ESANN'1999 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 21-23 April 1999, D-Facto public., ISBN 2-600049-9-X, pp. 231-236

From first order logic to $\mathbb{N}^d$: A data driven reformulation

Michèle Sebag

LMS, Ecole Polytechnique, 91128 Palaiseau
& LRI, Université Paris-XI, 91405 Orsay
Michele.Sebag@polytechnique.fr

Abstract. First order logic (FOL) offers a natural way of modeling domains such as chemistry: a molecule is most adequately described as a graph of atoms linked by simple or double bonds. To overcome the specific difficulties of dealing with FOL, this paper presents an automatic mapping from the initial problem domain onto the set of integer vectors $\mathbb{N}^d$, where d is a user-supplied integer. This mapping onto a metric space induces a (semi)-distance on the problem domain. Within supervised learning, the quality of the reformulation can thus be estimated from the predictive accuracy of a k-nearest neighbor classifier based on this distance. The approach is validated on a real-world problem pertaining to organic chemistry: toxicology prediction.

1. Introduction

Extending or reformulating the initial representation of the problem domain has been paid much attention in Machine Learning [8], and even more so since the inception of Support Vector Machines (SVM) [13]. To the best of our knowledge, SVMs mostly deal with attribute-value representations (e.g. examples are represented as vectors of continuous or discrete values) [10].

This paper is concerned with reformulating first-order logic domains, also called relational domains. The approach is illustrated in the field of organic chemistry: the *mutagenesis* problem consists of learning toxicology predictors from examples of carcinogenic and non-carcinogenic molecules [7]. Though a molecule can be described according to a set of attributes (e.g. hydrophobicity of the molecule, energy of given orbits, and so forth), the success of learning is conditioned to and limited by the quality of these attributes.
A molecule is best described through a set of atoms, and a set of bonds relating these atoms, e.g. as a graph:

$$atom(a_1, carb), ..., atom(a_{40}, hydr), bond(a_1, a_2, simple), ...bond(a_{25}, a_{32}, double)$$

However, learning and reasoning within graphs is intractable in the general case (see section 2). This paper focuses on mapping the initial problem domain onto a more manageable space, the set of integer vectors $\mathbb{N}^d$, where d is a user-

ESANN'1999 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 21-23 April 1999, D-Facto public., ISBN 2-600049-9-X, pp. 231-236

supplied integer. This mapping is data-driven and does not require any prior knowledge (e.g. such as a generative model of the data [6]).

This mapping is based on a set of particular patterns: let $D(E,F)$ denote the set of all patterns (e.g. sub-graphs) appearing in a training example E and not appearing in example F. This pattern could be thought of as the logical analog of the set of hyperplanes separating E from F. From $D(E,F)$, we derive a function mapping the problem domain onto the integer set $\mathbb{N}$. By randomly selecting d pairs of examples in the training set, we thus map the problem domain onto $\mathbb{N}^d$.
Two possibilities are considered in the paper: *DISTILL*, designed for supervised learning, extracts pairs of examples belonging to distinct target concepts; *UNDISTILL*, designed for unsupervised learning, simply extracts pairs of distinct examples. In both cases, the user controls the granularity of the reformulation and its computational cost through parameter d.

This paper is organized as follows. Section 2 briefly describes some work related to relational learning. Section 3 studies the reformulation of a problem domain using a set of patterns, discusses its limitations and the properties of the mapping. Section 4 describes the particular reformulation of the problem domain based on the $D(E,F)$, and details an experimental validation of the approach.

2. Dealing with relational representations

Learning from relational examples, also known as *Inductive Logic Programming* (ILP) [9] suffers from an additional difficulty compared to learning from attribute-value examples. Comparing two attribute-value examples is linear in the number of attributes, while comparing two graphs is exponential in the size of the graphs in the general case. For this reason, relational learners employ syntactic or search biases to restrict the exploration to short patterns (e.g. 3-atom patterns) [12]. A way of overcoming this drawback is to deal with distance matrices, representing an example through its distance to other examples [4].

However, defining distances on relational domain is far from being straightforward: in essence, the better a distance captures the whole structure of the examples, the worse its complexity [1, 3]. Specific distances have been designed for particular structures of problem domains, e.g. edit distances on sequences or tree-structured data. Such distances involve a set of fixed operators together with their cost; the distance of any two examples is estimated from the best-cost sequence of operators transforming one example into the other one [2].

