Relational Information Gain

Marco Lippi[1], Manfred Jaeger[2], Paolo Frasconi[1], and Andrea Passerini[3]

[1] Dipartimento di Sistemi e Informatica, Università degli Studi di Firenze, Italy
[2] Department for Computer Science, Aalborg University, Denmark
[3] Dipartimento di Ingegneria e Scienza dell'Informazione, Universit degli Studi di Trento, Italy.

Abstract. We introduce relational information gain, a refinement scoring function measuring the informativeness of newly introduced variables. The gain can be interpreted as a conditional entropy in a well-defined sense and can be efficiently approximately computed. In conjunction with simple greedy general-to-specific search algorithms such as FOIL, it yields an efficient and competitive algorithm in terms of predictive accuracy and compactness of the learned theory.

1 Introduction

Many ILP or relational learning systems build discriminative models by a stepwise refinement of logical-relational features. For example, in general-to-specific rule learners like FOIL [12], features are the bodies of Horn clauses that are constructed by adding one literal at a time. In models that adopt a decision-tree style design (in a wide sense), like TILDE [2], Multi-Relational Decision Trees [9], Relational Probability Trees [10], or Type Extension Trees (TETs) [6], features are represented by branches in the tree structure, which are constructed in an iterative top-down process.

A distinguishing characteristic of incremental feature construction in relational learning is the possibility to refine a current feature for a given set of entities X by introducing new entities Y and their attributes via relations $r(X, Y)$ (assumed to be binary for notational simplicity). For example in FOIL this is done by adding a literal $r(X, Y)$ to the body of the clause.

The search for the best feature refinement is typically directed by some scoring function that evaluates its usefulness for discriminating the class label of X. A refinement that does not introduce any new entities can be scored in a relatively straightforward manner using standard information gain metrics. A refinement introducing new entities is more difficult to evaluate, however: standard metrics can measure the *direct informativeness* of such a refinement, i.e. the direct improvement in the feature's discriminative power. However, it is widely recognized that the main benefit of introducing Y is not always its direct informativeness, but the possibility it opens up to construct in further refinement steps informative features for X by imposing suitable conditions on Y. Two main approaches have been used to take into account this *potential informativeness* of a literal introducing new entities: *determinate literals* [11] are literals where for each X there exists exactly one Y with $r(X, Y)$. Determinate literals are not directly informative, but their inclusion in the clause is computationally inexpensive, which is why e.g. FOIL adds all possible determinate literals to a clause in order to exploit their potential informativeness. A second approach consists of *lookahead* techniques [1, 4], where for the scoring of the literal $r(X, Y)$ already further possible refinement steps using Y are considered. Both determinate literals and lookahead have severe limitations: the former represent only a very special kind of potentially informative literals, and the latter is subject to a combinatorial search space explosion when performing lookahead over multiple refinement steps (which is why, in practice, lookahead is constrained to certain user-defined refinement patterns).

The goal of this paper is to develop a notion of *relational information gain (rig)* for scoring candidate literals introducing new variables, such that both direct and potential informativeness can be measured. Specifically, we have the following desiderata for rig:

1. rig captures a sound and general information theoretic concept of reduction in conditional entropy of the class label distribution. It thereby is widely applicable, and not a specialized heuristic scoring function for a specific model or search strategy.

2. rig increases as a function of *direct informativeness* of literal $r(X, Y)$, defined as the information about the target relation associated with the existence of X, Y such that $r(X, Y)$ (or, more generally, with the number of such pairs).

3. rig increases as a function of *potential informativeness* of literal $r(X, Y)$, defined as the maximum information that can be gained about the target relation thanks to the introduction of Y via $r(X, Y)$ and further refinements using Y (without lookahead, only based on the immediate relational properties of r).

In the following sections we develop a rig score that is motivated by these desiderata.

2 Data: relational and pseudo-iid

Since information gain is a statistical concept based on a probabilistic data model, we first investigate what kind of statistical model of relational data is appropriate to support the definition of rig. We assume that the data consists of a single relational or logical database containing constants $c_1, \ldots, c_n$, boolean attributes *class*, $a_1, \ldots, a_k$, and binary relations $r_1, \ldots, r_l$ (it is only for ease of exposition that we restrict ourselves to boolean attributes, a unary class predicate, and only binary relations). Thus, the data can also be seen as observations of the boolean random variables *class*(c_1), *class*$(c_2), \ldots$, $a_1(c_1), \ldots$, $r_l(c_n, c_n)$ (i.e. the ground atoms of the Herbrand base). However, none of these random variables are assumed to be independently sampled. In general, the database is a single draw from a joint distribution over ground atoms (one could also add a distribution from which the domain itself is sampled, but this adds little in our current context).

