Weka tutorial and Inductive Logic Programming

Louis Oliphant
oliphant@cs.wisc.edu
CS 540 section 2

Announcements

- Projects Due Thursday
- Homework 5 due next Thursday
- Final
  - December 19th
  - Room 1325 (classroom)
  - 10:05am
  - Emphasis on information since midterm

Weka – What you should know

- arff file format
  - loading files
  - looking at basic statistics of data
- classifying
  - picking classifier
    - setting parameters
  - methods of evaluation
- clustering
  - k-means
  - comparison to classification
- There is a lot more to weka than this

Weka

- watch in class tutorial

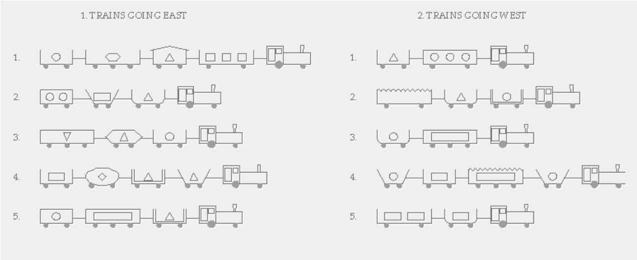
- additional information at weka site: http://www.cs.waikato.ac.nz/ml/weka/

Inductive Logic Programming – Motivation

- So far all models
  - have used fixed length feature vectors
  - have a fixed, previously decided, class feature (except Bayesian networks)
- We have been learning propositional logic models
- But a lot of real world data doesn't fit this paradigm, you can make the data fit into a fixed length feature vector

Staring Example

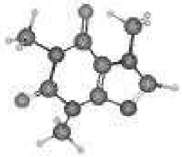
example of East-West trains (Michalski)

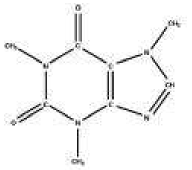


What makes a train go Eastward?

Real World Example

Mutagenicity of chemical molecules
(King, Srinivasan, Muggleton, Sternberg, 1994)





What makes a molecule to be mutagenic ?

ILP can use Multiple Tables

Many feature vectors

has_car

Train	Car
t1	c11
t1	c12
t1	c13
t1	c14
t2	c21
…	…

car_properties

Car	Length	Shape	Axes	Roof	…
c11	short	rectangle	2	none	…
c12	long	rectangle	3	none	…
c13	short	rectangle	2	peaked	…
c14	long	rectangle	2	none	…
c21	short	rectangle	2	flat	…
…	…	…	…	…	…

ILP Setup

- First Order Definite Clauses
- Learns a Theory of clauses (rules)
- Using:
  - Background Knowledge and
  - Examples (positive and negative)

Example

- Learn grandparent(X,Y) relation
- Background Knowledge:
  - father(philip,charles) father(philip, anne)
  - mother(mum, margaret) mother(mum, elizabeth)
  - married(diana, charles) married(elizabeth, philip)
  - male(philip) male(charles)
  - female(beatrice) female(margaret)
  - parent(X,Y):-father(X,Y).
  - parent(X,Y):-mother(X,Y).
- Examples:
  - grandparent(mum, charles) grandparent(elizabeth, beatrice)
  - ¬grandparent(mum, harry), ¬grandparent(spencer, peter)

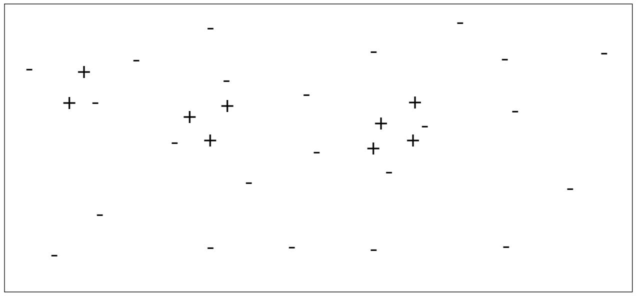
ILP Algorithm

- ILP is a covering algorithm:

```
theory=empty
While (positives left uncovered):
   construct a rule
   theory=theory+rule
   remove positives covered by rule
return theory
```

ILP Algorithm

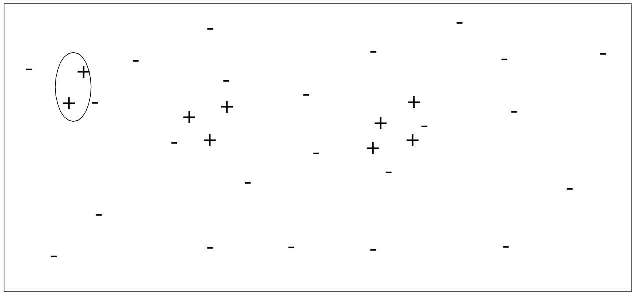
- Covering Algorithms
  while (positives left uncovered)



