

Inductive Logic Programming



JERZY STEFANOWSKI

Institut Informatyki
Politechnika Poznańska

Plan

1. Wprowadzenie do ILP
2. Przykładowe problemy
3. Podstawowe metody
4. Algorytm FOIL
5. Zastosowania
6. Dyskusja ograniczeń
7. Uwagi końcowe



Motywacje

- Techniki uczenia maszynowego tradycyjnie wykorzystują reprezentacje (atrybut-wartość):
 - Prostota reprezentacji,
 - Efektywność i łatwość przetwarzania,
 - Istnienie wielu metod uwzględniania niedoskonałości danych,
 - Zaproponowano wiele paradygmatów,....
- Czy to jest wystarczające?
- Ograniczenia:
 - języka reprezentacji - brak możliwości uwzględniania **relacji** zachodzących między obiektami lub częściami obiektów,
 - **Wiedza dziedzinowa** (ang. *background knowledge*) może być wyrażana w ograniczonej postaci.

Indukcyjne programowanie logiczne

- Ograniczenia reprezentacji (atrybut-wartość) doprowadziły do rozwoju dziedziny **Indukcyjnego Programowania Logicznego** (ang. ILP).
 - Podstawa reprezentacji wiedzy → **logika predykatów**,
 - także, możliwość stosowania zmiennych.
 - Uczenie się → znalezienie logicznej formuły opisującej pojęcie docelowe w zależności od innych relacji określonej w tej dziedzinie, najczęściej reprezentowana przez zbiór klauzul.
 - ILP → oferuje
 - Bogatszą i deklaratywną reprezentację (także postulat zrozumiałości),
 - „Silne” podstawy formalne (logiki)
 - Łatwość uwzględniania wiedzy dziedzinowej
 - Powiązanie z proceduralnymi programami (Prolog i ska)
- Lecz także ograniczenia (...)

Reprezentacja w logice predykatów

- Reprezentacje zdaniowe:
 - przykład → **stałej długości** wektor **wartości atrybutów**,
 - Atrybuty zawarte w definicji zbioru danych.
- Reprezentacje w logice pierwszego rzędu:
 - Przykład → **ustrukturalizowany** obiekt o zmiennej wielkości
 - Sekwencja, zbiór elementów, graf,...
 - Hierarchiczny, np., zbiór sekwencji.
 - Atrybuty są **wybirane** z potencjalnie bardzo dużego zbioru.

Zadanie indukcyjnego programowania logicznego

Dane

- zbiór przykładów pozytywnych E^+
- zbiór przykładów negatywnych E^-
- wiedza dziedzinowa (background knowledge) B , wyrażona jako zbiór definicji predykatów

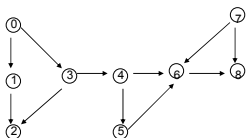
Zadanie → skonstruować formułę logiczną H (hipotezę) taką, że

- wszystkie przykłady są pozytywne E^+ można logicznie wyprowadzić z $B \wedge H$ (pełność)
- żadnego z przykładów negatywnego E^- nie można wyprowadzić z $B \wedge H$ (spójność)

Uwaga: E^+ nie jest logicznie wyprowadzalny z samego B

Dlaczego ILP? (slide za S.Matwin)

- expressiveness of logic as representation (Quinlan)

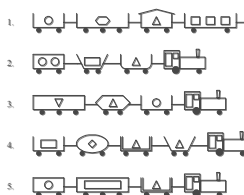


- can't represent this graph as a fixed length vector of attributes
- can't represent a "transition" rule:

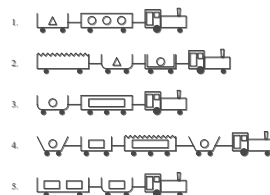
A can-reach B if A link C, and C can-reach B
without variables

East-West trains – „sztandarowy” przykład ILP

1. TRAINS GOING EAST



2. TRAINS GOING WEST



„Pociągi” – reprezentacja w rachunku predykatów

- Example:** eastbound(t1).