A problem we now encounter is that this data model does not really allow us to say very much about entropies and information gain. To begin with, we would have to be able to estimate the entropy of the class label distribution. However, *class*$(c_1), \ldots$, *class*(c_n), in this model are single realizations of non-independent random variables. Even if we assume that the *class*(c_n) at least are identically distributed (i.e. assuming that the generating distribution makes no distinctions among the different objects in the domain), we cannot use the observed empirical frequencies of *pos* and *neg* labels to estimate properties of the class label distribution, including its entropy.

It seems that in order to leverage certain types of statistical analysis tools, one actually has to compromise the holistic relational data model, and extract from the overall relational structure a number of separate sub-structures, which are then treated as iid. Such a transformation of relational data into a collection of *pseudo-iid* data fragments is performed in several relational learning systems. For example the *local training sets* employed by FOIL can be seen in this way; the learning routines in the *Proximity* system (`http://kdl.cs.umass.edu/software`) operate on collections of sub-graphs extracted from the underlying database.

We will assume that computations of conditional entropy are based on pseudo-iid data views where the cases are labeled (tuples of) domain elements $\boldsymbol{c} = (c_{i_1}, \ldots, c_{i_m})$, together with boolean attributes representing the internal relational structure of $\boldsymbol{c}$, as well as the relational connections between $\boldsymbol{c}$ and other entities $c_j \notin \boldsymbol{c}$.

3 Relational Information Gain

For notational simplicity, suppose we are evaluating a literal $r(X, Y)$ where X can be bound to an object to be classified. In general we would allow labeled tuples of objects (e.g. as constructed as the local training sets in FOIL), and literals that contain (one or several) variables bound to some components of these tuples, as well as new variables. $r(X, Y)$ is potentially informative because it may be possible to find further literals imposing conditions on Y, s.t. the conjunction of these literals allows us to discriminate positive and negative X. Consider two extreme cases: first, assume that $r(X, Y)$ is true if and only if $Y = c_n$. Then $r(X, Y)$ is determinate, but does not enable any discrimination between positive and negative examples, i.e. it is not potentially informative. Now consider $r(X, Y)$ encoding a bijection between the c_i. Again, $r(X, Y)$ is determinate, but now it also is potentially informative, because if we succeed to find an attribute a_h that characterizes the set $B = \{c_i \mid \exists c_j : \ \mathit{class}(c_j) = \mathit{pos} \wedge r(c_j, c_i)\}$, then the conjunction $r(X, Y), a_h(Y)$ will allow for a perfect classification of X.

When literals are not ideal deterministic literals as above, then there usually will not exist such a clear-cut definition of the set B of objects associated via $r(X, Y)$ with the positive class. However, potential informativeness means that there exists some set B, such that there is a correlation between $\mathit{class}(X)$ and membership in B of the Y with $r(X, Y)$. This leads us to define for a subset $B \subseteq \{c_1, \ldots, c_n\}$ and literal $r(X, Y)$ the following integer-valued attribute, which can be added as a new column to a pseudo-iid data view:

$$F_{r(X,Y),B}(c_i) := \mid \{c_j \in B \mid r(c_i, c_j)\} \mid . \tag{1}$$

For each B, (1) defines an attribute that has a standard information gain

$$\mathit{ig}(F_{r(X,Y),B}) := H(\mathit{class}(X)) - H(\mathit{class}(X) \mid F_{r(X,Y),B}). \tag{2}$$

When evaluating $r(X, Y)$, however, no specific set B is given. We therefore define the relational information gain of $r(X, Y)$ as

$$\mathit{rig}(r(X, Y)) = \max_{B \subseteq \{c_1, \ldots, c_n\}} \mathit{ig}(F_{r(X,Y),B}). \tag{3}$$

Thus, we are taking an optimistic attitude towards evaluating potential informativeness (or rather, stressing the "potential") by basing the definition on the most discriminating subset B, even though we do not know whether we will be able to characterize this optimal B using the available attributes and relations. While the definition of *rig* is mainly motivated by potential informativeness, it also captures direct informativeness, which is simply measured by $F_{r(X,Y),B}$ for $B = \{c_1, \ldots, c_n\}$. The exact maximization in (3) is presumably computationally intractable. In our experiments we therefore use a greedy approximation algorithm, that constructs B by testing for $k = 1, \ldots, n$ whether $ig(F_{r,B\cup\{c_k\}}) > ig(F_{r,B})$, and adding c_k to B if this is the case .