Another possibility is to refine and prune the search space of graph matching using domain knowledge, e.g. in the field of shape retrieval [5]. Yet another very elegant approach subsumes the problem of the initial representation of the problem domain, assuming a generative model of the domain. A bridge to SVM is offered by defining the kernel functions associated to each example, from the model maximizing the likelihood of the example [6].

ESANN'1999 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 21-23 April 1999, D-Facto public., ISBN 2-600049-9-X, pp. 231-236

3. Pattern-driven Reformulation

Let us examine how a set of patterns can be used to reformulate a problem domain Ω. Let us consider first boolean functions h_i defined on Ω, termed conjunctive patterns. A set of d conjunctive patterns $h_1, \ldots h_d$ induces a mapping π from Ω onto $\{0,1\}^d$, associating to any example E in Ω the vector $\pi(E) = ((h_1(E), \ldots, h_d(E))$.

Such a mapping π naturally induces a semi-distance d_π on Ω^2, $d_\pi(E, E')$ being defined as the Hamming distance of $\pi(E)$ and $\pi(E')$. One sees that d_π is symmetrical, satisfies the triangular inequality, but fails to satisfy the identity relation ($d_\pi(E, E') = 0 \not\Rightarrow E = E'$) as π is not necessarily injective.

The quality of such pattern-based reformulations increases with the shattering power of the h_i, defined as the average number of patterns discriminating two any examples[1]. However, π usually does not preserve the structure of the problem domain. Typically, most patterns h_i considered in machine learning would cover (take value *true* for) less than half the problem domain; equivalently, $h_i(E)$ is more often set to 0 than to 1. Still, conjunctive pattern-based reformulation maps the problem domain onto a boolean space, where 0 and 1 play equivalent roles.

We therefore consider more complex patterns. Let g_i be a disjunction of boolean functions $h_{i,j}$ defined on the problem domain Ω, termed disjunctive patterns. Function g_i induces a mapping from Ω onto $\mathbb{N}$, associating to any example E the number $g_i(E)$ of conjunctive patterns $h_{i,j}$ in g_i that are satisfied (take value *true*) for E. A set of d disjunctive patterns $g_1, \ldots g_d$ thereby induces a mapping π from Ω onto the set of integer vectors $\mathbb{N}^d$. A definition of a distance d_π naturally follows, taken as the Euclidean distance between $\pi(E)$ and $\pi(E')$.

Note that π preserves to some extent the logical nature of the initial problem domain. Let g_i^M denote the *M-of-N* hypothesis constructed from the disjunctive $g + i$: E satisfies g_i^M iff E satisfies at least M among the $h_{i,j}$. The sequence $g_i^1, .. g_i^M, g_i^{M+1}, ..$ is a sequence of boolean functions of increasing specificity. These can be considered as a sequence of embedded balls, with $g_i^\bullet$ denoting the smallest non empty ball. For any example E, the i-th coordinate of $\pi(E)$ reflects how close is E to satisfy $g_i^\bullet$: the numerical order in $\mathbb{N}$ encodes some logical structure of the initial problem domain.

4. Automatic Relational Reformulation

This section is devoted to constructing disjunctive patterns from relational examples, and using them to automatically transform the problem domain. The quality of the reformulation is evaluated by the predictive accuracy of a k-nearest-neighbor (kNN) classifier based on the induced distance.

[1] Still, the pattern-based distance does not involve in any way the conclusions eventually associated to h_i; this suggests that the relevance of the reformulation is potentially independent from the discriminant power of the h_i. This conjecture will be experimentally confirmed in (see section 4.).

ESANN'1999 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 21-23 April 1999, D-Facto public., ISBN 2-600049-9-X, pp. 231-236

Consider, for any pair (E, F) of training examples, the set $D(E, F)$ of patterns discriminating those examples. In attribute-value representation, $D(E, F)$ simply is the disjunction of maximally discriminant selectors [8][2]:

E	*[elem = carbon],*	*[type = 22],*	*[charge = - 3.45],*	*[benzen=yes]*
F	*[elem = oxygen],*	*[type = 17],*	*[charge = .22]*	

Table 1: *D(E,F) = [elem = carbon] ∨ [type > 17] ∨ [charge < .22]*

Consider now relational examples, each example being described as a set of nodes and functions of tuples of these nodes. Let σ denote a mapping from the nodes in example E onto nodes in example F; σ defines an alignment of E and F, i.e. make them comparable in an attribute-value language. Note $D(E, F, \sigma)$ the disjunctive pattern discriminating E from F according to mapping σ (Table 2). Then, $D(E, F)$ can be characterized as the conjunction of all $D(E, F, \sigma)$ for σ ranging over the set Σ_F of all mappings from nodes in E onto nodes in F [11].

