CS 539 Project 8
Rule Learning

Chris Winsor

Target Datasets

We analyze characteristics of "Rule Based Learning" techniques (FOIL and JRIP) using 10 different datasets.

Ackermann Dataset

The dataset represents a mathematical function which grows more rapidly than exponent.

FOIL

Generates 3 clauses with a total of 8 terms:

- Ackermann(0,B,C) :- succ(B,C).
- Ackermann(A,0,C) :- succ(0,D), succ(E,A), Ackermann(E,D,C).
- Ackermann(A,B,C) :- succ(D,A), succ(E,B), Ackermann(A,E,F), Ackermann(D,F,C).

JRIP

Does not apply (target class is numeric)

CRX (credit card) Dataset

The dataset provides attributes of a credit card applicant. The target class identifies if the individual is a good or bad candidate.

FOIL:

Generates 19 clauses with a total of 23 terms.
Summary: 52 errors in 200 trials 26%

JRIP

Generates a model with 5 rules:
Incorrectly Classified Instances 65 13.2653 %

Member Dataset

The dataset expresses a set relationship where the target identifies if the first attribute is in the list described by the second attribute.

FOIL:
Generates a model with two clauses and 3 terms:

- member(A,B) :- components(B,A,C).
- member(A,B) :- components(B,C,D), member(A,D).

Time 0.0 secs
Summary: 1 error in 4 trials 25% (this is explained as a failure of the FOIL code, not the FOIL search algorithm).

JRIP
Generates a model with 4 rules:

- (l_2 = 2.0) => member=- (6.0/0.0)
- (l_2 = x) => member=- (7.0/2.0)
- (l_3 = 2.0) => member=- (3.0/0.0)
- => member=+ (25.0/5.0)

Incorrectly Classified Instances 15 36.5854 %

nxm (n-choose-m) Dataset

The dataset expressed a function whereby given a set of size N, how many combinations of size M exist. The equation for this function is n! / (k! (n-k)!)).

FOIL
Generates a model with 2 clauses and a total of 10 terms:

- choose(A,B,C) :- dec(C,D), mult(A,B,E), mult(B,C,F), mult(B,D,D), mult(B,F,E).
- choose(A,B,C) :- dec(A,D), dec(B,E), mult(B,C,F), choose(D,E,G), mult(A,G,F).

Time 0.0 secs

JRIP
Generates a model with 2 rules:

- (Z <= 1) and (M <= 0) => ZisNcM=yes (6.0/0.0)
- => ZisNcM=no (204.0/15.0)

Incorrectly Classified Instances 20 9.5238 %

sort dataset

Given a list of between 0 and 4 numbers this dataset's target is the sorted version of that list.

FOIL

Generates a model with 4 clauses and a total of 13 terms:

- sort([],[]).
- sort(A,A) :- components(A,C,[]).
- sort(A,B) :- components(A,C,D), components(B,C,E), sort(D,E), components(E,F,G), lt(C,F).
- sort(A,B) :- components(A,C,D), components(B,E,F), sort(D,G), components(G,E,H), lt(E,C), components(I,C,H), sort(I,F).

Time 0.2 secs

JRIP

It is necessary to handle the four elements of the class (the 4-element list) as individual targets:

For C1 - Number of Rules : 5		
Incorrectly Classified Instances	0	0 %
For C2 - Number of Rules : 3		
Incorrectly Classified Instances	6	9.2308 %
For C3 - Number of Rules : 5		
Incorrectly Classified Instances	14	21.5385 %
For C4 - Number of Rules : 2		
Incorrectly Classified Instances	19	29.2308 %

qs44 dataset

This dataset also expresses the sort function. It differs from the previous dataset in FOIL's internal relations used to implement the sort:

FOIL:

Generates a model with 2 clauses and 6 terms:

- sort([],[]).
- sort(A,B) :- components(A,C,D), partition(C,D,E,F), sort(E,G), sort(F,H), components(I,C,H), append(G,I,B).

