CSE 505 - Computing with Logic

Project - Phase 2

Charuta Pethe
111424850

A New Algorithm to Automate Inductive Learning of Default Theories

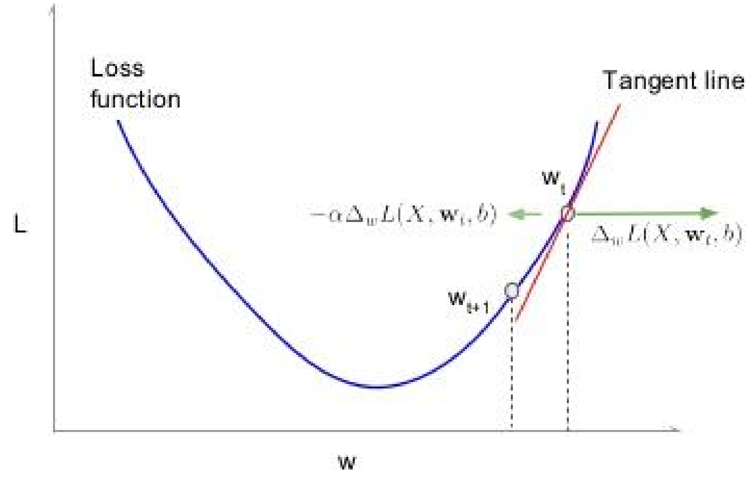
Farhad Shakerin, Elmer Salazar, Gopal Gupta

ICLP (International Conference on Logic Programming)

August 2017

Introduction

Classical ML methods



Algebraic solutions to optimization problems

Hard to understand and verify!

No intuitive description

Want to extend prior knowledge?

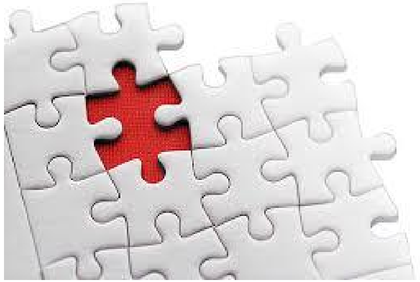
Relearn!

Introduction

Inductive Logic Programming

Learns Horn logic rules

No negation-as-failure



Not sufficiently expressive when background knowledge is incomplete!

Assumption!

Often, exceptions themselves follow a pattern, and can be learned

Introduction

Default theory

Describes the underlying model more accurately

Allows us to reason in absence of information

New Algorithm

First Order Learner of Default

FOLD: for categorical features

FOLD-R: for numeric features

Inductive Learning Problem

Given:

`B`	Clauses of the form $h \leftarrow l1, \ldots, lm, \text{not } lm{+}1, \ldots, \text{not } ln$
`E+, E-`	Two disjoint sets of positive and negative examples
`L`	Hypothesis language of predicates
`covers(H, E, B)`	Function which returns subset of E which is extensionally implied by the hypothesis H, given background knowledge B

Find:

`T`	A theory for which covers(T, E+, B) = E+ and covers(T, E-, B) = ϕ

FOLD Algorithm

Explanation with Example

B:
```
bird(X) :- penguin(X).
bird(tweety).   bird(coco).
cat(kitty).     penguin(polly).
```

Goal:
```
fly(X).
```

E+:
```
{tweety, coco} i.e. fly(tweety). fly(coco).
```

E-:
```
{kitty, polly}
```

FOLD Algorithm

Explanation with Example

Start with the general rule `goal(X1...Xn) ← true`

In each specialization step, rule out negative examples covered without significantly decreasing the number of positive examples covered.

Stop when information gain becomes 0.

If negative examples are still covered, call FOLD recursively to learn exception rules.

FOLD Algorithm

Information Gain

$$IG(L,R) = t\left(log_2\frac{p_1}{p_1+n_1} - log_2\frac{p_0}{p_0+n_0}\right)$$

- L is the candidate literal to add to rule R
- p_0 is the number of positive examples implied by the rule R
- n_0 is the number of negative examples implied by the rule R
- p_1 is the number of positive examples implied by the rule R + L
- n_1 is the number of negative examples implied by the rule R + L
- t is the number of positive examples implied by the rule R and covered by the rule R + L

FOLD Algorithm

Explanation with Example

Start with `fly(X) ← true`

covers `{tweety, coco, kitty, polly}`

Add the rule `fly(X) ← bird(X)`

covers `{tweety, coco, polly}`

Call FOLD recursively with E+ as `{polly}`

Returns `fly(X) ← penguin(X)` as EXCEPTION

FOLD Algorithm

Explanation with Example

Add the exception as an abnormality rule, and its negation in the default rule's body.

Final output:

```
fly(X) ← bird(X), not ab0(X).
ab0(X) ← penguin(X).
```

FOLD Algorithm

Explanation with Example

If there is noisy input data, enumeration is done when:

1. Information gain = 0 for all positive literals
2. No rule governing the negative examples can be found

If we add `jet` to the set E+, FOLD gives the output:

```
fly(X) ← bird(X), not ab0(X).
fly(X) ← member(X, [jet]).
ab0(X) ← penguin(X).
```

FOLD-R Algorithm

Numeric Extension of FOLD

For numeric features, introduce constraints such as

$\{A \leq h,\ A > h\}$

Use the algorithm C4.5 to find the best **numeric literal**, **arithmetic constraint** and **threshold** - which maximize the Information Gain.

FOLD-R Algorithm

Explanation with Example

Outlook	Temperature	Humidity	Wind	PlayTennis
sunny	75	70	True	Yes
sunny	80	90	True	No
sunny	85	85	False	No
sunny	72	95	False	No
sunny	69	70	False	Yes
overcast	72	90	True	Yes
overcast	83	78	False	Yes

FOLD-R Algorithm

Explanation with Example

Outlook	Temperature	Humidity	Wind	PlayTennis
overcast	83	65	True	Yes
overcast	81	75	False	Yes
rain	71	80	True	No
rain	65	70	True	No
rain	75	80	False	Yes
rain	68	80	False	Yes
rain	70	96	False	Yes

FOLD-R Algorithm

Explanation with Example

Output

```
play(X) ← overcast(X).
play(X) ← temperature(X, A), A ≤ 75, not ab0(X).
ab0(X) ← windy(X), rainy(X).
ab0(X) ← humidity(X, A), A ≥ 95, sunny(X).
```

Exceptions to the rule:

1. It is rainy and windy
2. It is sunny and the humidity is above 95%

Roadmap

Over the course of this project, I plan to:

1. Implement FOLD in Python
2. Implement C4.5 in Python
3. Implement FOLD-R in Python
4. Create my own examples to test FOLD and FOLD-R
5. List the ideal outputs for my examples
6. Run the algorithms on my examples
7. Compare ideal outputs against experimental outputs
8. Note the results and analyze them

Questions?

?-