Online Closure-Based Learning of Relational Theories

Frédéric Koriche

LIRMM, UMR 5506, Université Montpellier II CNRS
161, rue Ada 34392 Montpellier Cedex 5, France
koriche@lirmm.fr

Abstract. Online learning algorithms such as Winnow have received much attention in Machine Learning. Their performance degrades only logarithmically with the input dimension, making them useful in large spaces such as relational theories. However, online first-order learners are intrinsically limited by a computational barrier: even in the finite, function-free case, the number of possible features grows exponentially with the number of first-order atoms generated from the vocabulary. To circumvent this issue, we exploit the paradigm of closure-based learning which allows the learner to focus on the features that lie in the closure space generated from the examples which have lead to a mistake. Based on this idea, we develop an online algorithm for learning theories formed by disjunctions of existentially quantified conjunctions of atoms. In this setting, we show that the number of mistakes depends only logarithmically on the number of features. Furthermore, the computational cost is essentially bounded by the size of the closure lattice.

1 Introduction

A recurrent theme in machine learning is the development of efficient *online* learning algorithms, capable of producing better and better predictions in an incremental way [4]. Such algorithms are "anytime learners" that can be interrupted at each instant to provide a prediction whose correctness is related to the number of mistakes that have been made so far. The underlying model takes place in a sequence of trials. At any stage, the learner is first presented a new example, next it is asked to predict its associated class, and then it is told whether its prediction was correct or not. In case of mistake, an update procedure is activated and the current hypothesis is refined accordingly.

In a landmark paper [16], Littlestone introduced an elegant algorithm for learning k out of n variable disjunctions which he called Winnow. It resembles the perceptron algorithm in its simplicity, but employs multiplicative, rather than additive, weight updates on input variables. Consequently, the number of mistakes grows essentially as $k \log n$ instead of kn. The fact that the dependence on n is reduced to logarithmic, rather than linear, makes this algorithm potentially applicable even if the number of variables is enormous. For example, the SNoW algorithm, a variant of Winnow, has been shown to be effective in natural language settings with ten of thousands of features [10].

This remarkable property has lead researchers to examine the possibility of applying multiplicative update algorithms to large concept classes where the number of patterns is exponential in the input dimension. In this setting, the key question is: just how can we preserve attribute-efficiency in order to learn, in a reasonable amount of time and space, a function of k relevant features in presence of a possibly exponential number $N - k$ of irrelevant features ?

Computational learning theory has supplied mixed results. On the one hand, it has been shown that several geometrical classes are indeed attribute-efficient learnable, using appropriate data structures [11, 17]. The basic idea is to exploit commonalities among features, partitioning them into a polynomial number of equivalence classes that are used for prediction. The number of mistakes still depends only logarithmically on the number of patterns and the computational cost remains essentially polynomial on the input dimension. On the other hand, for logical theories such as monotone DNF formulas, Khardon et al. [14] have recently shown that, unless $P = \#P$, there is no polynomial time algorithm capable of simulating Winnow over exponentially many conjunctive features.

Such a computational barrier does *not* necessarily imply that a brutal force implementation of Winnow is the sole option to obtain complete correctness. In fact, even if the resulting partition is not always guaranteed to be polynomial, the idea of "compiling" a large space can be more efficient than systematically exploring the set of N features. Furthermore, Blum [3] observed that, in many situations the problem at hand exhibits a three-stage hierarchy: a small number of relevant features in the target function, a larger number of features that appear in each example, and an enormous number of possible features. In such circumstances, the combined strategies of "focusing" on a limited fragment of the space and "compiling" this fragment into a compact data structure seem to provide a useful approach to circumvent the counting problem.

Following this research avenue, we investigate the paradigm of *closure-based* learning which allows a learner to focus on the closure space generated by the closure of the examples which have lead to a mistake. Based on a well-known property of closure operators, the data structure maintained by the learner is a complete lattice of features. During each trial, the learner first receives an unlabeled example, next predicts its class according to its lattice, and then receives the correct label. In case of mistake, the lattice is refined by taking the closure of the data structure with the current observation.

This paradigm is applied to the problem of learning relational theories formed by disjunctions of existentially quantified conjunctions of atoms. This class of formulas have the same expressive power as select-project-join-union database queries, which are the queries that occur most often in practice [1]. Furthermore, relational theories provide a substrate for many ILP systems that operate in a concept learning framework [18]. Namely, any existentially quantified conjunction of atoms can be regarded as a decision rule predicting the target concept. If any of the conjunctions in some theory "fires" for a given example, then the example is classified as positive. If none of them fires, the example is classified as negative.

In the relational setting, each candidate "feature" is an existentially quantified conjunction of atoms. Consequently, the number of possible features is exponential in the number of first-order atoms. The central aim of closure-based learning is to alleviate this combinatorial barrier by allowing the learner to limit exploration in the space of first-order conjunctions. Based on this paradigm, we develop an online algorithm that extends Winnow to relational theories. We show that the number of mistakes still depends only logarithmically on the number of possible features. Furthermore, the computational cost is polynomial in the size of the closure lattice. In the worst case, this structure can be exponential in the number of its maximal elements. Yet, experiments in formal concept analysis reveal that this case rarely occurs in practice; on average, the size of closure lattices increases polynomially with the number of atoms [5, 9]. These encouraging results corroborate the practical applicability of our approach.

Outline. Section 2 introduces the necessary background about online relational learning. Section 3 presents an algebraic setting for closure-based induction. Section 4 is devoted to the development and the analysis of the closure-based Winnow algorithm. Notably, a mistake bound and a computational bound for this algorithm are reported in this section. Finally, section 5 compares the present approach with other results in online relational learning, and concludes with some perspectives of further research.

2 Preliminaries

In this section, we begin to introduce a logical setting for relational theories and next, we present the "standard" Winnow algorithm applied to relational theories. We conclude this section by bringing to the fore the main computational bottleneck of online relational learning.

2.1 Relational Logic

The linguistic component of this study is an existential positive fragment of first-order logic defined from a finite and pre-fixed vocabulary. Function symbols including constants, are not allowed. The vocabulary consists in a finite set of *predicate symbols* $\{\mathsf{p}_1, \cdots, \mathsf{p}_p\}$ and a finite set of *variables* $\{\mathsf{x}_1, \cdots, \mathsf{x}_k\}$. Each predicate symbol has a finite arity, which is the number of its arguments. We consider that the maximum arity over all predicate symbols is bounded by a constant a. Such an assumption is commonly advocated in the relational learning literature [12, 22]. An *atom* $\mathsf{p}(\mathsf{x}_1, \cdots, \mathsf{x}_t)$ is a t-ary predicate symbol followed by a bracketed t-tuple of variables. The set of all distinct atoms generated from the vocabulary is denoted $\mathbf{A}$. Using the above notations, we remark that the cardinality of $\mathbf{A}$ is upper bounded by pk^a, which is polynomial in the number of predicate symbols and the number of variables.

