Relational Data Mining:
A Quick Introduction

Sašo Džeroski
Institut Jožef Stefan, Ljubljana
Saso.Dzeroski@ijs.si
Overview

- Data mining in a nutshell
  - Data mining tasks
  - Patterns and models
- The single table assumption
- First-order/Relational
  - rule induction
  - tree induction
  - association rules
  - IBL and distance-based clustering
Knowledge Discovery in Databases

• The process of extracting useful knowledge from data
• Involves identifying valid, novel, and useful patterns in data
• The patterns should be ultimately understandable
Data Mining

• Data mining is one step in the KDD process
• It is concerned with finding patterns in data, i.e., applying specific algorithms for extracting patterns from data
• Significant pre-processing and post-processing is required for data mining results to be useful
Data: A single database table

<table>
<thead>
<tr>
<th>Person</th>
<th>Age</th>
<th>Sex</th>
<th>Income</th>
<th>Customer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ann Smith</td>
<td>32</td>
<td>F</td>
<td>10000</td>
<td>yes</td>
</tr>
<tr>
<td>Joan Dew</td>
<td>53</td>
<td>F</td>
<td>1000000</td>
<td>yes</td>
</tr>
<tr>
<td>Mary Stew</td>
<td>27</td>
<td>F</td>
<td>20000</td>
<td>no</td>
</tr>
<tr>
<td>Jane Brown</td>
<td>55</td>
<td>F</td>
<td>20000</td>
<td>yes</td>
</tr>
<tr>
<td>Bob Smith</td>
<td>30</td>
<td>M</td>
<td>100000</td>
<td>yes</td>
</tr>
<tr>
<td>Jack Brown</td>
<td>50</td>
<td>M</td>
<td>200000</td>
<td>yes</td>
</tr>
</tbody>
</table>
Data Mining Tasks

• Predictive modeling: predict a field (class) from some other fields (attributes)
  – classification: predicted field is discrete
  – regression: predicted field is numeric
• Subgroup discovery (given target variable, find groups where its value has unusual distribution)
• Clustering: separate a set of records into subsets, so that records in a subset are similar to each other
• Finding frequent patterns and association analysis
Patterns and models

• Equations and discriminants (linear and non-linear)
• Trees: classification and regression
• Rules: classification and regression
• Frequent itemsets and association rules
• Probabilistic models (networks)
• Partitions of the instance space / clusters
Equations/discriminants

- IF Income + Age - 20027 > 0
  THEN Customer = yes
  ELSE Customer = no
Classification trees

- **Income**
  - < 100000
  - >= 100000

- **Age**
  - < 32
  - >= 32

- **Customer**
  - = yes
  - = no
Classification rules

• IF Income >= 100000
  THEN Customer = yes
  (Support = 3; Confidence = 100%)

• IF Sex = F
  AND Age >= 32
  THEN Customer = yes
  (Support = 2; Confidence = 100%)
Frequent itemsets

• Customer=yes; Income >= 20000 (5)
• Sex=F (4)
• Income >= 100000;
  Income >=100000 AND Customer=yes (3)
• Income =< 20000 AND Age > 27;
  Income =< 20000 AND Age > 27 AND
  Sex = F AND Customer = yes (2)
Association rules

• IF Income <= 20000
  AND Age > 27
  THEN Sex = F
  AND Customer = yes
  (Support: 2; Confidence: 100%)

• The typical example:
  IF beer AND coke
  THEN potato chips AND peanuts
Probabilistic models

- Naïve Bayes: Customer -> Sex; Customer->Income; Customer-> Age
- Customer (yes: 5/6; no: 1/6)
- Customer | Sex
  - Sex=M (yes: 1; no: 0)
  - Sex=F (yes: ¾; no: ¼)
- Bayesian networks
Distance-based methods

• Clustering
  – Hierarchical agglomerative/divisive
  – K-means; k-medoids

• Prediction
  – Nearest neighbor
  – K-Nearest neighbor methods
The single table assumption

- Most data mining methods work on a single table: we have to put all the data in one
- For one-to-one and many-to-one relations, we can join in the extra fields to the original relation without problems
- For one to many relations, problems occur
  - either loss of meaning or
  - loss of information through aggregation
## A tale of two tables

