Confusion Matrix ===

  a  b  c   <-- classified as
  5  0  0 |  a = soft
  0  3  1 |  b = hard
  1  0 14 |  c = none
```

Contact lense association rules

```
 1. (1.00) Tear-prod=reduced 12 ==> Lenses=none 12
 2. (1.00) Astigm=yes Tear-prod=reduced 6 ==> Lenses=none 6
 3. (1.00) Astigm=no Tear-prod=reduced 6 ==> Lenses=none 6
 4. (1.00) Spectacles=hmetr Tear-prod=reduced 6 ==> Lenses=none 6
 5. (1.00) Spectacles=myope Tear-prod=reduced 6 ==> Lenses=none 6
 6. (1.00) Lenses=soft 5 ==> Astigm=no Tear-prod=normal 5
 7. (1.00) Astigm=no Lenses=soft 5 ==> Tear-prod=normal 5
 8. (1.00) Tear-prod=normal Lenses=soft 5 ==> Astigm=no 5
 9. (1.00) Lenses=soft 5 ==> Tear-prod=normal 5
10. (1.00) Lenses=soft 5 ==> Astigm=no 5
11. (0.86) Astigm=no Lenses=none 7 ==> Tear-prod=reduced 6
12. (0.86) Spectacles=myope Lenses=none 7 ==> Tear-prod=reduced 6
13. (0.83) Astigm=no Tear-prod=normal 6 ==> Lenses=soft 5
14. (0.83) Spectacles=hmetr Astigm=yes 6 ==> Lenses=none 5
15. (0.80) Lenses=none 15 ==> Tear-prod=reduced 12
16. (0.75) Astigm=yes Lenses=none 8 ==> Tear-prod=reduced 6
17. (0.75) Spectacles=hmetr Lenses=none 8 ==> Tear-prod=reduced 6
18. (0.75) Age=presb 8 ==> Lenses=none 6
```

Contact lense clauses

```
 1. (.76 .00) Tear-prod=reduced ==> Lenses=none
 2. (.76 .12) Lenses=none ==> Tear-prod=reduced
 3. (.67 .04) Lenses=none ==> Age=presb or Tear-prod=reduced
 4. (.63 .04) Astigm=no and Tear-prod=normal ==> Lenses=soft
 5. (.54 .00) Astigm=no and Tear-prod=normal ==> Age=presb or Lenses=soft
 6. (.50 .08) Astigm=yes and Tear-prod=normal ==> Lenses=hard
 7. (.50 .08) Lenses=none ==> Age=pre-presb or Tear-prod=reduced
 8. (.47 .04) Lenses=none ==> Specs=hmetr or Tear-prod=reduced
 9. (.47 .04) Lenses=none ==> Astigm=yes or Tear-prod=reduced
10. (.47 .00) Lenses=soft ==> Astigm=no
11. (.47 .00) Lenses=soft ==> Tear-prod=normal
12. (.47 .00) Specs=myope and Astigm=yes and Tear-prod=normal ==> Lenses=hard
13. (.47 .00) Lenses=none ==> Age=presb or Specs=hmetr or Tear-prod=reduced
14. (.47 .00) Lenses=none ==> Age=presb or Astigm=yes or Tear-prod=reduced
15. (.45 .00) Specs=hmetr and Astigm=no and Tear-prod=normal ==> Lenses=soft
16. (.44 .29) Astigm=no ==> Lenses=soft
17. (.44 .29) Tear-prod=normal ==> Lenses=soft
```

Trains again

```
(.81 .00) eastbound(A):-hasCar(A,B),clength(B,short),not croof(B,none).
(.62 .20) eastbound(A):-hasCar(A,B),cshape(B,rect),clength(B,short).
(.61 .05) eastbound(A):-hasCar(A,B),hasLoad(B,C),not cshape(B,u_sh),lshape(C,tria).
(.55 .00) eastbound(A):-hasCar(A,B),clength(B,short),croof(B,flat).
(.53 .25) eastbound(A):-hasCar(A,B),clength(B,short),not cshape(B,u_sh).
(.51 .05) eastbound(A):-hasCar(A,B),hasLoad(B,C),cshape(B,rect),lshape(C,tria).
(.51 .20) eastbound(A):-hasCar(A,B),hasLoad(B,C),not croof(B,none),not lshape(C,rect)
(.51 .20) eastbound(A):-hasCar(A,B),hasLoad(B,C),not croof(B,none),not lnumber(C,3).
```

Trains again