E:	e_1, e_2;	$[elem(e_1) = carb], [elem(e_2) = hydr], [bond(e_1, e_2) = true]$
F:	f_1, f_2, f_3;	$[elem(f_1) = oxy], [elem(f_2) = carb], [elem(f_3) = hydr]$
		$[bond(f_1, f_3) = true],\ [bond(f_2, f_3) = false]$
σ_1:	$\sigma_1(e_1) = f_1; \sigma_1(e_2) = f_3$	$D(E, F, \sigma_1) = \{e_1;\ [elem(e_1) = carb]\}$
σ_2:	$\sigma_1(e_1) = f_2; \sigma_1(e_2) = f_3$	$D(E, F, \sigma_2) = \{e_1, e_2;\ [bond(e_1, e_2) = true]\}$

Table 2: *Constructing $D(E, F)$ from relational E and F*

The complexity of $D(E, F)$ is linear in the size of E and in the size of Σ_F, but the latter is exponential in the size of F. We thus construct a stochastic (Monte-Carlo) approximation of $D(E, F)$, defined as the conjunction of $D(E, F, \sigma_i)$, over L mappings σ_i uniformly drawn in Σ_F. The user controls both the accuracy of the approximation and the computational cost, through the number L of samples considered during the stochastic approximation.
For any example G, the integer function $D(E, F)(G)$ (how many conjunctive patterns in $D(E, F)$ does G satisfy), which is similarly exponential in the size of G, is again approximated using a Monte-Carlo procedure based on K samplings of the set Σ_G of mappings of nodes in E onto nodes in G [11].

A data-driven reformulation strategy is then achieved as follows:
• Select d pairs of examples E_i, F_i,
• Construct (an approximation of) the patterns $D(E_i, F_i)$ separating them,
• For any G, map G onto $\mathbb{N}^d$ $(G \mapsto (D(E_i, F_i)(G)))$, the i-th coordinate of G in this numerical representation reflecting how close G is from E_i compared to F_i.

DISTILL (for *Distance Induction with Stochastic Inductive Learning*) implements this strategy in a supervised learning context, by uniformly selecting

[2]We restrict ourselves to selectors $[att = V]$, where V denotes a discrete value or a numerical interval depending on whether *att* is a nominal or numerical attribute.

ESANN'1999 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 21-23 April 1999, D-Facto public., ISBN 2-600049-9-X, pp. 231-236

examples E_i and F such as they belong to distinct target concepts. *UNDISTILL*, for *Unsupervised DISTILL*, uniformly selects pairs of examples $E_i \neq F_i$.

The approach is evaluated on the well-studied mutagenesis problem [7]. Among the reference results are those of *PROGOL*, obtaining $88\% \pm 2\%$ correct prediction [12] and *STILL*, which takes the vote of all $D(E,F)$ [11], with $93\% \pm 4\%$ correct prediction for a good parameter setting. The runtimes respectively are 40,500 seconds for *PROGOL* and 120 seconds for *STILL* (HP-735).

DISTILL is evaluated from the average predictive accuracy of the 1-NN classifier based on d_π. Stochastic parameters are set to the same default values as for *STILL* ($L = 300$, $K = 3$), and the same protocol validation is used (average results on 25 independent runs, one run testing the rules learned from 90% of the data onto the remaining 10%)..

One goal of experiment is to see how the quality of reformulation depends on the number d of patterns considered, and d is varied in 10..100. We also study how the approach is suited to unsupervised learning, by comparing the results of *DISTILL* and *UNDISTILL*.

Tables 3.a and 3.b respectively give the results obtained by *DISTILL* and *UNDISTILL* (with run times in seconds on a HP-710).