A *relational conjunction* (henceforth called *feature*) is a closed formula in prenex normal form, containing only existential quantifiers, and whose matrix is a conjunction of atoms. A *relational theory* (or *theory*) is a disjunction of relational conjunctions. For convenience, we shall sometimes represent theories as sets of features and features as sets of atoms. The size of a feature F, denoted $|F|$, is the number of all atoms occurring in it. Note that the restriction on the number of variables does not limit the size of features to be constant. Indeed, long conjunctions of size $O(pk^a)$ can be constructed since variables can appear in more than one atom. The space of all features constructed from the vocabulary is denoted $\mathbf{F}$. The cardinality of this space is denoted N. Notably, we observe that N is upper bounded by 2^{pk^a}.

Example 1. Our running example is a variant of the so-called Bongard problem (see e.g. [13]). In this problem, the learner is presented some scenes involving objects and geometrical relationships among them. The underlying task is to distinguish positive scenes from negative ones. We consider here the vocabulary composed by the unary predicate symbols circle, square and $\mathsf{triangle}$, the binary predicate symbols left, in and larger, and the variables x_1 and x_2. The theory T below involves three relational conjunctions.

$$\exists \mathsf{x}_1 \exists \mathsf{x}_2 (\mathsf{circle}(\mathsf{x}_1) \wedge \mathsf{square}(\mathsf{x}_2) \wedge \mathsf{in}(\mathsf{x}_1, \mathsf{x}_2)),$$
$$\exists \mathsf{x}_1 \exists \mathsf{x}_2 (\mathsf{circle}(\mathsf{x}_1) \wedge \mathsf{square}(\mathsf{x}_2) \wedge \mathsf{larger}(\mathsf{x}_2, \mathsf{x}_1)),$$
$$\exists \mathsf{x}_1 \exists \mathsf{x}_2 (\mathsf{circle}(\mathsf{x}_1) \wedge \mathsf{circle}(\mathsf{x}_2) \wedge \mathsf{in}(\mathsf{x}_1, \mathsf{x}_2))$$

Examples are interpretations that involve objects and relationships among them. A *domain* is a finite set of objects. A *ground atom* over a domain D is an expression $\mathsf{p}(o_1, \cdots, o_t)$, where p is a t-ary predicate symbol and $o_1, \ldots, o_t$ are objects in the domain D. An *interpretation* is a pair $I = (D^I, P^I)$ where D^I is a domain and P^I is a set of ground atoms over D^I. An interpretation I is a *model* of a relational conjunction F if there is a substitution θ mapping variables in the feature F to objects in D^I and such that $A\theta \in P^I$ for each atom A in F. By extension, an interpretation I is a *model* of a relational theory T if there is a relational conjunction F in T such that I is a model of F.

Example 2. Consider the following interpretation I involving three objects. We can observe that I is a model of the theory T examined in example 1. Indeed, we notice that I is a model of the first two conjunctions described in T.

$$I = (\{1, 2, 3\}, \{\mathsf{circle}(1), \mathsf{circle}(2), \mathsf{square}(3), \mathsf{in}(1, 3), \mathsf{larger}(1, 3)\})$$

Given an interpretation I, the *feature space* of I, denoted $\mathbf{F}(I)$, is the set of all features F in $\mathbf{F}$ such that I is a model of F. An element F of $\mathbf{F}(I)$ is called a *maximal feature* if there is no proper superset F' of F in $\mathbf{F}(I)$. The set of all maximal features of I is called the *basis* of I and denoted $B(I)$. The following property states that the problem of checking whether I is a model of some feature F can be reduced to a covering test of F in the basis of I.

Proposition 1. *Let I be an interpretation and F a relational conjunction. Then I is a model of F if and only if there is a feature F' in $B(I)$ such that $F \subseteq F'$.*

Proof. First, suppose that I is a model of F. Then F is an element of $\mathbf{F}(I)$ and hence, F is covered by at least one maximal feature in $B(I)$. Now, suppose that I is a model of a maximal feature F' in $B(I)$ such that $F \subseteq F'$. Then, there is a substitution θ mapping variables in F' to objects in D^I and such that $F'\theta \subseteq P^I$. It follows that $F\theta \subseteq P^I$ and hence, I is a model of F. □

Interestingly, we remark that the cardinality of the basis of I is bounded by d^k, which is the number of possible substitutions over D^I. The basis of I can be found time quadratic in d^k. Namely, for each substitution θ over D^I, we first generate the saturated feature F formed by all atoms A in the language such that $A\theta \in I$. Next, we check whether a proper superset of F is present in the current basis of I. If this is not the case, we add F to the basis and we eliminate from it all proper subsets of F.

Example 3. The basis of the interpretation I specified in example 2 is given by the four following features.

$$\exists \mathsf{x}_1 \exists \mathsf{x}_2 (\mathsf{circle}(\mathsf{x}_1) \wedge \mathsf{circle}(\mathsf{x}_2))$$
$$\exists \mathsf{x}_1 \exists \mathsf{x}_2 (\mathsf{square}(\mathsf{x}_1) \wedge \mathsf{square}(\mathsf{x}_2))$$
$$\exists \mathsf{x}_1 \exists \mathsf{x}_2 (\mathsf{circle}(\mathsf{x}_1) \wedge \mathsf{square}(\mathsf{x}_2) \wedge \mathsf{in}(\mathsf{x}_1, \mathsf{x}_2) \wedge \mathsf{larger}(\mathsf{x}_1, \mathsf{x}_2))$$
$$\exists \mathsf{x}_1 \exists \mathsf{x}_2 (\mathsf{circle}(\mathsf{x}_2) \wedge \mathsf{square}(\mathsf{x}_1) \wedge \mathsf{in}(\mathsf{x}_2, \mathsf{x}_1) \wedge \mathsf{larger}(\mathsf{x}_2, \mathsf{x}_1))$$

2.2 Online Relational Learning

The online learning model can be regarded as a game between two players, the learner and the environment. A target relational theory T^* containing r features, is fixed by the environment and hidden from the learner. During each trial, the learner first receives an interpretation from the environment, next it makes a prediction based on its current hypothesis and then the learner receives the correct response. In the setting of online relational learning, the quantities that the learner would like to minimize are the number of mistakes it makes and the computational resources it spends along the process. Notice that learner is merely *passive* and cannot ask membership queries or statistical queries.