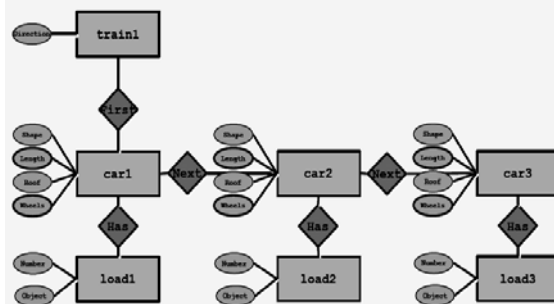
Background theory:

car(t1,c1).	car(t1,c2).	car(t1,c3).	car(t1,c4).
rectangle(c1).	rectangle(c2).	rectangle(c3).	rectangle(c4).
short(c1).	long(c2).	short(c3).	long(c4).
none(c1).	none(c2).	peaked(c3).	none(c4).
two_wheels(c1).	three_wheels(c2).	two_wheels(c3).	two_wheels(c4).
load(c1,l1).	load(c2,l2).	load(c3,l3).	load(c4,l4).
circle(l1).	hexagon(l2).	triangle(l3).	rectangle(l4).
one_load(l1).	one_load(l2).	one_load(l3).	
three_loads(l4).			

Hypothesis:

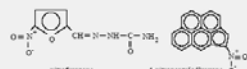
eastbound(T) :- car(T,C), short(C), not none(C).

Pociągi – sekwencja



PREDICTING MUTAGENICITY

Nitro-aromatic compounds



New structural alert



Examples: 188 regression-friendly and 42 unfriendly compounds

Background knowledge: atoms, bonds, atom groups and properties

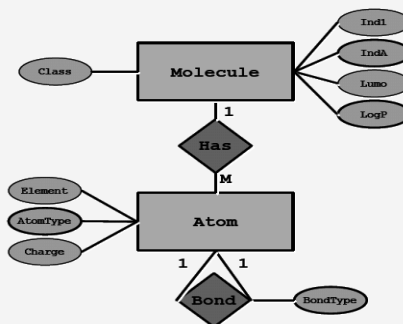
ILP system applied: PROGOL

A new structural alert for high mutagenicity discovered.

Better than regression on unfriendly set (88% vs 69%), comparable on regression-friendly set (88% vs 89%)

Predykcja mutagenów

Mutagenesis



Predykcja mutageniczności

- Zastosowanie do problemów SAR w farmacji
- Tradycyjnie stosowane podejścia nie uwzględniają strukturalnych zależności w cząsteczce
- Wiedza dziedzinowa nt. budowy atomowej, wiązań i siły wiązania
- Dla „regression unfriendly compounds” reguła:

Związek o wysokiej mutageniczności, jeśli

$LUMO \leq -1,937$ lub ($LUMO \leq -1,57$ i atom węgla łączy sześćoelementowy pierścień aromatyczny) lub ($LUMO \leq -1,176$ i aryl-aryl wiązanie między pierścieniami beznynu)

LUMO to energia najniższej zajętej orbity w cząsteczce

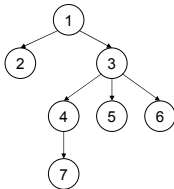
W jaki sposób użyć rachunku predykatów w ILP?

- Rozważmy problem relacji rodzinnych.
 - Cel → nauczyć się pojęcia „dziadek”
 $grandparent(X, Y)$ X jest dziadkiem Y
 - Nauczyciel dostarcza przykładów pozytywnych tego pojęcia:
 $grandparent(1,4)$, $grandparent(1,5)$, $grandparent(1,6)$,
 $grandparent(3,7)$.
- gdzie „1”, „2”... oznaczają pewne imiona.
- Wszystkie inne relacje $grandparent$ między osobami 1,...,7 są przykładami negatywnymi.
 - Co będzie wiedzą dziedzinową?

Przykład relacji rodzinnych – cd.

- Relacje rodzinne opisujemy z wykorzystaniem predykatu $parent(X, Y)$, co oznacza, że X jest rodzicem Y.
- Drzewo relacji w rozważanej rodzinie jest zakodowane w postaci relacji:

$parent = \{(1,2), (1,3), (3,4), (3,5), (3,6), (4,7)\}$



Uczenie się klauzul z relacji w danych

- Założenia: w wiedzy B dostępny jest wyłącznie predykat $parent$, można używać zmiennych.
 - Jeśli ograniczyć się do zmiennych wykorzystywanych w konkluzji klauzuli to możliwe definicje pojęcia $grandparent$ są następujące:
 $grandparent(X, Y) :- parent(X, Y).$
 $grandparent(X, Y) :- parent(Y, X).$
 $grandparent(X, Y) :- parent(X, X).$
 $grandparent(X, Y) :- parent(Y, Y).$
- Żadna z nich nie pokrywa przykładów pozytywnych.
- Zmieńmy ograniczenia!