4 Experiments

We tested rig in conjunction with FOIL's search scheme on both synthetic and real data: when a new variable is introduced, the refinement is evaluated by rig, while in other cases FOIL's traditional weighted information gain (wig) is used. Since the two scores take on incomparable values, we implemented a simple randomized algorithm that chooses between variable-introduction (if available) and non-variable-introduction refinements with probability 0.5 and subsequently uses rig or wig, respectively, to choose the best refinement in its class. Unless otherwise stated, for all the experiments both FOIL and rig-FOIL threshold accuracy for clause selection was set to 50%, and we repeated 20 runs of rig-FOIL. To choose among the 20 randomly generated theories, we simply used training set accuracy (in all our experiments we observed nearly perfect correlation between train and test set accuracy). We tested rig-FOIL on one artificial and seven relational data sets and compared against standard wig-FOIL [12] and Aleph [14].

4.1 Slotchain data

We use synthetic *slotchain* data (cf. [6]) to test rig's ability to identify potentially informative literals. In this data, an example X is *positive*, if and only if an entity Z with $att(Z) = true$ can be reached via the chain of relations $r_{0,0}, r_{1,0}, r_{2,0}, r_{3,0}$. Thus, the target clause to find in this data is

$$positive(X) \leftarrow r_{0,0}(X, Y_1), r_{1,0}(Y_1, Y_2), r_{2,0}(Y_2, Y_3), r_{3,0}(Y_3, Z), att(Z). \quad (4)$$

The $r_{i,0}$-literals are neither directly informative nor determinate. The data set consists of approximately 5,400 true ground facts and also includes "noise relations" $r_{i,j}$ $(i = 0, \ldots, 3, j = 0, \ldots, 2)$ that have no predictive value. Using the above strategy, rig-FOIL is able to recover the target clause 4 from data (about one quarter of the randomly generated theories contained the target clause and, as expected, these always outperformed on the training set the remaining theories). On the same data, both standard FOIL and Aleph failed to retrieve the target clause.

4.2 Real data

The *UW-CSE* data set [13] consists of 3,380 tuples and 12 predicates describing the Department of Computer Science and Engineering at the University of Washington.

Table 1. Results on UW-CSE and KDD 2001 data sets: Precision (P), recall (R), F-measure (F_1), and number of clauses (N).

	FOIL				rig-FOIL				Aleph			
	P	R	F_1	N	P	R	F_1	N	P	R	F_1	N
UW-CSE	44.9	27.4	34.1	21	61.7	38.5	47.4	20	48.2	23.9	32.0	36
Localization	73.9	70.1	72.0	63	78.0	73.6	75.7	70	80.9	82.8	81.8	68
Function	64.2	59.8	61.9	34	73.1	65.0	68.8	27	63.8	81.2	71.4	76

Table 2. Results on the Alzheimer data sets: Accuracy (A) and number of clauses (N).

	FOIL		rig-FOIL		Aleph	
	A	N	A	N	A	N
Amine	84.1	26	84.3	26	80.4	39
Toxic	92.2	16	94.2	18	82.8	30
Acetyl	77.6	49	79.7	33	78.1	44
Memory	70.8	26	70.9	24	69.7	42

The task is to predict the binary relation $\texttt{advisedBy}(\texttt{x}, \texttt{y})$ using in turn four scientific areas for training and one for testing as in [13]. In this case, the threshold accuracy for clause selection for rig- and wig-FOIL was set to 30%. Results in Table 1 correspond to micro-averaged precision, recall, and F_1 measure. The KDD Cup 2001 data set 2 [5] consist of 862 (training) and 381 (test) anonymized genes for which several relational features are given, including protein-protein interactions. Two classification tasks are defined: *localization* (predict whether a gene/protein is located in the nucleus of the cell) and *function* (predict whether the gene/protein function is related to cell growth). For these data sets, 10-fold cross-validation results are summarized in Table 1. Finally, in the Alzheimer data sets [8], the goal is to predict the relative biochemical activity of variants of the Tacrine drug. Four binary classification tasks are defined as in [8]: *amine*, *toxic*, *acetyl*, and *memory*. For each task, the target predicate $\texttt{positive}(\texttt{a}, \texttt{b})$ is interpreted as "drug a is preferred to drug b" and examples form a partially ordered set. Negative literals were disallowed in the experiments on these data sets (this prevents anti-symmetry to be learned) and molecule pairs were randomly split into 10 folds for cross-validation accuracy estimation.