Before presenting the algorithm, we need additional definitions. Given a feature F, the *classifier* of F is a map that assigns to each interpretation I a boolean value given by: $F(I) = 1$ if I is a model of F, and $F(I) = 0$ otherwise. Similarly, given a theory T, the classifier of T is a map that assigns to each interpretation I the value $T(I) = 1$ if I is a model of T, and the value $T(I) = 0$ otherwise. A *linear threshold function of* $\mathbf{F}$ is a function Φ that associates to each feature F in $\mathbf{F}$ a weight in $\mathbb{R}^+$. Intuitively, $\Phi(F)$ captures the degree of relevance of the feature F in the learning process. The classifier of Φ is a map that assigns to each interpretation I a boolean value defined as follows:

$$\Phi(I) = \begin{cases} 1 \text{ if } \left(\sum_{F \in \mathbf{F}} \Phi(F) \cdot F(I)\right) \geq N, \text{ and} \\ 0 \text{ otherwise} \end{cases}$$

Initialization

0 Set $\Phi(F) \leftarrow 2$ for each relational conjunction $F \in \mathbf{F}$

Trials

1 Receive an interpretation I

2 If $\left(\sum_{F \in \mathbf{F}} \Phi(F) \cdot F(I)\right) \geq N$ then

predict $\Phi(I) \leftarrow 1$

else

predict $\Phi(I) \leftarrow 0$

3 Receive $T^*(I)$. If $T^*(I) \neq \Phi(I)$ then for each F such that $F(I) = 1$ do

Demotion: if $\Phi(I) = 1$ then set $\Phi(F) \leftarrow \frac{1}{2}\,\Phi(F)$

Promotion: if $\Phi(I) = 0$ then set $\Phi(F) \leftarrow 2\,\Phi(F)$

Fig 1: Standard Relational Winnow

We have now all notions in hand to present the standard Winnow algorithm. The key idea is to maintain a linear threshold function that approximates the target theory. The algorithm is presented in figure 1. Initially, $\Phi(F) = 2$ for each feature in $\mathbf{F}$. On each received interpretation I, if $\Phi(I)$ predicts the correct class of I then no change is made. If $\Phi(I) = 1$ and I is a negative example, then a *demotion* occurs: the weights of each feature involved in the prediction are divided by 2. Dually, if $\Phi(I) = 0$ and I is a positive example, then a *promotion* occurs: the weights of each feature that predicted correctly are multiplied by 2. By an adaptation of Littlestone's analysis, the number of mistakes made by the learner depends on N only logarithmically and on r polynomially.

Although "feature-efficient", the standard Winnow algorithm is confronted with an important computational barrier. Namely, an explicit representation of a linear threshold function of $\mathbf{F}$ takes $\Omega(2^{pk^a})$ size. The complexity issue is exacerbated still further by the fact that for any received interpretation I, a covering test must be done for each candidate feature F in the space $\mathbf{F}$. This test can be performed by enumeration in $O(|F|d^k)$ time, where d is the number of objects in the domain D^I. A similar result is obtained if the test is performed by computing the basis of I. Based on these considerations, the prediction step takes $O(d^k 2^{pk^a})$ time. Consequently, even for constant values of a and k, a brutal force implementation of relational Winnow is clearly infeasible.

Example 4. Let us consider the vocabulary presented in example 1. Given 2 variables, 3 unary predicate symbols and 3 binary predicate symbols, the number of atoms is 18. If 64 bits are needed to encode each weight, then an explicit representation of a linear relational threshold function would require 2^{24} bits. For 3 and 4 variables, we would need 2^{44} bits and 2^{66} bits. The last requirement is well beyond the capacity of computational machinery into the foreseeable future.

3 Closure-Based Induction

As observed in the previous section, the main computational bottleneck of online relational learning lies in the cardinality of the feature space. To alleviate this barrier, we advocate the paradigm of closure-based induction that allows the learner to "focus" on a limited portion of its feature space and to "compile" this portion into a semantically equivalent data structure. In this section, we introduce a formal setting for closure-based induction. We begin to examine the notion of relational closure space, next we define a projection operator over closure spaces, and then we concentrate on linear functions of closure spaces.

3.1 Relational Closure Spaces

Let T be a relational theory. Then we say that T is *closed* if for any nonempty subset S of T, the feature $\bigcap S$ is an element of T. Furthermore, we say that T is a *closure space* if T is closed and contains the maximal feature $\mathbf{A}$. Interestingly, we remark that any relational closure space is a Moore family of subsets of $\mathbf{A}$. Consequently, by an application of a well-known theorem about Moore families of subsets (see e.g. [2, 8]), any relational closure space forms a complete lattice under set-inclusion.

Given a relational theory T, the *feature space* of T, denoted $\mathbf{F}(T)$, is the set of all features F in $\mathbf{F}$ such that F is included in some element F' of T. We can see that if T is a closure space, then its feature space covers all elements in $\mathbf{F}$. Now, given a feature F in $\mathbf{F}(T)$, the *closure of F with respect to T*, denoted $C_T(F)$, is the feature formed by the intersection of all supersets of F in T:

$$C_T(F) = \bigcap \{F' \in T : F \subseteq F'\}$$

The *closure of T*, denoted $C(T)$ is given by the set $\{C_T(F) : F \in \mathbf{F}(T)\}$. The following property states that the "closure" of a relational theory is necessarily "closed" under intersection.

Proposition 2. *Let T be a relational theory. Then T is closed iff $T = C(T)$.*

Proof. Let $T' = C(T)$ and $T'' = \{\bigcap S : S \subseteq T\}$. We must show that $T' = T''$. Let F be an element of T'. By construction of T', there exists a feature F' in $\mathbf{F}$ such that $F = C_T(F')$. Let S be the set of all supersets of F' in T. Since $C_T(F') = \bigcap S$, it follows that $F = \bigcap S$. Therefore, $F \in T''$. Now, let F be an element of T'' and V be the set of all supersets of F in T. By construction of T'', there exists a subset S of T such that $F = \bigcap S$. Since $S \subseteq V$ and $S \neq \varnothing$ it follows that $\bigcap V \subseteq \bigcap S$. Hence, $C_T(F) \subseteq F$. Furthermore, for every element F' in V, we have $F \subseteq F'$. It follows that $F \subseteq \bigcap V$. Thus $F \subseteq C_T(F)$. By combining the two results, we obtain $F = C_T(F)$, and hence $F \in T'$. □

Given two closed relational theories T and T', the *intersection product* of T and T', denoted $T \circ T'$, is defined by the set $\{F \cap F' : F \in T \text{ and } F' \in T\}$. The intersection product provides a natural operator for constructing composite closed theories from basic building blocks. The following proposition states that the intersection product of two closed theories is necessarily a closed theory.

