On Relational Learning for Information Extraction

Patricia Jiménez, José Luis Arjona, and J.L. Álvarez

Abstract. The extraction and integration of data from multiples sources are required in current companies which manage their business process by heterogeneous collaborating applications. However, integrating web applications is an arduous task because they are intended for human consumption and they do not provide APIs to access to their data automatically. Web Information extractors are used for this purpose but, they mostly provide ad-hoc highly domain dependent solutions. In this paper we aim at devising Information Extractors with a FOIL based core algorithm. It is a widely used first order rule learning algorithm since their rules are substantially more expressive and allow to learn complex concepts that cannot be represented in the attribute-value format. Furthermore, we focus on integrating other scoring functions to check if we can improve the rule search guide speeding up the learning process in order to make FOIL tractable in real-world domains such as Web sources.

1 Introduction

The World Wide Web has become one of the largest repository of knowledge and the immediate standard to publish information. However, this information is offered via Hypertext Markup Language (HTML), what makes its perception easier for humans but not appropriate for automatic processing, since web sources do not usually

Patricia Jiménez
University of Sevilla, Avda. Reina Mercedes, 41012, Sevilla
e-mail: patriciajimenez@us.es

J.L. Arjona
University of Huelva (La Rábida), Palos de la Frontera 21071, Huelva
e-mail: arjona@dti.uhu.es

J.L. Álvarez
University of Huelva, La Rábida, Palos de la Frontera 21071, Huelva
e-mail: alvarez@dti.uhu.es

J.M.C. Rodríguez et al. (Eds.): Trends in PAAMS, AISC 157, pp. 117–124.
springerlink.com

provide an Application Programmatic Interface (API) to interact with its interface automatically.

Web-Wrappers are the software component used for this purpose. They are intended to emulate the behaviour of a person who is interacting with a web user interface. It consists of an Enquirer, which maps user queries onto search forms, a Navigator, which executes filled search forms and reaches data web pages, an Information Extractor, which extract the information of interest from data web pages and return it stored as structured data for further processing. Finally, the Verifier attempts to find erroneous result sets. Our focus is on providing engineering support to devise Information Extractors.

Unfortunately, most Information Extractors today are inferred in a very ad-hoc way, in the sense of they can effectively extract information from a specific web site and achieve very good performance, but they may not be applied to other web sites with the same success. The ability to scale with the number and variety of information sources becomes the central challenge to information extraction (IE).

We wish to tackle this problem from inductive logic programming perspective. We aim at devising Information Extractors automatically with a FOIL [12, 13] based core algorithm. It is a widely used first order rule learning algorithm since their learnt rules are substantially more expressive, and allow the system to learn relational and recursive concepts that cannot be represented in the attribute-value format assumed by most machine learning algorithms. The application of relational learning can be decisive in domains that exhibit substantial variability such as Web pages.

Modified versions of FOIL are the basis for most adaptive IE systems that use relational learning techniques. For example, SRV system of Freitag [3] is one of the most successful ILP and top-down based learning system used for IE, which strongly follows the idea of the standard FOIL algorithm. The system is capable of learning extraction rules explaining single slots from natural and HTML documents. Moreover, Freitag extended SRV's feature predicates to make SRV able to exploit HTML structure by adding HTML-specific features. Although SRV almost always performs better than other learners, it does not solve all any new fields outright.

In order to deal with complex real-world domains for IE where the search space is not tractable, we work on devising an improved version of FOIL by applying some optimisations and heuristics. In this paper we present one of the optimisations consisting of replacing Information-based scoring function with other scoring functions coming from statistics, machine learning and data mining literature. We wish to find out through 16 ilp and classification tasks whether there is some scoring function that guides the search of the rules in a more efficient way, speeding up the learning process.

The paper is organised as follows: section 2, introduces an overview of FOIL algorithm. In section 3, we present some previous works on improving FOIL. Next, a common notation and a set of scoring functions are proposed. Then, we explain the experiments performed and we discuss the significance of our results. Finally, our future work is addressed in section 6.

2 FOIL Overview

Training data in FOIL comprises a target predicate, which is defined by a collection of positives and negatives examples according to whether they satisfy the target predicate or not, and a set of support predicates, which are defined extensionally, by a set of ground tuples.The goal is to learn a set of rules that explain the target predicate in terms of itself and the support predicates. The set of first order rules are represented as function-free Horn clauses and can optionally contain negated body literals.