Uczenie się klauzul z relacji w danych

- Ograniczenia – wolno użyć jednego więcej argumentu, który nie występuje w konkluzji klauzuli.
- Można wprowadzić cztery nowe literały wykorzystujące nową zmienną Z: $parent(X, Z)$, $parent(Y, Z)$, $parent(Z, X)$ oraz $parent(Z, Y)$.
- Założmy, że wybrano hipotezę:
 $grandparent(X, Y) :- parent(X, Z).$
- Należy ją ocenić korzystając z podstawień dla trójki (X, Z, Y) .
- W przykładzie istnieje 7 = 343 możliwych realizacji (X, Z, Y)

Lecz dla danych powiązań rodzinnych tylko poniższe przykłady są zgodne z przykładami negatywnymi:

$(1,2,4)$, $(1,2,5)$, $(1,2,6)$, $(1,3,4)$, $(1,3,5)$, $(1,3,6)$, $(3,4,7)$, $(3,5,7)$, $(3,6,7)$

Inne trójki są negatywne, np. $(1,2,1)$

Uczenie się klauzul z relacji w danych

- Dotychczasowa klauzula
 $grandparent(X, Y) :- parent(X, Z)$ jest niespójna
 - Dokonaj dalszej specjalizacji, np.
 $grandparent(X, Y) :- parent(X, Z), parent(Z, Y)$
 - Klauzula pokrywa następujące trójki (X, Z, Y)
 $\oplus (1,3,4)(1,3,5)(1,3,6)(3,4,7)$
– żadnego negatywnego przykładu.
- Można pozostać z powyższą hipotezą
- Skąd wiadomo, które predykaty i zmienne dodawać?
np. $grandparent(X, Y) :- parent(X, Z), parent(Z, Y)$ nie jest dobrym kandydatem!
 - Jak realizować uczenie się klauzul systematycznie?

Wybrane zastosowania

- Farmacja i bioinformatyka
 - Predykcja mutageniczności związków chemicznych
 - Projektowanie nowych związków chem. / leków (tzw. Structure/Activity Relationships)
 - Predykcja struktury białek i ich biologicznej funkcji
- Mechanika i projektowanie inżynierskie
 - Metoda elementów skończonych (tzw. mesh)
 - Analiza sterowania procesami technologicznymi
- Zastosowanie w ochronie środowiska
 - Klasyfikacja wody w rzekach, predykcja biodegradacji związków chemicznych.
- Przetwarzanie języka naturalnego
 - Automatyczna konstrukcja parserów j. naturalnego,
 - Tłumaczenie zapytań w j. naturalnych do dedukcyjnych baz danych
 - Uczenie się „past tense” czasowników w j. ang., itp.
 - Analiza morologiczna, ... (także lematyzacja)
- Text- / Web-mining

FINITE ELEMENT MESH DESIGN

Given a geometric structure and loadings/boundary conditions
Find an appropriate resolution for a finite element mesh

Examples: ten structures with appropriate meshes (cca. 650 edges)

Background knowledge

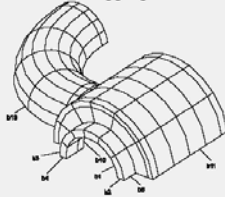
- Properties of edges (short, loaded, two-side-fixed, ...)
- Relations between edges (neighbor, opposite, equal)

ILP systems applied: GOLEM, CLAUDIEN

Many interesting rules discovered (according to expert evaluation)

Finite element mesh design (ctd.)

Example structure with an appropriate mesh



Example rules

```
mesh(Edge, 7) ← usual_length(Edge),  
neighbour_xy(Edge, EdgeY), two_side_fixed(EdgeY),  
neighbour_zx(EdgeZ, Edge), not_loaded(EdgeZ)  
mesh(Edge, N) ← equal(Edge, Edge2), mesh(Edge2, N)
```

Warmr

- **First order association rule :**
 - IF Query1 THEN Query2
 - Shorthand for
 - IF Query1 THEN Query1 and Query2
 - to obtain variable bindings
- IF ?- participant(P,C,PA,X), course(P,Y,advanced) THEN ?- PA=no
- IF ?- participant(P,C,PA,X), course(P,Y,advanced) succeeds for P THEN ?- participant(P,C,PA,X), course(P,Y,advanced), PA=no succeeds for P

Other upgrades

- **Bayesian Logic Programs and PRMS**
 - Upgrade Bayesian Nets
- **Stochastic Logic Programs (Muggleton) upgrade stochastic context free grammars**
- **Merlin (Bostrom) upgrades induction of finite state automata to class of logic programs**