Proposition 3. *Let T and T' be two closed theories. Then $T \circ T'$ is closed.*

Proof. Let T'' denote $T \circ T'$. By proposition 2, T'' is closed if and only if for every nonempty subset S of T'', the feature $\bigcap S$ is an element of T''. Since the relational vocabulary is finite, we consider without loss of generality that $S = \{F''_1, \cdots, F''_n\}$. By construction, $F''_i = F_i \cap F'_i$ for some F_i in T and F'_i in T'. It follows that $\bigcap S = (\bigcap_{i=1}^{n} F_i) \cap (\bigcap_{i=1}^{n} F'_i)$. Since T and T' are closed, then the feature $\bigcap_{i=1}^{n} F_i$ is an element of T and the feature $\bigcap_{i=1}^{n} F'_i$ is an element of T'. Therefore $\bigcap S$ is an element of T''. □

The *congruence relation* of a theory T, denoted $\sim_T$, is the binary relation on $\mathbf{F}(T)$ defined by following condition: $F \sim_T F'$ if and only if $C_T(F) = C_T(F')$. Based on the axioms of equality, $\sim_T$ is an equivalence relation on $\mathbf{F}(T)$. The *congruence class* of a feature F with respect to T, denoted $[F]_T$, is the set of all features F' in $\mathbf{F}(T)$ such that $F \sim_T F'$. In the following, the cardinality of $[F]_T$ is denoted $\|F\|_T$. The following property states that congruence relations can be refined using the product operation.

Proposition 4. *Let T and T' be two closed theories. Then $\sim_{T \circ T'} = \sim_T \cap \sim_{T'}$.*

Proof. Let T'' be $T \circ T'$ and F be a feature in $\mathbf{F}(T'')$. We must prove that $C_{T''}(F) = C_T(F) \cap C_{T'}(F)$. Let S'' be the set of all supersets of F in T''. By construction, there exists a subset S of T and a subset S' of T' such that $\bigcap S'' = \bigcap S \cap \bigcap S'$. Let us show that $C_T(F) = \bigcap S$. Let V be the set of all supersets of F in T. Obviously, $S \subseteq V$. Let G be an element of V. We know that $F \subseteq G$. Furthermore, $F \subseteq F'$ for at least one element F' in S'. Therefore, $F \subseteq G \cap F'$ and hence, G must be an element of S. It follows that $V \subseteq S$. Therefore, $S = V$ and hence, $C_T(F) = \bigcap S$. Based on an analogue strategy, we can show that $C_{T'}(F) = \bigcap S'$. Since $C_{T''}(F) = \bigcap S''$, the result follows. □

We conclude this part by an important topological property of the closure operation. The following result states that the closure of a theory generates a complete partitioning of its feature space; the number of equivalence classes is determined by the size of the closure of the theory.

Proposition 5. *Let T be a relational theory. Then the congruence relation of T induces a complete partitioning of $\mathbf{F}(T)$ into $|C(T)|$ congruence classes.*

Proof. We know that the relation $\sim_T$ is an equivalence relation on the space $\mathbf{F}(T)$. Therefore, $\sim_T$ induce a complete partitioning of $\mathbf{F}(T)$. Now, let $T' = C(T)$ and $T'' = \{[F]_T : F \in \mathbf{F}\}$. We must show that $|T'| = |T''|$. Let f be the function that maps to each feature F in T' the congruence class $f(F) = [F]_T$ in T''. Let F and F' be two distinct elements of T'. Since $C_T(F) \neq C_T(F')$ it follows that $f(F) \neq f(F')$. Thus, f is injective and hence, $|T'| \leq |T''|$. Dually, let g be a function that associates to each class $[F]_T$ of T'' the feature $g([F]_T)$ in T' such that $g([F]_T) = C_T(F)$. Let $[F]_T$ and $[F']_T$ be two distinct congruence classes of T''. Since $C_T(F) \neq C_T(F')$, it follows that $g([F]_T) \neq g([F']_T)$. Thus g is injective and hence, $|T''| \leq |T'|$. □

3.2 The Projection Operation

The key idea of closure-based induction is to enable the learner to focus on limited regions of its feature space and to compile these regions into compact structures. This idea is captured by a projection operator that takes as input a closure space maintained by the learner and an interpretation sent by the environment, and that returns as output a closed theory which partitions the feature space of the interpretation into a set of congruence classes.

Let T be a closure space and I be an interpretation. Then, the *projection* of T onto I, denoted $P(T, I)$, is given by the intersection product of T and the closure of $B(I)$. In formal terms: $P(T, I) = T \circ C(B(I))$. The *update* of T by I, denoted $U(T, I)$, is given by the set $T \cup P(T, I)$. The next property states that the theories generated from projection and update are closed.

Proposition 6. *Let T be a closure space and I be an interpretation. Then $P(T, I)$ is closed and $U(T, I)$ is a closure space.*

Proof. By application of proposition 3, we know that $P(T, I)$ is closed. Let us examine $U(T, I)$. By definition:

$$U(T, I) = T \cup \big(T \circ C(B(I))\big)$$

We remark that $T = T \circ \{\mathbf{A}\}$. By reporting this observation in the equation:

$$U(T, I) = (T \circ \{\mathbf{A}\}) \cup \big(T \circ C(B(I))\big)$$

By factorizing, we obtain:

$$U(T, I) = T \circ \big(C(B(I)) \cup \{\mathbf{A}\}\big)$$

Since $\bigcap S = \bigcap(S \cup \{\mathbf{A}\})$ for any nonempty subset S of features, it follows that:

$$U(T, I) = T \circ \big(C(B(I) \cup \{\mathbf{A}\})\big)$$

The two terms in the right hand side of the equation are closed theories containing the element $\mathbf{A}$. Hence, by proposition 3, $U(T, I)$ is a closure space. □

The salient characteristic of the projection operator is to compile the feature space $\mathbf{F}(I)$ of an interpretation I into a structure that exploits the commonalities between features. This is formalized in the next property.

Proposition 7. *Let T be a closure space and I be an interpretation. Then the congruence relation of $P(T, I)$ induces a complete partitioning of $\mathbf{F}(I)$ into $|P(T, I)|$ congruence classes.*

Proof. By proposition 6, $P(T, I)$ is closed. Thus, by proposition 5, it follows that the congruence relation of $P(T, I)$ induces a complete partitioning of the feature space of $P(T, I)$ into $|P(T, I)|$ congruence classes. So, we simply need to show that $\mathbf{F}(P(T, I)) = \mathbf{F}(I)$. Let F be an element of $\mathbf{F}(P(T, I))$. By construction, $F \subseteq F'$ for some element in F' in $P(T, I)$, and $F' \subseteq F''$ for some element F'' in $C(B(I))$. Thus, F is covered by some maximal element in the basis of I and hence, by proposition 1, $F \in \mathbf{F}(I)$. Conversely, let F be an element of $\mathbf{F}(I)$. Then, by proposition 1, $F \subseteq F'$ for some element F' in $B(I)$. Since $F' \cap \mathbf{A} = F'$, it follows that $F' \in P(T, I)$. Hence, $F \in \mathbf{F}(P(T, I))$. □

