What is MRDM?

• Problem: Data in multiple tables
  – Want rules/patterns/etc. across tables
• Solution: Represent as single table
  – Join the data
  – Construct a single view
  – Use standard data mining techniques
• Example: “Customer” and “Married-to”
  – Easy single-table representation
• Bad Example: Ancestor of
Relational Data Network

Basis of Solutions: Inductive Logic Programming

• ILP Rule:
  – customer(CID,Name,Age,yes)
    Age > 30 ∧ purchase(CID,PID,D,Value,PM) ∧
    PM = credit card ∧ Value > 100

• Learning methods:
  – Database represented as clauses (rules)
  – Unification: Given rule (function/clause),
    discover values for which it holds
Example

- How do we learn the “daughter” relationship?
  - Is this classification? Association?
- Covering Algorithm: “guess” at rule explaining only positive examples
  - Remove positive examples explained by rule
  - Iterate

<table>
<thead>
<tr>
<th>Training examples</th>
<th>Background knowledge</th>
</tr>
</thead>
<tbody>
<tr>
<td>daughter(mary, ann).</td>
<td>parent(ann, mary). female(ann).</td>
</tr>
<tr>
<td>daughter(eve, tom).</td>
<td>parent(ann, tom). female(mary).</td>
</tr>
<tr>
<td>daughter(tom, ann).</td>
<td>parent(tom, eve). female(eve).</td>
</tr>
<tr>
<td>daughter(eve, ann).</td>
<td>parent(tom, ian).</td>
</tr>
</tbody>
</table>

How to make a good “guess”

- Clause subsumption:
  Generalize
  - More general clause (daughter(mary,Y) subsumes daughter(mary,ann)
- Start with general hypotheses and move to more specific
Issues

• Search space – efficiency
• Noisy data
  – positive examples labeled as negative
  – Missing data (e.g., a daughter with no parents in the database)
• What else might we want to learn?

WARMR: Multi-relational association rules
Multi-Relational Decision Trees

procedure DIVIDE-AND-CONQUER(TechnicalConditions, DecisionBranches, Examples)
if TERMINATIONCONDITION(Examples) then
  NewLeaf = CREATE-NEW-LEAF(Examples)
  return NewLeaf
else
  PossibleTestsNow = GENERATE-TESTS(TechnicalConditions, DecisionBranches)
  BestTest = FIND-BEST-TEST(PossibleTestsNow, Examples)
  (Split1, Split2) = SPLIT-EXAMPLES(Examples, TechnicalConditions, BestDecision)
  LeftSubtree = DIVIDE-AND-CONQUER(TechnicalConditions, DecisionBranches, Split1)
  RightSubtree = DIVIDE-AND-CONQUER(TechnicalConditions, DecisionBranches, Split2)
  return [BestTest, LeftSubtree, RightSubtree]