It uses separate-and-conquer method rather than divide-and conquer, focusing on creating a single rule at a time and removing the positive examples covered by each learnt rule. Then, it is invoked again to learn a second rule based on the remaining training examples. It is called a sequential covering algorithm because it sequentially learns a set of rules that together cover the full set of positive examples. Additionally, FOIL employs a mechanism to speed up this process, pruning vast parts of the literal space when they show to be no better literals than the ones found so far.

In order to learn each rule, it follows a top-down approach, starting with a rule with an empty list of antecedents, and guided by a greedy search, the body of the rule is extended iteratively by adding the best new literals chosen according to a scoring function. This information-based scoring function, is designed to ensure that the learner will choose literals that include many positive examples and exclude many negative ones, while maintaining good overall coverage. Construction of a single rule stops if it matches only positive examples or reaches a predefined minimum accuracy. Furthermore, FOIL include MDL criterion [14] that stops the growth of the rule if the encoding length of the rule exceeds the number of bits needed for explicitly encoding the positive examples it covers. Thus, the induction of overly long and specific rules is prevented, especially in noisy-domains.

3 Related Work

Many authors have already tried to improve the performance of FOIL in several ways. Some earlier proposals are:

mFOIL [6] employs techniques from attribute-value learning to improve its noise handling capacities. It also offers two alternatives to the information-based scoring function, laplace-estimate and m-estimate. Moreover, FOIL's encoding length is replaced with criteria relying on statistical significance testing. Finally, it conducts beam-search to overcome, at least partially, some of the disadvantages of FOIL's greedy hill-climbing search. mFOIL is able to process intensionally defined background predicates and allows the user to define additional constraints to gain efficiency.

FOIDL [10] is also able to process intensionally defined background knowledge and negative examples are not needed. It assumes output completeness, i.e., the tuples in the relation show all valid outputs. Finally, it supports the induction of decision lists. That is an ordered set of rules each ending with a cut. When answering

a query, the decision list returns the answer of the first rule in the ordered set which succeeds in answering the query. Rules are in reverse order, being the most general rules, those that cover many positive examples, at the end of the decision list.

FOCL [11] develops a hybrid method that combines inductive learning and explanation-based components. The latter allows advantageously to accept as input a partial, possible incorrect rule as an approximation of the target predicate. The candidate rules are evaluated using FOIL's information-based scoring function.As mFOIL, it also relies on user-defined constraint to restrict literal search space.

FOSSIL [2], uses a statistical correlation-based scoring function. It can be used to deal with noise by cutting off all literals that have a scoring function value below a certain threshold. They demonstrated that this threshold was independent of the number of training examples and of the amount of noise in the data. Moreover, they provide a new stopping criterion independent of the number of training examples and dependent on this statistical correlation scoring function.

The system nFOIL [8] integrates the nave Bayes learning scheme with FOIL. Two main changes on FOIL are required: first, examples that are already covered have still to be considered when learning additional rules; second, scoring functions is based on class conditional likelihood rather than information-based. nFOIL was shown to perform better than FOIL and to be competitive with more sophisticated approaches.

FZFOIL [4] uses interest-based measures to compute the score of a literal to overcome some lacks of Information scoring function and increase accuracy of the learnt rules. FZFOIL also manage fuzzy knowledge background predicates, where tuples are associated with these predicates with a certain degree of agreement. Induced rules are represented in both, ordinary and fuzzy logic format, and might generate incomplete and/or inconsistent rules. It process intensionally defined background predicates as well.

4 Our Contribution

We have implemented in Java an algorithm fairly similar to the last version of FOIL (FOILv6.4) which, unlike FOIL, it also supports defining background predicates intensionally, in the well-known Prolog-rules representation. It is possible through the JPL library that provides an interface between Java and Swi-Prolog. Some improvements need still to be incorporated in order to supply a more closer version of FOILv6.4.

This is a first attempt to check the behaviour of different scoring functions mainly taken from [15] and [7]. According to [4] we think that replacing information-based scoring function could improve the efficiency of the learning process at guiding the search, while retaining expressiveness.

The proposed scoring functions have been adapted according to the notation of the well-known confusion matrix, as appear in table 1. In any confusion matrix, tp (true positives) denotes the number of positive examples and fp (false positives) denotes the number of negative examples satisfied by a candidate literal. Similarly,

fn (false negatives) and tn (true negatives) denote the number of positive and negative examples respectively, excluded by this literal. Finally, N is the total number of examples in the current training set.