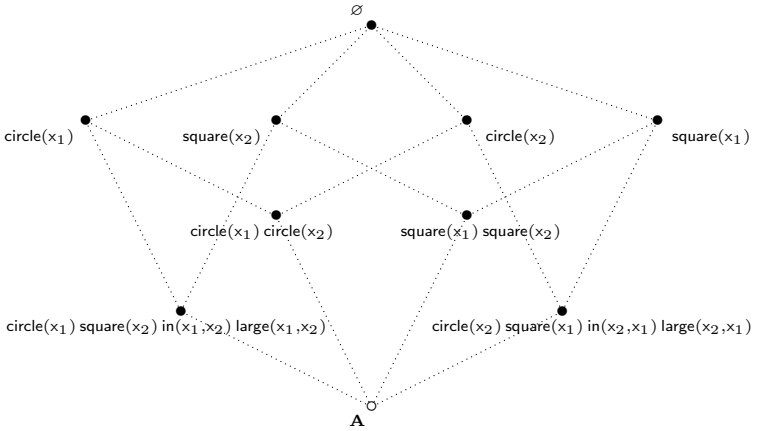

Fig. 2. Update of T by I

Example 5. Let $T = \{\mathbf{A}\}$ and consider the interpretation I given in example 2. The update of T by I is represented by the lattice in figure 2. The projection of T by I is formed by the set of all "•" nodes. Based on the above result, $P(T, I)$ induces a complete partitioning of $\mathbf{F}(I)$ into 9 congruence classes. By comparison, $\mathbf{F}(I)$ contains 33 features.

3.3 Linear Functions of Closed Theories

We have now all elements in hand to define online relational predictors in the setting of closure-based induction. Let T be a closure space. A *linear threshold function of* T is a map H that associates to each feature F in T a weight in $\mathbb{R}^+$. Intuitively, T can be regarded as a complied representation of $\mathbf{F}$ that is iteratively constructed during the mistakes made by the learner. The function H simply labels each closed feature F in T according to its degree of relevance. The *classifier* of H is a map that assigns to each interpretation I the boolean value $H(I)$ defined according to the following condition:

$$H(I) = \begin{cases} 1 \text{ if } \left(\sum_{F \in P(T,I)} H(C_T(F)) \cdot \|F\|_{P(T,I)}\right) \geq N, \text{ and} \\ 0 \text{ otherwise} \end{cases}$$

The prediction obtained from the classifier H can be explained as follows. Initially, the learner has at its disposal a closure space T and a linear threshold function H of T. Given an observation I, the learner first computes the projection of T onto I. Then, for each feature F in the projected set, the learner evaluates the degree of relevance of the congruence class of F. In doing so, the learner considers that each element in the class has the same weight, which is given by the feature $C_T(F)$ in T. Thus, the learner only needs to multiply this weight by the number of features in the congruence class. This strategy is applied for all congruence classes and the overall sum is compared with the threshold N.

We conclude this section by establishing a one-to-one correspondence between the two forms of linear functions investigated in this study. Let Φ be a linear threshold function of $\mathbf{F}$ and H be a linear threshold function of some given closure space T. Then, we say that H is a *closure-based representation* of Φ if $\Phi(F) = H(C_T(F))$ for every feature F in the space $\mathbf{F}$.

Proposition 8. *Let T be a closure space. Let Φ and H be linear threshold functions of $\mathbf{F}$ and T, respectively. If H is a closure-based representation of Φ, then for each interpretation I, $\Phi(I) = H(I)$.*

Proof. Suppose that H is a closure-based representation of Φ. Let F be an element of $P(T, I)$. By proposition 4, we know that $[F]_{P(T,I)} \subseteq [F]_T$. Since $\Phi(F) = H(C_T(F))$, then for each feature F' in the congruence class $[F]_{P(T,I)}$ we have $\Phi(F') = H(C_T(F')) = H(C_T(F))$. By adding up all weights:

$$\sum\{\Phi(F') : F' \in [F]_{P(T,I)}\} = H(C_T(F)) \cdot \|F\|_{P(T,I)}$$

Furthermore, by proposition 7, we know that the congruence relation of $P(T, I)$ induces a complete partitioning of $\mathbf{F}(I)$. It follows that:

$$\sum_{F \in \mathbf{F}(I)} \Phi(F) = \sum_{F \in P(T,I)} H(C_T(F)) \cdot \|F\|_{P(T,I)}$$

Using the definition of Φ, we therefore obtain:

$$\sum_{F \in \mathbf{F}} \Phi(F) \cdot F(I) = \sum_{F \in P(T,I)} H(C_T(F)) \cdot \|F\|_{P(T,I)}$$

Finally, since the classifiers Φ and H are defined on the same threshold N, we must have $\Phi(I) = H(I)$. □

Example 6. Consider the following scenario. The learner starts from the theory $T = \{\mathbf{A}\}$ and the linear function H such that $H(\mathbf{A}) = 2$. After receiving the interpretation I given in example 2, the projection of T onto I forms the theory represented in figure 2. We remark that: $\sum_{F \in P(T,I)} H(C_T(F)) \cdot \|F\|_{P(T,I)} = 66$. Since $N = 2^{18}$, the example I is classified as negative. Suppose that I is, in fact, a positive example of the target concept. In this case, we consider that the new closure space T is obtained from the update of the initial theory $\{A\}$ by I. Furthermore, we consider that the new linear function H is obtained from the original function by multiplying by 2 the weight of each feature F in $P(T, I)$. Now suppose that the learner receives a new interpretation J given by:

$$J = (\{1, 2\}, \{\mathsf{triangle}(1), \mathsf{triangle}(2), \mathsf{larger}(1, 2), \mathsf{left}(1, 2)\})$$

The projection of T onto J is represented by the set of all "•" nodes in figure 3. We remark that: $\sum_{F \in P(T,J)} H(C_T(F)) \cdot \|F\|_{P(T,J)} = 62$. Again, the example is classified as negative. Suppose that J is, in fact, positive. Then, the new closure space T is obtained from the update of the original theory by J. This theory is represented by the complete lattice in figure 3. We notably remark that T partitions the feature space $\mathbf{F}$ into 15 congruences classes. By comparison, $\mathbf{F}$ contains $262,144$ features.