As shown in Tables 1 and 2, rig-FOIL generally produces more accurate and more compact theories than standard FOIL and performs comparably or better than Aleph. A paired two-sided t-test, ($p < 0.05$) revealed that rig-FOIL significantly outperform FOIL on *function*, *toxic*, and *acetyl* and Aleph on *amine* and *toxic*. The Alzheimer data sets have been recently used in [7] to benchmark some structure learning algorithms for Markov logic networks (MLN) [13] and, in particular, a novel algorithm based L_1-regularization weight learning on a structure defined by a very large set of Aleph-generated clauses. We therefore extracted the learned clauses and used them as the structure for MLN, whose parameters were subsequently learned discriminatively using the voting perceptron approach implemented in the Alchemy system. Following [7], the transitivity relation on the target predicate positive was incorporated as background knowledge in these experiments (note that FOIL cannot learn transitivity because of its limitations in handling recursive predicates [3]). Accuracies on *amine*, *toxic*, *acetyl*, and *memory* are 89.2, 92.8, 93.7, and 87.9, respectively and are comparable to the best results reported in [7] (90.5, 91.9, 87.6, 81.3, respectively, but using a different split for 10-fold cross-validation).

5 Discussion and conclusions

Relational information gain is a refinement scoring function that has a sound information-theoretic justification. Our artificial data experiments clearly show the ability of rig in discovering potential informativeness of a literal, without requiring a lookahead. In real data sets, rig in conjunction with FOIL greedy search is able to construct accurate and compact theories. Rig, however, is not tied to FOIL, and might also be used to drive feature refinements in other types of logical-relational models, especially also those with a decision tree architecture.

References

1. H. Blockeel and L. De Raedt. Lookahead and discretization in ILP. In *Proc. of the 7th Int. Workshop on ILP*, pages 77–84, 1997.
2. H. Blockeel and L. De Raedt. Top-down induction of first-order logical decision trees. *Artificial Intelligence*, 101(1–2):285–297, 1998.
3. RM Cameron-Jones and JR Quinlan. Avoiding pitfalls when learning recursive theories. In *Proc. of the 13th Int. Joint Conf. on Artificial Intelligence*, 1993.
4. L. P. Castillo and S. Wrobel. A comparative study on methods for reducing myopia of hill-climbing search in multirelational learning. In *Proc. of the 21st Int. Conf. on Machine Learning*, 2004.
5. J. Cheng, C. Hatzis, H. Hayashi, M.A. Krogel, S. Morishita, D. Page, and J. Sese. KDD Cup 2001 report. *SIGKDD Explor. Newsl.*, 3(2):47–64, 2002.
6. P. Frasconi, M. Jaeger, and A. Passerini. Feature discovery with type extenstion trees. In *Proc. of the 18th Int. Conf. on Inductive Logic Programming*, pages 122–139, 2008.
7. T.N. Huynh and R.J. Mooney. Discriminative structure and parameter learning for markov logic networks. In *Proc. of the 25th Int. Conf. on Machine Learning*, 2008.
8. R. D. King, A. Srinivasan, and M. J, E Sternberg. Relating Chemical Activity to Structure: an Examination of ILP Successes. *New Generation Computing*, 13(2,4):411–433, 1995.
9. A. J. Knobbe, A. Siebes, and D. van der Wallen. Multi-relational decision tree induction. In *Proceedings of PKDD-99*, pages 378–383, 1999.
10. J. Neville, D. Jensen, L. Friedland, and M. Hay. Learning relational probability trees. In *Proceedings of SIGKDDD'03*, 2003.
11. J.R. Quinlan. Determinate literals in inductive logic programming. In J. Mylopoulos and R. Reiter, editors, *Proc. of the 12th Int. Joint Conf. on Artificial Intelligence*, 1991.
12. JR Quinlan and RM Cameron-Jones. FOIL: A midterm report. In *European Conference on Machine Learning*, page 3. Springer, 1993.
13. Matthew Richardson and Pedro Domingos. Markov logic networks. *Machine Learning*, 62(1–2):107–136, 2006.
14. A Srinivasan. The aleph manual. `http://web.comlab.ox.ac.uk/oucl/research/areas/machlearn/Aleph/`, 2001.