Table 1 List of Scoring functions

Scoring Function	**Formula**	**Scoring Function**	**Formula**
Information (I)	$-log\frac{tp}{tp+fp}$	Laplace Accuracy (Lap)	$\frac{tp+1}{tp+fp+2}$
Leverage (Lev)	$\frac{tp\cdot tn-fn\cdot fp}{N^2}$	ϕ-coefficient (ϕ)	$\frac{tp\cdot tn-fn\cdot fp}{\sqrt{(tp+fn)\cdot(tp+fp)\cdot(fp+tn)\cdot(fn+tn)}}$
Confidence (Conf)	$\frac{tp}{tp+fp}$	Satisfaction (Sat)	$\frac{tp\cdot tn-fp\cdot fn}{(tp+fp)\cdot(tn+fp)}$
F-measure (F_1)	$2\cdot\frac{tp}{2\cdot tp+fn+fp}$	kappa (κ)	$\frac{2\cdot(tp\cdot tn-fp\cdot fn)}{N^2-(tp+fn)\cdot(tp+fp)-(fp+tn)\cdot(tn+fn)}$
Odds-ratio (OR)	$\frac{tp\cdot tn}{fp\cdot fn}$	Yule's Q (Q)	$\frac{tp\cdot tn-fp\cdot fn}{tp\cdot tn+fp\cdot fn}$
Lift (L)	$\frac{N\cdot tp}{(tp+fp)\cdot(tp+fn)}$	Jaccard(ζ)	$\frac{tp}{tp+fp+fn}$
Collective Strength (CS)	$\frac{tp+tn}{(tp+fp)\cdot(tp+fn)+(fn+tn)\cdot(fp+tn)}\times\frac{N^2-(tp+fp)\cdot(tp+fn)-(fn+tn)\cdot(fp+tn)}{N-tp-tn}$		

5 Experiments

To carry out our analysis, we have performed 16 experiments taken from ILP, machine learning and classification problems literature. Each learning task involved a limited amount of background information, just that required for the task at hand, and training examples noise-free.

Amongst the trials carried out, we can highlight kindship from Hinton [5], arch task, introduced by Winston [16], Michalski East-West trains problem [9], play tennis and contact lenses classification problems taken from [17] and several Ivan Bratko's tasks on a universe of three-length lists taken from well-known text Prolog Programming for Artificial Intelligence [1].