<table>
<thead>
<tr>
<th>Client number</th>
<th>Date of purchase</th>
<th>Magazine purchased</th>
<th>Client number</th>
<th>Name</th>
<th>Address</th>
<th>Age</th>
</tr>
</thead>
<tbody>
<tr>
<td>2303</td>
<td>04-15-94</td>
<td>car</td>
<td>2303</td>
<td>Jones</td>
<td>1 Elm st.</td>
<td>35</td>
</tr>
<tr>
<td>2303</td>
<td>06-21-93</td>
<td>music</td>
<td>2309</td>
<td>Smith</td>
<td>2 Oak dr.</td>
<td>27</td>
</tr>
<tr>
<td>2309</td>
<td>05-30-92</td>
<td>comic</td>
<td>2313</td>
<td>King</td>
<td>3 Low rd.</td>
<td>52</td>
</tr>
<tr>
<td>2313</td>
<td>11-11-11</td>
<td>sports</td>
<td>2319</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2319</td>
<td>11-11-11</td>
<td>house</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Data Integration

Join data from several tables
• One-to one relationships
  (extra information on age, ...)

<table>
<thead>
<tr>
<th>Client number</th>
<th>Date of purchase</th>
<th>Magazine purchased</th>
<th>Age purchased</th>
</tr>
</thead>
<tbody>
<tr>
<td>2303</td>
<td>04-15-94</td>
<td>car</td>
<td>35</td>
</tr>
<tr>
<td>2303</td>
<td>06-21-93</td>
<td>music</td>
<td>35</td>
</tr>
<tr>
<td>2309</td>
<td>05-30-92</td>
<td>comic</td>
<td>27</td>
</tr>
<tr>
<td>2313</td>
<td>NULL</td>
<td>sports</td>
<td>52</td>
</tr>
<tr>
<td>2303</td>
<td>NULL</td>
<td>house</td>
<td>35</td>
</tr>
</tbody>
</table>
Data Integration (ctd)

Aggregate data from several tables

- **One-to-many relationships**
  (e.g. number of purchases)

<table>
<thead>
<tr>
<th>Client number</th>
<th>Last purchase</th>
<th>Number of mags.</th>
<th>Age</th>
</tr>
</thead>
<tbody>
<tr>
<td>2303</td>
<td>04-15-94</td>
<td>3</td>
<td>35</td>
</tr>
<tr>
<td>2309</td>
<td>05-30-92</td>
<td>1</td>
<td>27</td>
</tr>
<tr>
<td>2313</td>
<td>NULL</td>
<td>1</td>
<td>52</td>
</tr>
</tbody>
</table>
(Multi-)Relational Data Mining

- DM: Finding patterns in data
- (M)RDM: Finding patterns in (multi-)relational data; Note that relational databases are really multi-relational databases
- Patterns involving multiple relations are typically expressed in relational/first-order logic
  - this is more powerful than formalisms used by single table data mining methods
    - variables
    - recursion
Database and FOL/LP terms

- Relation name
- Attribute of relation
- Tuple \((a_1, \ldots, a_n)\) of relation \(p\)
- Relation \(p\) as set of tuples
- Relation \(p\) defined as a view
- Predicate symbol
- Argument of predicate
- Ground fact \(p(a_1, \ldots, a_n)\) of predicate \(p\)
- Predicate \(p\) defined extensionally
- Predicate \(p\) defined intensionally
A database with two relations

<table>
<thead>
<tr>
<th>Person</th>
<th>Age</th>
<th>Sex</th>
<th>Income</th>
<th>Customer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ann Smith</td>
<td>32</td>
<td>F</td>
<td>10000</td>
<td>yes</td>
</tr>
<tr>
<td>Joan Gray</td>
<td>53</td>
<td>F</td>
<td>1000000</td>
<td>yes</td>
</tr>
<tr>
<td>Mary Stew</td>
<td>27</td>
<td>F</td>
<td>20000</td>
<td>no</td>
</tr>
<tr>
<td>Jane Brown</td>
<td>55</td>
<td>F</td>
<td>20000</td>
<td>yes</td>
</tr>
<tr>
<td>Bob Smith</td>
<td>30</td>
<td>M</td>
<td>100000</td>
<td>yes</td>
</tr>
<tr>
<td>Jack Brown</td>
<td>50</td>
<td>M</td>
<td>200000</td>
<td>yes</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Husband</th>
<th>Wife</th>
<th>Marriages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bob Smith</td>
<td>Ann Smith</td>
<td></td>
</tr>
<tr>
<td>Jack Brown</td>
<td>Jane Brown</td>
<td></td>
</tr>
</tbody>
</table>
Relational Data Mining Example