Time 0.3 secs

JRIP

<same as "sort" above>

diabetes dataset

The dataset consists of 768 individuals tested for diabetes. 268 are tested positive and 500 tested negative. Attributes are:

- A) number of times pregnant continuous
- B) plasma glucose level continuous
- C) blood pressure continuous
- D) skin fold thickness continuous
- E) 2 hour insulin continuous
- F) body mass index continuous This is (kg/(height)^2)
- G) diabetes pedigree function continuous
- H) age continuous

FOIL

The dataset was discretized using Weka with each attribute discretized into 8 bins. It was then converted into a FOIL.d file format. Three non-target relations were established:

- "lessThan(N,N)"
- "minus(N,N,N)"
- "reverse(N,N)"

FOIL was run allowing negative literals.

FOIL generates a model with 33 clauses and 116 terms (3.5 terms/clause).
There are 37/768 misclassified (an error rate of 5%).
Time 1.5 secs

JRIP

JRIP generates a model 3 rules

- (plas >= 124) and (plas >= 155) => class=tested_positive (122.0/24.0)
- (plas >= 124) and (mass >= 31) and (pedi_x10 >= 44) => class=tested_positive (58.0/18.0)
- => class=tested_negative (588.0/130.0)

Incorrectly Classified Instances 194 25.2604 %

spam dataset

The dataset consists of 4601 instances of e-mail tested for spam. 1813 are tested positive and 3788 tested negative. There are 57 attributes which characterize the e-mail message including word frequency, character frequency and capital run lengths.

FOIL

An initial attempt to run FOIL using the full dataset (57 attributes 4601 instances) made no forward progress.

In an attempt to speed things up the following steps were taken: Preprocessing included attribute discretizing (Weka Discretize) then attribute selection (Weka AttributeSelection/CfsSubsetEval). The resulting dataset had 15 attributes with a maximum number of 5 bins.

In the FOIL.d file - a single Type "N" was introduced which was discrete ordered with eight "bin" values (1..8) as constants. Two non-target relations were established "lessThan(N,N)", "minus(N,N,N)". FOIL was run (allowing negative literals) using the following command:

```
./foil6 < spambase13.d
```

The resulting definition had 35 clauses, with a total of 67 literals (2 literals/clause). There are 306/4601 incorrectly classified (an accuracy of 93%). It took 14.7 seconds to build the model.

JRIP

Ripper was run on the original spambase dataset with 57 attributes and 4601 instances. The model consisted of 17 rules.

Incorrectly Classified Instances	350	7.607 %

Summary

The table below summarizes the results. A distinction is made between datasets that were 'theoretical' (based on a mathematical foundation) and some were real-world. The former included ackermann, member, nxm, sort, qs44 and the latter included crx, spam, diabetes.

Dataset	"Theoretical" or "Real World"	FOIL Clauses	JRIP Rules
ackermann	theoretical	3	-
member	theoretical	2	4
nxm	theoretical	2	2
sort	theoretical	4	12
qs44	theoretical	2	12
crx	real world	19	5
diabetes	real world	33	3
spam	real world	35	17

On theoretical datasets FOIL produced models with fewer terms. For example for sort and qs44 FOIL produced models with 4 and 2 terms as compares to JRIP's model with 12. Besides being compact FOIL's models were also generalized and would work outside the bounds of the immediate examples. JRIP was able to work through these problems but the representations it produced were less crisp and not generalized.

On real-world datasets JRIP outperformed FOIL On crx, diabetes and spam JRIP produced a model which was significantly more compact than FOIL. In these cases it might be possible to improve FOIL's model by adding more functions.

"Sort" presented illustrated how FOIL was able to use a bounded world to generate a general rule. In this case both attribute and target class were variable-length lists. FOIL handled this length variability using an intermediate 'component' function. JRIP had no way to handle variable length attribute or target. For JRIP it was necessary to pad the list to its full length for all cases. Here FOIL clearly showed the advantage of first-order language through its use of an intermediate variable.