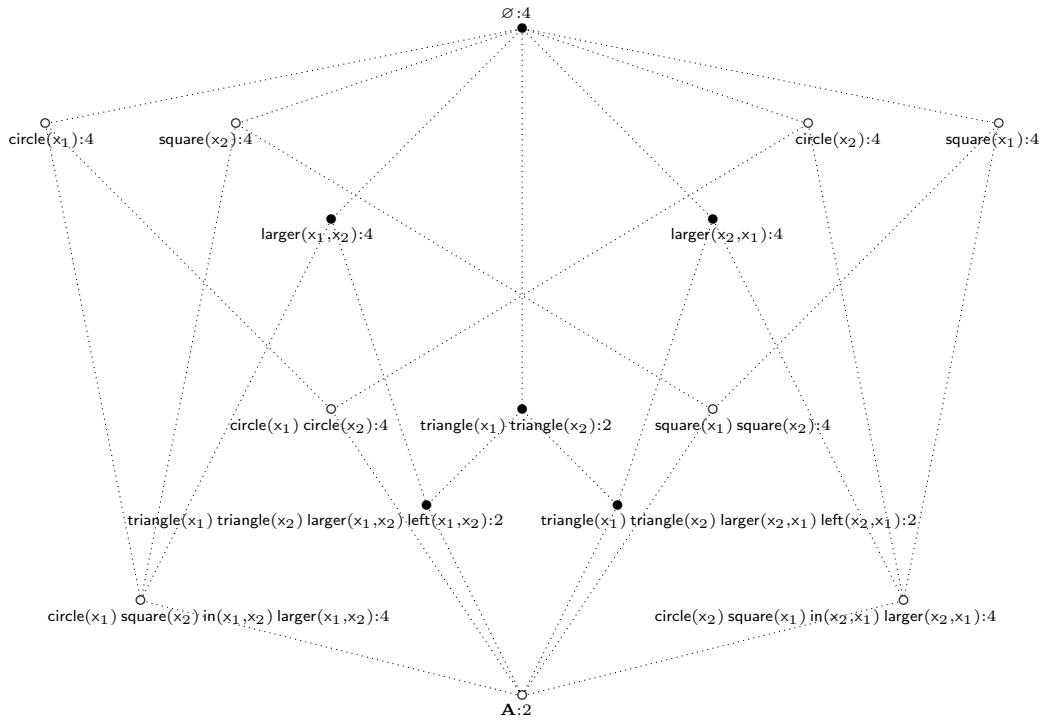

Fig. 3. Update of T by J

4 Online Closure-Based Learning

After an excursion into the algebraic aspects of our framework, we now focus on closure-based relational learning. In this section, we begin to present an online learning algorithm for relational committees, next we provide a mistake bound for this algorithm, and then we examine its computational cost.

The algorithm is specified in figure 4. The learner starts with the closure space $\{\mathbf{A}\}$, where $H(\{\mathbf{A}\})$ is set to 2. The order of the events in any trial is organized as follows. First, the learner receives an interpretation I from its environment. Next it predicts a class label for I by projecting its closure space T onto I and by computing the value $H(I)$ of its corresponding classifier. In doing so, the learner exploits the topological structure of its closure space T in order to determine the overall weight of the feature space $\mathbf{F}(I)$. Finally, the learner receives the correct label. If the algorithm has made a mistake, then it updates its linear threshold function H and its theory T. The learner starts by expanding the domain of H to $P(T, I)$. The weights of the features are increased or decreased, according to the type of mistake that has been made. Then, the learner updates its closure space T by I, and waits for a new example.

Initialization

0 Set $T \leftarrow \{\mathbf{A}\}$ and $H(\mathbf{A}) \leftarrow 2$

Trials

1 Receive an interpretation I

2 If $\left(\sum_{F \in P(T,I)} H(C_T(F)) \cdot \|F\|_{P(T,I)}\right) \geq N$, then

predict $H(I) \leftarrow 1$

else

predict $H(I) \leftarrow 0$

3 Receive $T^*(I)$. If $T^*(I) \neq H(I)$ then

Demotion: if $H(I) = 1$ then $\forall F \in P(T, I)$, set $H(F) \leftarrow \frac{1}{2} H(C_T(F))$

Promotion: if $H(I) = 0$ then $\forall F \in P(T, I)$, set $H(F) \leftarrow 2\, H(C_T(F))$

Set $T \leftarrow T \cup P(T, I)$

Fig 4: Closure-Based Relational Winnow

4.1 Mistake Bound

We have now all elements in hand to provide the first main result of this study. In the next theorem, we consider that the target expression T^* is a relational theory containing r relational conjunctions. The goal for the learner is to identify these r relevant features in a feature space $\mathbf{F}$ containing an exponential number $N - r$ of irrelevant features. Based on a natural correspondence between the standard algorithm and the closure-based algorithm, we can state that the number of mistakes depends only logarithmically on N and linearly on r.

Theorem 1. *For the class of relational theories containing r existentially quantified conjunctions of atoms defined over p predicate symbols and k variables, online closure-based Winnow has a mistake bound of:*

$$2(rpk^a + 1)$$

Proof. Let Φ and H be the linear threshold functions maintained by the standard algorithm (fig. 2) and the closure-based algorithm (fig. 4). We show that, if both algorithms have received the same sequence $\mathbf{I}$ of examples, then for any new received example J, we have $\Phi(J) = H(J)$. Based on proposition 8, a sufficient condition for this is to prove that H is a closure-based representation of Φ. This is demonstrated by induction on the size of the sequence $\mathbf{I}$.

First, suppose that the sequence $\mathbf{I}$ is empty. We remark that for each feature F in $\mathbf{F}$, $\Phi(F) = H(\mathbf{A}) = 2$. Since $C_{\{\mathbf{A}\}}(F) = \mathbf{A}$, it follows that $\Phi(F) = H(C_T(F))$. Hence, H is a closure-based representation of Φ.

Now, suppose that $\mathbf{I}$ is not empty. We focus on the last trial in the sequence. Let I be the example observed during this trial. Let E_{bef} and E_{aft} denote the expression E at the beginning of the trial and at the end of the trial. Finally, let F be a feature in $\mathbf{F}$. By induction hypothesis, we assume that $\Phi_{\text{bef}}(F) = H_{\text{bef}}(C_{T_{\text{bef}}}(F))$ at the beginning of the trial. We must show that $\Phi_{\text{aft}}(F) = H_{\text{aft}}(C_{T_{\text{aft}}}(F))$. If this condition holds, then H is still a closure-based representation of Φ at the end of the last trial of the sequence. Consequently, $\Phi(J) = H(J)$ during any new trial involving J. Suppose that no mistake occurred. In this case, $\Phi_{\text{aft}} = \Phi_{\text{bef}}$. Similarly, $H_{\text{aft}} = H_{\text{bef}}$ and $T_{\text{aft}} = T_{\text{bef}}$. Hence, we have $\Phi_{\text{aft}}(F) = H_{\text{aft}}(C_{T_{\text{aft}}}(F))$. Suppose that a mistake occurred. Then both classifiers are either "demoted" or "promoted". We only examine the demotion case, since an analogue strategy applies to the promotion case.