• IF People(Person,Income,Age,Sex,Customer)
  AND Income >= 100000
  THEN Customer = yes

• IF People(Person,Income,_,_,_,_)
  Income >= 100000,
  AND Marriages(Person,Wife)
  AND People(Wife,_,_,_,_,Customer)
  THEN Customer = yes
Rule induction

• Most common form of ILP
• Typically two classes + and -
• Rules are clauses in first-order logic
  – grandfather(X,Y) :- father(X,Z), parent(Z,Y)
• Many rule induction systems exist
  – FOIL (Quinlan 1990)
  – PROGOL (Muggleton 1995)
An example database

mother(blaguna,saso).
father(veljo,saso).
parent(blaguna,saso).

mother(blaguna,sonja).
father(veljo,sonja).
parent(blaguna,sonja).

parent(veljo,saso).
parent(veljo,sonja).

---------------

male(veljo).
female(blaguna).
human(veljo).

male(saso).
female(sonja).
human(saso).

female(sonja).
human(blaguna).
ILP - Rule induction

- Rules define relations in terms of other relations
  
  \[\text{parent}(X,Y) :- \text{mother}(X,Y)\]
  \[\text{parent}(X,Y) :- \text{father}(X,Y)\]

- Rules can also express constraints about relations

  \[\text{female}(X) :- \text{mother}(X,Y)\]
  \[\text{male}(X) :- \text{father}(X,Y)\]
  
  IF X is the mother of Y, X must be female
A biodegradability example

Representing compounds in the atom-bond formalism

atom(AtomID,Element,AtomType,Charge)

bond(Atom1,Atom2,BondType).

slow.

atom(1,c,10,0.388).
atom(2,c,10,0.388).
atom(3,cl,93,-0.212).
atom(4,cl,93,-0.212).
atom(5,h,3,0.037).
atom(6,cl,93,-0.213).
atom(7,cl,93,-0.213).
atom(8,h,3,0.037).

bond(1,2,1).
bond(1,3,1).
bond(1,4,1).
bond(1,5,1).
bond(2,6,1).
bond(2,7,1).
bond(2,8,1).
Biodegradability rules

slow OR moderate :-
atom(A1,Elem1,Type1,Charge1), Elem1 = n, Charge1 < 0.8

• The biodegradability rate of the compound is slow or moderate if the compound contains a nitrogen atom of charge less than 0.8

slow OR moderate :-
atom(A1,Elem1,Type1,Charge1), Type1 = 1,bond(A5,A6,7)

• The biodegradability rate of the compound is slow or moderate if the compound contains a hydrogen atom (Type = 1) and an aromatic bond (7)
Background knowledge for chemical compounds

- An example: carbon rings
  - Benzene ring: benzene(RingList)
  - Carbon six ring (non aromatic): carbon_6_ring(RingList)
  - Carbon five aromatic ring:
    carbon_5_aromatic_ring(RingList)
  - Carbon five ring (non aromatic):
    carbon_5_ring(RingList)

benzene([C1,C2,C3,C4,C5,C6]) :-
  bond(C1,C2,7), bond(C2,C3,7), bond(C3,C4,7),
  bond(C4,C5,7), bond(C5,C6,7), bond(C6,C1,7).

...
RDM – Key approaches

• Transforming RDM problems to propositional form
• Inverting deductive reasoning in FOL (inverting resolution, entailment)
• Adapting/upgrading propositional approaches to FOL
Decision trees in ILP

• First order logical decision trees
  – Binary trees
  – Each node contains a logical conjunction
  – Nodes can share variables
  – Test in node = conjunction in node + conjunctions on path from root to node

• Complex semantics

• High expressiveness
Example FOLDT

\[
(\forall x: \neg \text{worn}(x)) \implies \text{ok}
\]
\[
(\exists x: \text{worn}(x) \land \text{irreplaceable}(x)) \implies \text{sendback}
\]
\[
(\exists x \forall y: \text{worn}(x) \land \neg (\text{worn}(y) \land \text{irreplaceable}(y))) \implies \text{fix}
\]
Expressiveness