Table 2 Results

Tests		I	Lap	Lev	ϕ	Conf	Sat	F_1	κ	OR	Q	Lift	J	CS
member	R	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
75+/120	T	3.20	3.74	3.20	3.85	3.59	3.23	3.62	3.45	3.48	20.84	3.45	3.46	3.60
	C	8.90	8.90	8.09	8.90	8.90	8.90	8.90	8.90	8.90	8.90	8.90	8.90	8.90
del	R	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	time	1.00	1.00	0.56
81+/4800	T	137.2	124.1	184.4	220.1	125.8	117.9	135.8	130.6	172.6	-	119.8	132.4	185.4
	C	14.57	14.57	24.43	14.57	14.57	14.57	14.57	14.57	14.57	-	14.57	14.57	19.82
last	R	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.020	1.00	1.00	1.00
39+/120	T	5.64	5.55	5.38	5.71	5.67	5.48	5.40	5.37	16.58	11.06	17.63	5.35	3.76
	C	9.17	9.17	9.17	9.17	9.17	9.17	9.17	9.17	10.49	9.75	10.49	9.17	8.17
insert	R	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	time	1.00	1.00	1.00
81+/4800	T	130.7	118.3	170.5	195.6	117.9	117.3	129.2	128.6	166.4	-	120.7	130.3	84.13
	C	14.19	14.19	23.96	14.19	14.19	14.19	14.19	14.19	14.19	-	14.19	14.19	14.40
sublist	R	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
202+/1600	T	30.64	24.84	30.28	59.03	29.52	29.83	63.48	24.32	48.34	95.16	23.51	24.82	83.98
	C	12.71	12.71	12.71	12.71	12.71	12.71	14.75	12.71	12.71	14.75	12.71	12.71	26.33
even	R	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
10+/40	T	0.83	0.76	0.72	0.78	0.73	0.81	0.83	0.80	1.72	1.48	1.46	0.76	0.73
	C	5.00	5.00	5.00	5.00	5.00	5.00	5.00	5.00	5.00	17.36	17.36	5.00	5.00
permutation	R	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
52+/256	T	40.58	36.96	7.86	23.18	34.06	254.5	5.87	7.85	22.04	193.6	250.7	5.82	9.19
	C	17.97	17.97	22.44	29.68	17.97	49.69	25.72	22.44	29.53	26.38	68.28	25.72	22.72
playtennis	R	1.00	1.00	0.57	0.79	1.00	0.43	0.50	0.57	0.43	0.43	0.14	0.50	0.71
14+/72	T	7.22	7.18	13.30	13.84	8.61	3.71	7.12	5.52	5.65	4.35	1.84	6.80	11.82
	C	172.0	170.4	107.2	137.8	170.2	50.19	70.00	66.17	52.28	51.82	18.40	70.00	118.1
lenses	R	1.00	1.00	0.50	1.00	1.00	1.00	0.58	0.50	0.50	0.50	0.58	0.58	1.00
24+/72	T	3.68	3.74	1.61	6.72	4.02	4.88	3.04	1.56	1.58	1.37	2.30	2.98	11.10
	C	102.7	102.7	8.76	111.5	102.7	101.4	24.01	8.76	8.76	8.76	24.79	24.01	112.3
plus	R	1.00	1.00	-	1.00	1.00	1.00	-	1.00	1.00	-	1.00	-	1.00
6+/27	T	1.95	2.01	-	6.11	2.14	2.00	-	2.11	4.91	-	7.02	-	8.02
	C	18.06	18.06	-	17.06	18.06	18.06	-	18.06	19.47	-	22.76	-	15.66
mult	R	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
48+/1056	T	43.38	34.07	77.63	111.6	35.54	33.04	41.39	39.37	65.24	221.8	36.66	40.05	24.04
	C	29.10	29.10	29.58	29.10	29.10	29.10	29.10	29.10	29.10	29.77	29.10	29.10	16.49
animals	R	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.29	0.29	1.00	1.00	1.00
17+/68	T	4.02	3.66	4.35	7.41	4.00	4.29	5.91	4.41	2.39	2.23	4.26	5.88	3.09
	C	18.81	18.81	18.81	21.81	18.81	18.81	18.81	18.81	7.00	8.17	18.81	18.81	21.42
network	R	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	-	1.00	1.00	1.00
19+/81	T	0.86	1.39	0.98	1.01	1.59	1.17	5.49	1.65	0.83	-	1.39	5.71	0.81
	C	12.34	12.34	12.34	13.92	12.34	12.34	42.05	12.34	12.34	-	12.34	42.05	12.34
kindship	R	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
12+/576	T	6.21	5.85	6.54	9.61	5.76	5.82	6.13	6.55	8.16	8.22	6.04	6.30	6.36
	C	9.49	9.49	9.49	9.49	9.49	9.49	9.49	9.49	9.49	9.49	9.49	9.49	26.68
trains	R	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
5+/10	T	0.72	0.64	0.72	0.76	0.72	0.72	0.73	0.75	0.73	0.83	0.80	0.76	0.72
	C	8.82	8.82	8.82	8.82	8.82	8.82	8.82	8.82	8.82	8.92	8.92	8.82	8.82
arch	R	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	-	1.00	1.00	1.00
2+/1728	T	6.60	5.69	6.83	8.44	6.29	6.18	5.83	6.07	9.38	-	9.08	6.19	8.33
	C	19.97	19.97	19.97	19.97	19.97	19.97	19.97	19.97	19.97	-	19.97	19.97	19.97

The adopted measures to help us to compare the results are[1]:

1. ***Recall (R)***. It determines if a set of rules is complete, i.e., if it satisfies all positive examples belonging to the target predicate.
2. ***Complexity (C)*** of the induced set of rules. It is computed in terms of bits from Minimum Description Length Principle [14].
3. ***Time (T)*** employed for rules learning process measured in seconds and restricted to 1000s.

The results in table 2 exhibit that in the 81,25% of cases there is some scoring function that performs better than the information-based one, in terms of time and complexity, maintaining the full recall. Information-based, Laplace, Confidence and Satisfaction scoring functions seem to be the most promising scoring functions. Nevertheless, they do not always provide the best results. In a few cases, Leverage and Collective Strength obtained best times inducing the same or similar rules that the ones induced applying information-based scoring function. Inducing rules in shorter time may be due to both, a more effective alpha-beta pruning and search guide, which depend completely on the scoring function selected.

Note that Yule's Q scoring function is the worst employed. It is not able to induce a set of rules or it wastes a lot of time to induce them. Neither OR scoring function provided results to take into account, only in one case behaved not much more speedy than applying information-based scoring function. We also should know that Collective Strength and ϕ-coefficient do not have implemented the alpha beta pruning yet, because it requires significant changes in the implementation. But even competing at a disadvantage, the former gets to be the most promising one in a 18,75% of cases. Unfortunately, the latter produced no significant results. It exceeded the time achieved by information based-scoring function in the 93,75% of cases. Finally, Lift and Jaccard scoring functions also got better results than those obtained with information-based scoring function in a 31,75% and 43,75% of cases respectively. However, they never were the most promising ones in any task.