(a)) *DISTILL*

d	Accuracy	Time
10	88.6 ± 4.8	7
30	93.6 ± 5	19
50	94.7 ± 3.7	31
70	**96.7** ± 4.3	43
90	95.3 ± 2.4	56

(b) *UNDISTILL*

d	Accuracy	Time
10	86.7 ± 6.9	6
30	94.2 ± 3.8	19
50	93.3 ± 3.8	31
70	93.3 ± 5.3	44
90	**94.7** ± 2.6	56

Table 3*: Predictive accuracy on the 188-compound problem*

Surprisingly, the performances of *DISTILL* and *UNDISTILL* are not significantly different. In the frame of supervised learning, this demonstrates the robustness of pattern-based reformulation with respect to noise. Furthermore, our example-driven reformulation shows adapted to unsupervised learning as well.

5. Discussion and Perspectives

Rather than syntactically comparing two examples, we propose to compare the way these respectively behave with respect to a set of patterns. We used a particular set of patterns $D(E,F)$, which can be viewed as the logical analog of the hyperplanes separating any two training examples E and F. Each such pattern defines a new integer attribute on the problem domain. *DISTILL* (resp. *UNDISTILL*) constructs a set of d such attributes by randomly selecting pairs of examples E_i, F_i that belong to different class (resp. are different). Experimental validation shows that both *DISTILL* and *UNDISTILL* obtain

ESANN'1999 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 21-23 April 1999, D-Facto public., ISBN 2-600049-9-X, pp. 231-236

outstanding results compared to the state of the art on the mutagenesis dataset, even for low values of d ($d \geq 30$).

This work presents some odd particularities compared to the spirit of Support Vector Machines [10]. For one thing, we use patterns defined from a pair of training examples, whereas kernel functions are defined from a single training example. Second, we use a poor (uniform) selection of the pairs of examples, while SVM is primarily concerned with an optimal selection of the support vectors. Further work will apply standard SVM on the attribute-value reformulation of the problem domain. One interesting question is whether and how the support vectors in the reformulated representation relates to the pairs of examples leading to the "optimal" reformulation.

References

[1] G. Bisson. Learning in First Order Logic with a similarity measure. In *Proceedings of* 10^{th} *AAAI*, 1992.

[2] U. Bohnebeck, T. Horváth, and S. Wrobel. Term comparisons in first order similarity measures. In *Proceedings of ILP-98*, pages 65–79. Springer Verlag, LNAI 1446, 1998.

[3] W. Emde and D. Wettscherek. Relational instance based learning. In L. Saitta, editor, *Proceedings of the* 13^{th} *International Conference on Machine Learning*, pages 122–130. Morgan Kaufmann, 1996.

[4] T. Hofmann and J.M. Buhmann. Pairwise data clustering by deterministic annealing. *IEEE PAMI*, 19(1):1–25, 1997.

[5] B. Huet, A. Cross, and E. Hancock. Graph matching for shape retrieval. In M.J. Kearns, S.A. Solla, and D.A. Cohn, editors, *Proc. NIPS-11*. MIT Press, 1999.

[6] T. Jaakkola and D. Haussler. Exploiting generative models in discriminative classifiers. In M.J. Kearns, S.A. Solla, and D.A. Cohn, editors, *Proc. NIPS-11*. MIT Press, 1999.

[7] R.D. King, A. Srinivasan, and M.J.E. Sternberg. Relating chemical activity to structure: an examination of ILP successes. *New Gen. Comput.*, 13, 1995.

[8] R.S. Michalski. A theory and methodology of inductive learning. In R.S Michalski, J.G. Carbonell, and T.M. Mitchell, editors, *Machine Learning: an artificial intelligence approach*, volume 1, pages 83–134. Morgan Kaufmann, 1983.

[9] S. Muggleton and L. De Raedt. Inductive logic programming: Theory and methods. *Journal of Logic Programming*, 19:629–679, 1994.

[10] B. Schölkopf, C. Burges, and A. Smola. *Advances in Kernal Methods*. MIP Press, 1998.

[11] M. Sebag and C. Rouveirol. Tractable induction and classification in FOL. In *Proceedings of IJCAI-97*, pages 888–892. Morgan Kaufmann, 1997.

[12] A. Srinivasan, S.H. Muggleton, M.J.E. Sternberg, and R.D. King. Theories for mutagenicity: a study in first order and feature-based induction. *Artificial Intelligence*, 85:277–299, 1996.

[13] V. N. Vapnik. *The Nature of Statistical Learning*. Springer Verlag, 1995.