We know that $T_{\text{aft}} = T_{\text{bef}} \cup P(T_{\text{bef}}, I)$. First, consider that $F \notin \mathbf{F}(I)$. In this case, $\Phi_{\text{aft}}(F) = \Phi_{\text{bef}}(F)$. Furthermore, F must be an element of some congruence class in T_{bef}. Therefore, $C_{T_{\text{aft}}}(F) = C_{T_{\text{bef}}}(F)$. Since $H_{\text{aft}}(C_{T_{\text{bef}}}(F)) = H_{\text{bef}}(C_{T_{\text{bef}}}(F))$, we have $H_{\text{aft}}(C_{T_{\text{aft}}}(F)) = H_{\text{bef}}(C_{T_{\text{bef}}}(F))$. Hence, $\Phi_{\text{aft}}(F) = H_{\text{aft}}(C_{T_{\text{aft}}}(F))$. Now, consider that $F \in \mathbf{F}(I)$. In this case, we must have $\Phi_{\text{aft}}(F) = \frac{1}{2}\Phi_{\text{bef}}(F)$. Furthermore, F is an element of some congruence class in $P(T_{\text{bef}}, I)$. It follows that, $C_{T_{\text{aft}}}(F) = C_{P(T_{\text{bef}},I)}(F)$. Since $H_{\text{aft}}(C_{P(T_{\text{bef}},I)}(F)) = \frac{1}{2}H_{\text{bef}}(C_{T_{\text{bef}}}(F))$, we have $H_{\text{aft}}(C_{T_{\text{aft}}}(F)) = \frac{1}{2}H_{\text{bef}}(C_{T_{\text{bef}}}(F))$. Therefore, $\Phi_{\text{aft}}(F) = H_{\text{aft}}(C_{T_{\text{aft}}}(F))$.

We thus have shown that closure-based Winnow is a simulation of standard Winnow. Consequently, if the later algorithm has a mistake-bound of m, then the former algorithm must have a mistake bound of m. By an adaptation of Littlestone's analysis (see also [21]), standard Winnow has a mistake bound of $2(r \log_2 N + 1)$. Since N is upper bounded by 2^{pk^a}, the result follows. □

4.2 Computational Complexity

Obviously, the main source of complexity in closure-based Winnow resides in the prediction phase. This phase can be divided into two steps. Namely, given a closure space T and an interpretation I, the learner computes first the projection of T onto I. Then, for each closed feature F in the projection, the learner evaluates the weight of F and the cardinality of the congruence class of F. The following property suggests a simple incremental procedure to build projections.

Proposition 9. *Let T be a closure space and I be an interpretation. Suppose that the basis of I is given by the set $\{F_1, \cdots, F_n\}$ and let $(P_0, \cdots, P_n)$ be the sequence of sets of features recursively defined as follows:*

(1) $P_0 = \varnothing$,
(2) $P_n = P_{n-1} \cup \{F \cap F_i : F \in T \cup P_{i-1}\}$.

Then P_n is the projection of T onto I.

Proof. Let B_n denote the set $\{F_1, \cdots, F_n\}$. The proof is done by induction on n. First, suppose that $n = 1$. In this case, we know that $C(B_1) = B_1 = \{F_1\}$. Since $P_1 = \{F \cap F_1 : F \in T\}$, it follows that $T \circ C(B_1) = P_1$, as desired.

Now, consider that $n > 1$ and, by induction hypothesis, assume that P_{n-1} is given by $T \circ C(B_{n-1}(I))$. We first prove that

$$C(B_n) = C(B_{n-1}) \cup (C(B_{n-1}) \circ \{F_n\}) \cup \{F_n\}$$

We know that $C(B_n)$ is closed under intersection. Let C_n denote the set of all intersections of nonempty subsets of B_n containing F_n. By construction, C_n is given by $\{F_n\} \cup \{\bigcap S \cap F_n : \varnothing \subset S \subseteq B_{n-1}\}$. Since $C(B_{n-1})$ is the set $\{\bigcap S : \varnothing \subset S \subseteq B_{n-1}\}$, it follows that: $C_n = \{F_n\} \cup \{F_n \cap F : F \in C(B_{n-1})\}$. Finally, since the second term corresponds to $C(B_{n-1}) \circ \{F_n\}$, the result follows. Now, we examine the main property. By construction, we have:

$$T \circ C(B_n) = (T \circ C(B_{n-1})) \cup (T \circ \{F_n\}) \cup (T \circ C(B_{n-1}) \circ \{F_n\})$$

By induction hypothesis, we know that $T \circ C(B_n) = P_{n-1}$. By reporting this result, $T \circ C(B_n)$ is $P_{n-1} \cup (T \circ \{F_n\}) \cup (P_{n-1} \circ \{F_n\})$. By factorizing, it follows that $T \circ C(B_n) = P_{n-1} \cup ((T \cup P_{n-1}) \circ \{F_n\})$. Since the second term is the set $\{F \cap F_n : F \in T \cup P_{i-1}\}$, the result follows. □

The following property suggests a simple method to evaluate the cardinality of any congruence class of a closed set.

Proposition 10. *Let T be a closed theory and $\{F_1, \cdots, F_n\}$ be a linear ordering of T where $F_i \subset F_j$ implies $i \leq j$ for any pair of indexes i and j. Then the cardinality of each congruence class in T is recursively determined as follows:*

(1) $\|F_1\| = 2^{|F_1|}$,
(2) $\|F_n\| = 2^{|F_n|} - \sum\{\|F_i\| : 1 \leq i < n \text{ and } F_i \subseteq F_n\}$.

Proof. Let T_n and $\mathbf{F}_n$ denote the sets of every subset of F_n in T and $\mathbf{F}(T)$, respectively. By proposition 5, we know that the congruence relation of T induces a complete partitioning of $\mathbf{F}(T)$. Since each element in $\mathbf{F}_n$ must be covered by some congruence class in T_n, it follows that the congruence relation of T_n induces a complete partitioning of $\mathbf{F}_n$. We thus have,

$$\mathbf{F}_n = \bigcup\{[F_i] : 1 \leq i \leq n \text{ and } F_i \subseteq F_n\}$$

We now examine the main property. First, consider that $n = 1$. In this case, $|\mathbf{F}_1| = \|F_1\|$. Since $|\mathbf{F}_1| = 2^{|F_1|}$, the result follows. Now, consider that $n > 1$. From the previous equation, we have:

$$[F_n] = \mathbf{F}_n - \bigcup\{[F_i] : 1 \leq i < n \text{ and } F_i \subseteq F_n\}$$

Since $|\mathbf{F}_n| = 2^{|F_n|}$ and congruence classes are mutually disjoint, we obtain:

$$\|F_n\| = 2^{|F_n|} - \sum\{\|F_i\| : 1 \leq i < n \text{ and } F_i \subseteq F_n\}$$

Based on these considerations, we are now in position to present the second key result of this paper. The next theorem states that the computational cost is polynomial in the size of the closure lattice.