```
(.81 .00) eastbound(A):-hasCar(A,B),clength(B,short),not croof(B,none).
(.62 .20) eastbound(A):-hasCar(A,B),cshape(B,rect),clength(B,short).
(.61 .05) eastbound(A):-hasCar(A,B),hasLoad(B,C),not cshape(B,u_sh),lshape(C,tria).
(.55 .00) eastbound(A):-hasCar(A,B),clength(B,short),croof(B,flat).
(.53 .25) eastbound(A):-hasCar(A,B),clength(B,short),not cshape(B,u_sh).
(.51 .05) eastbound(A):-hasCar(A,B),hasLoad(B,C),cshape(B,rect),lshape(C,tria).
(.51 .20) eastbound(A):-hasCar(A,B),hasLoad(B,C),not croof(B,none),not lshape(C,rect)
(.51 .20) eastbound(A):-hasCar(A,B),hasLoad(B,C),not croof(B,none),not lnumber(C,3).
```

	car(A,B), clength(B,short), not croof(B,none)	¬(…)	
eastbound(A)	8 (4)	2 (6)	10
¬eastbound(A)	0 (4)	10 (6)	10
	8	12	20

Trains again

```
(.81 .00) eastbound(A):-hasCar(A,B),clength(B,short),not croof(B,none).
(.62 .20) eastbound(A):-hasCar(A,B),cshape(B,rect),clength(B,short).
(.61 .05) eastbound(A):-hasCar(A,B),hasLoad(B,C),not cshape(B,u_sh),lshape(C,tria).
(.55 .00) eastbound(A):-hasCar(A,B),clength(B,short),croof(B,flat).
(.53 .25) eastbound(A):-hasCar(A,B),clength(B,short),not cshape(B,u_sh).
(.51 .05) eastbound(A):-hasCar(A,B),hasLoad(B,C),cshape(B,rect),lshape(C,tria).
(.51 .20) eastbound(A):-hasCar(A,B),hasLoad(B,C),not croof(B,none),not lshape(C,rect)
(.51 .20) eastbound(A):-hasCar(A,B),hasLoad(B,C),not croof(B,none),not lnumber(C,3).
```

	car(A,B), cshape(B,rect), clength(B,short)	¬(…)	
eastbound(A)	10 (7)	0 (3)	10
¬eastbound(A)	4 (7)	6 (3)	10
	14	6	20

Confirmation as search heuristic

- Theorem: if H´←B´ is a ***proper specialisation*** of H←B, then

$$conf(H' \leftarrow B') \leq (N - n_{H \neg B}) \;/\; (N + n_{H \neg B})$$

	B		¬B		
H	2	(3)	4	(3)	6
¬H	3	(2)	1	(2)	4
	5		5		10

	B´		¬B´		
H´	3	(2.1)	4	(4.9)	7
¬H´	0	(.9)	3	(2.1)	3
	3		7		10

Upgrading to first-order logic

- Use function-free Prolog as representation language
  - normal-form logic, simple syntax
  - specialisation by refinement well understood
- For rule evaluation, generate all grounding substitutions
  - specialisation may increase sample size
  - if problematic, use ISP declarations

The Tertius system

- A*, anytime top-down search algorithm
- optimal refinement operator
- 7500 lines of GNU C
- propositional Weka plug-in available
- P.A. Flach & N. Lachiche (2001), 'Confirmation-guided discovery of first-order rules with Tertius', *Machine Learning* 42(1/2): 61–95
- `www.cs.bris.ac.uk/Research/MachineLearning/Tertius/`

Concluding remarks

- Confirmation-guided rule discovery
  - new heuristic especially suited for knowledge discovery
  - optimal A* search implemented in Tertius
  - applications: subgroup discovery, multiple predicate learning, integrity constraints
- Future work: sampling, multiway contingency tables