FOL formula equivalent with tree:

\[(\forall x: \neg \text{worn}(x)) \implies \text{ok}\]
\[(\exists x: \text{worn}(x) \land \text{irreplaceable}(x)) \implies \text{sendback}\]
\[(\exists x \forall y: \text{worn}(x) \land \neg (\text{worn}(y) \land \text{irreplaceable}(y))) \implies \text{fix}\]

Logic program equivalent with tree:

\[a \leftarrow \text{worn}(X)\]
\[b \leftarrow \text{worn}(X), \text{irreplaceable}(X)\]
\[\text{ok} \leftarrow \neg a\]
\[\text{sendback} \leftarrow b\]
\[\text{fix} \leftarrow a \land \neg b\]
Decision trees in ILP - example

<table>
<thead>
<tr>
<th>Name</th>
<th>Job</th>
<th>Speed</th>
<th>Fine?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ann</td>
<td>teacher</td>
<td>150</td>
<td>Y</td>
</tr>
<tr>
<td>Bob</td>
<td>politician</td>
<td>160</td>
<td>N</td>
</tr>
<tr>
<td>Chris</td>
<td>engineer</td>
<td>120</td>
<td>N</td>
</tr>
<tr>
<td>Dave</td>
<td>writer</td>
<td>155</td>
<td>N</td>
</tr>
<tr>
<td>Earnest</td>
<td>politician</td>
<td>120</td>
<td>N</td>
</tr>
</tbody>
</table>

(AVL) Speed > 120 and Job ≠ politician => Fine? = Y

(ILP) speed(x,s), s > 120, not job(x, politician), not (knows(x,y), job(y, politician)) => fine(x,Y)
speed(x, s), s > 120, not job(x, politician), not (knows(x, y), job(y, politician)) => fine(x, Y)
Predictive Clustering Trees

• Can perform classification, regression and clustering
• Conceptual clustering, an explicit description of the cluster hierarchy produced
• Also simultaneous prediction of several target variables
• Implemented in the ILP system TILDE (Blockeel 1998)
Relational IBL and Clustering

• Relational distance measure: when calculating similarity between two objects, takes into account similarity of related objects, e.g., between the children of the compared persons

• RIBL (Emde and Wettschereck 1996)

• Relational Distance-Based Clustering (RDBC) (Kirsten and Wrobel 1998)
First-order Association Rules

- IF `win(W),start(W,A),in_window(W,120,B), alarm_class(B,bsc_message)` THEN `in_window(W,120,C),alarm_class(C,trans), same_urgency(A,C),same_urgency(C,B)`

- Example from handling alarms in telco nets:
  ‘If a window starts with an alarm A and contains alarm B of class bsc_message then it will also contain alarm C of class trans of same urgency as the alarms A and B.’

- WARMR (Dehaspe 1998)

- Generalizes association rules/sequential patterns
Multi-Relational Discovery of Subgroups

- patient(PID, Name, Age, Sex, Outcome)
- patient_diagnosis(PID, DID, Date, HID)
- hospital(HID, Name, Location, Size, Owner, Class)

Interesting subgroup: ‘Patients older than 65 who were diagnosed at a small hospital have an unusually high mortality rate.’

patient(PID, _, A, _, _), A > 65,
patient_diagnosis(PID, _, _, H),
hospital(H, _, _, small, _, _)
Successful Applications

• In bioinformatics
  – Drug design: predicting activity of compounds
  – Predicting mutagenicity, carcinogenicity
  – Predicting protein structure and function

• Environmental applications
  – Predicting biodegradability; predicting physical/chemical parameters of river water quality from bioindicator data

• Mechanical and traffic engineering

• Growing interest in text and web mining apps
RDM Summer School Program

• An introduction to RDM: Saso Dzeroski
• An introduction to ILP and propositionalization: Peter Flach & Nada Lavrac
• Upgrading propositional learners to a relational setting: Luc De Raedt
• Logical decision trees: Hendrik Blockeel
• Relational subgroup discovery: Stefan Wrobel
• Relational distance-based methods: Stefan Wrobel
• Kernel-Based Learning from Structured Data: Thomas Gaertner

• Learning Statistical Models from Relational Data: Lise Getoor
• Bayesian logic programs: Luc De Raedt and Kristian Kersting
• Applications of RDM to bioinformatics: Ross King
• Overview of RDM applications: Saso Dzeroski
• Inductive databases: Luc De Raedt

• Future research / open issues in ILP/RDM: Panel discussion