Clearly, the results stated that most of these scoring functions are not recommendable for classification tasks which are the playTennis and lenses tests. Only Information Gain, Laplace and Confidence reached a full recall. In other tasks as plus, insert or del, there were some functions that could not induced the set of rules by time constraints or simply because they didn't find the rules.

6 Conclusions

We have implemented a customised version of the well-known FOIL algorithm and we have proposed to integrate different scoring functions taken from the literature, in order to guide the search for a rule efficiently.

Thirteen scoring functions were applied over 16 tests from ILP and classification domain in order to compare them. Although many of these scoring functions performed reasonable well, the experiment highlights the strong dependency between the task and the scoring function applied, since they got good results in some tasks

[1] Note that we do not allow to cover negative examples so, rules are 100%accurate.

and not so well in others. For this reason, we think of applying some algorithm to rank them as in [15] and decide the scoring function that best fits for a specific task.

As future work, we plan include new adapted scoring functions for extending the possibilities. Next steps are addressed to provide new optimisations and heuristics that make our approach tractable in real-world applications related to IE.

Acknowledgements. This research is supported by the European Commission (FEDER), the Spanish and the Andalusian R&D&I programmes (grants TIN2007-64119, P07-TIC-2602, P08-TIC-4100, and TIN2008-04718-E).

References

1. Bratko, I.: Prolog Programming for Artificial Intelligence. In: McGettrick, A.D., Van Leeuwen, J. (eds.). Addison-Wesley (1986)
2. Fürnkranz, J.: FOSSIL: A Robust Relational Learner. In: Proc. of the Eur. Conf. on Mach. Learn. (1994), doi:10.1007/3-540-57868-4_54
3. Freitag, D.: Information Extraction from HTML: Application of a General Machine Learning Approach. In: Proc. Fifteenth Natl. Conf. on Artif. Intell., pp. 517–523 (1998)
4. Gomez, A.J., Fernandez, G.: Induccion de definiciones logicas a partir de relaciones: mejoras en los heuristicos del sistema FOIL. In: Congr. Nac. Program. Declar., pp. 292–302 (1992)
5. Hinton, G.E.: Learning distributed representations of concepts. In: Proc. of the Eighth Annu. Conf. of the Cogn. Sci. Soc., pp. 1–12 (1986)
6. Lavrac, N., Dzeroski, S.: Inductive Logic Programming: Techniques and Applications. In: Lavrac, N., Dzeroski, S. (eds.) Inductive Logic Programming, pp. 173–179. Hellis Horwood, New York (1994)
7. Lavrac, N., Flach, P.A., Zupan, B.: Rule Evaluation Measures: A Unifying View. In: Proc. of the 9th Int. Workshop on Inductive Log. Program. (1999), doi:10.1007/3-540-48751-4_17
8. Landwehr, N., Kersting, K., De Raedt, L.: nFOIL: Integrating Naïve Bayes and FOIL. In: The 20th Natl. Conf. on Artif. Intell., pp. 795–800 (2005)
9. Michalski, R.S.: Pattern recognition as rule-guided inductive inference. IEEE Trans. on Pattern Analysis and Mach. Intell. 2, 349–361 (1980)
10. Muggleton, S.: Inverse Entailment and Progol. New Gener. Comput. J. (1995), doi:10.1007/BF03037227
11. Pazzani, M.J., Kibler, D.F.: The Utility of Knowledge in Inductive Learning. Mach. Learn. 9, 57–94 (1992)
12. Quinlan, J.R., Cameron-Jones, R.M.: FOIL: A Midterm Report. In: Proc. of the Eur. Conf. on Mach. Learn. (1993), doi:10.1007/3-540-56602-3_124
13. Quinlan, J.R., Cameron-Jones, R.M.: Induction of Logic Programs: FOIL and Related Systems. New Gener. Comput. J. 13, 287–312 (1995)
14. Rissanen, J.: Universal coding, information, prediction, and estimation. IEEE Trans. Inf. Theory 30, 629–636 (1984)
15. Tan, P., Kumar, V., Srivastava, J.: Selecting the right objective measure for association analysis. Inf. Syst. (2004), doi:10.1016/S0306-4379(03)00072-3
16. Winston, P.H.: Learning Structural Descriptions from Examples. In: Winston, P.H. (ed.) The Psychology of Computer Vision, pp. 157–209. McGraw-Hill, New York (1975)
17. Witten, I.H., Frank, E., Hall, M.A.: Data Mining: Practical machine learning tools and techniques with Java implementations, pp. 9–13. Morgan Kauffman (2000)