ILP Algorithm

- Covering Algorithms

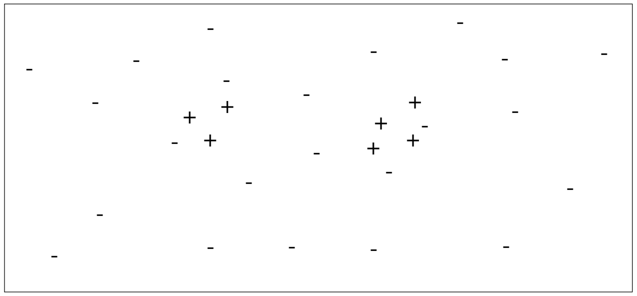
  construct a rule
  theory=theory+rule



ILP Algorithm

- Covering Algorithms

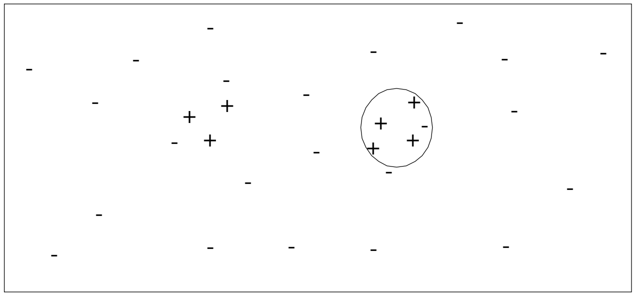
  remove positives covered by rule



ILP Algorithm

- Covering Algorithms

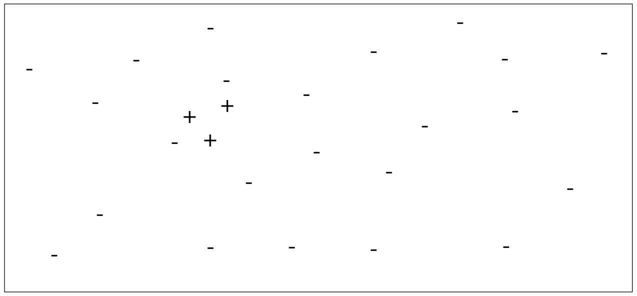
  learn a rule
  theory=theory+rule



ILP Algorithm

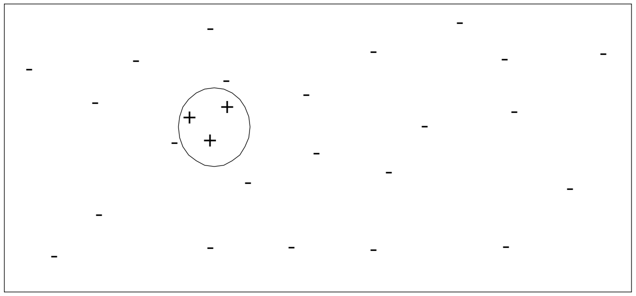
- Covering Algorithms

  remove positives covered by rule



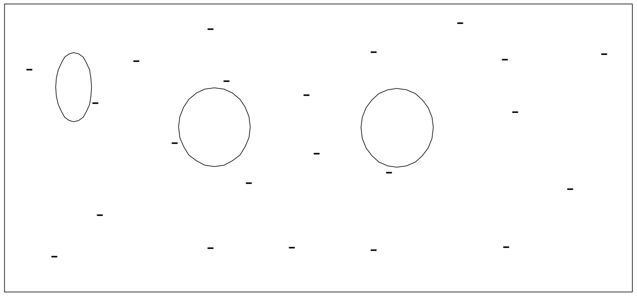
ILP Algorithm

- Covering Algorithms
  learn a rule
  theory = theory+rule



ILP Algorithm

- Covering Algorithms
  return theory



Learning a Rule

- Several Methods for learning Rules:
- Top Down Approaches
  - start with most general rule (cover everything)
  - specialize the rule until it covers only (mostly) positives
- Bottom Up Approaches
  - start with most specific rule (covers a single example)
  - generalize the rule as long as it doesn't cover any (to many) negatives

Learning a Rule

- Top Down Approaches
  - FOIL (1993)
  - PROGOL (1995)
- Bottom Up Approaches
  - GOLEM (1992)

FOIL

- Start with most general rule:
  grandparent(X,Y):-.
- Consider adding all possible predicates to the body of the rule:
  grandparent(X,Y):-parent(X,Y).
  grandparent(X,Y):-parent(X,Z).
  grandparent(X,Y):-female(X).
- score each possible rule using foil_gain (similar to information gain)
- Keep the rule that has the highest score
- Repeat until you have a rule that covers only (mostly) positives

PROGOL

- Pick a positive seed example from the remaining uncovered positive examples:
  - grandparent(mary,john)
- Create a list of all true information about that seed example
  - female(mary), male(john), parent(mary,fred),male(fred), parent(george,mary),parent(fred,john),...
- Variabilize the information
  - female(M),male(J),parent(M,F),male(F),parent(G,M), parent(F,J)...
- Only consider adding variabilized predicates from this list (bottom clause)
- This guarantees that the seed example (at least) will be covered by the rule

GOLEM

- Pick several pairs of positive examples
- Construct the Least General Generalization for each pair
  - List ALL the things the two examples have in common
- Select the one that covers the most positives
- Now select a new set of positive examples
- Construct the Least General Generalization between each new positive and the current rule
- Keep the one that covers the most positives
- Repeat as long as not covering any (to many) negatives

Conclusion

- Weka
- Benefits of Inductive Logic Programming (ILP)
- Covering Algorithms
- top down / bottom up approaches to rule generation
- FOIL, PROGOL, and GOLEM