Theorem 2. *Let s be the size of the closure space maintained by the learner at the beginning of some trial. Let d and b denote the number of objects and the size of the closure of the basis of the received interpretation. Then, the time complexity of the trial is in $O(b^2s^2 + d^{2k})$.*

Proof. Let T be the closure space maintained by the learner and I be the received interpretation at the beginning of the trial (line 1). We assume that elements in T are sorted. We first examine the complexity of the prediction step (line 2). As observed earlier, the construction of the basis takes $O(d^{2k})$ time. Based on the method suggested by proposition 9, the projection of T onto I takes $O(bs)$ time. Furthermore, the resulting theory is sorted and contains at most bs features. For each feature F in $P(T, I)$, the weight $H(C_T(F))$ can be evaluated in $O(s)$ time. Furthermore, using the method suggested in proposition 10, the value $\|F\|_{P(T,I)}$ can be obtained in $O(bs)$ time. Since there are at most bs features in $P(T, I)$, the counting task takes $O(bs(s + bs))$ time, which is in $O(b^2s^2)$. We now turn to the complexity of the update step (line 3). Updating the linear function H requires $O(bs)$ time since the weights were already computed in the prediction step. The update of T by I requires $O(bs \log_2(s))$ time. □

If we consider constant values of the maximum arity a and the number of variables k, then the computational cost is essentially dependent on the size of the closure space T. This space T is isomorphic to a concept lattice [8] formed by the context (G, M, I) where the set of "objects" G is given by the set $B(T)$ of all maximal elements of T with respect to set-inclusion, the set of "attributes" M is given by $\mathbf{A}$ and the "incidence relation" I is given by the membership relation between $\mathbf{A}$ and $B(T)$. Following [15], this lattice can be exponential in the number of its maximal elements. Yet, as stressed in introduction to this paper, experiments in formal concept analysis suggest that such an exponential bound is rarely observed in practice. On average, the size of a closure lattice tends to be quadratic in the number of the attributes (or atoms) [5, 9].

5 Conclusions

Online relational learning is intrinsically characterized by a dilemma between effectiveness and computational complexity. On the one hand, the mistake bound of multiplicative weight algorithms is only logarithmic in the input dimension, making them useful to handle large spaces such as relational theories. On the other hand, standard online relational learners are fundamentally limited by the counting problem that requires a systematic exploration of these spaces. The key contribution of this study is to provide a model of closure-based learning that allows a learner to focus on limited regions of its hypothesis space and to compile these regions into a closure lattice. This paradigm was applied to the development of an online algorithm for learning relational theory. The number of mistakes depends only logarithmically on the number of features and the computational cost is polynomially bounded by the size of the closure lattice.

Related Work. In the past few years, there have been an increased an significant interest in the development of online learning algorithms for relational domains. In a seminal work, Golding and Roth [10] developed a relational architecture, the SNoW system, that learns linear threshold functions with quantified propositions. This architecture has been applied to several structured domains, including visual recognition [19] and information extraction [20]. The basic idea underlying the notion of quantified proposition is to limit the scope of each quantifier to a single predicate. In other words, only atoms are quantified and thus, any formula can be treated essentially as a logical combination of boolean variables [7]. Based on this representation, the number of mistakes still depends only logarithmically on the number of quantified atoms. Soon afterwards, Valiant [21, 22] extended this approach by addressing the class of quantified projections, an intermediate class between quantified disjunctions and quantified DNF formulas. Based on a combination of Winnow algorithms, the method preserves attribute-efficiency by exhibiting a logarithmic dependence on the number of quantified atoms.

The main interest of these approaches is to extend the expressiveness of pure propositional systems while maintaining a polynomial cost during the learning phase. By comparison, our paradigm is orthogonal to these approaches. Namely, the use of multi-class, first-order decision rules provides a far more expressive language. In particular, existentially quantified conjunctions of atoms are able to capture both relations among objects and dependencies between relations. Yet, despite the use of closure-based operations, the dependence of the computational cost on the input dimension is not guaranteed to be polynomial.

Finally, the recent work by Chawla et. al. [6] is also concerned with generalizing Winnow algorithms to large spaces. But their approach is essentially propositional and uses a randomized approximation technique that does not always guarantee complete correctness of the learning system.

Perspectives. Several directions of future research are possible. First and topmost, the practical issue of online closure-based learning needs to be explored. In particular, the development of a competence map for our algorithm is the subject of ongoing research. A second interesting research avenue is to develop pruning techniques for closure spaces. A potential strategy is to merge congruence classes that have the same weight vectors. An other approach is to use lower and upper bounds on the possible weights in order to limit the number of distributions. Third and finally, the framework described suggests a broader variety of relational classes that might be handled using the paradigm of closure-based learning. In particular, the extension of this approach to first-order Horn theories looks promising.

References

[1] S. Abiteboul, R. Hull, and V. Vianu. *Foundations of Databases.* Addison-Wesley, 1995.

[2] G. Birkhoff. *Lattice Theory.* American Mathematical Society, Third Edition, 1967.

[3] A. Blum. Learning boolean functions in an infinite attribute space. *Machine Learning*, 9(4):373–386, 1992.

[4] A. Blum. On-line algorithms in machine learning. In *Online Algorithms*, volume 1442 of *Lecture Notes in Computer Science*, pages 306–325, 1998.

[5] C. Carpineto, G. Romano, and P. d'Amado. Inferring dependencies from relations: a conceptual clustering approach. *Computational Intelligence*, 15(4):415–441, 1999.

[6] D. Chawla, L. Li, and S. Scott. Efficiently approximating weighted sums with exponentially many terms. In *Proceedings of the 14th Annual Conference on Computational Learning Theory*, pages 82–98, 2001.

[7] C. M. Cumby and D. Roth. Relational representations that facilitate learning. In *Principles of Knowledge Representation and Reasoning: Proceedings of the 7th International Conference*, pages 425–434, 2000.

[8] B. Ganter and R. Wille. *Formal Concept Analysis: Mathematical Foundations*. Springer-Verlag, 1997.

[9] R. Godin, R. Missaoui, and H. Alaoui. Incremental concept formation algorithms based on Galois lattices. *Computational Intelligence*, 11:246–267, 1995.

[10] A. R. Golding and D. Roth. A Winnow-based approach to context-sensitive spelling correction. *Machine Learning*, 34:107–130, 1999.

[11] S. A. Goldman, S. Kwek, and S. D. Scott. Agnostic learning of geometric patterns. *Journal of Computer and System Sciences*, 62(1):123–151, 2001.

[12] D. Haussler. Learning conjunctive concepts in structural domains. *Machine Learning*, 4:7–40, 1989.

[13] R. Khardon. Learning horn expressions with LogAn-H. In *Proceedings of the 17th International Conference on Machine Learning*, pages 471–478, 2000.

[14] R. Khardon, D. Roth, and R. A. Servedio. Efficiency versus convergence of boolean kernels for on-line learning algorithms. In *Advances in Neural Information Processing Systems*, volume 14, pages 423–430, 2001.

[15] S. Kuznetsov. On computing the size of a lattice and related decision problems. *Order*, 18(4):313–321, 2001.

[16] N. Littlestone. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine Learning*, 2(4):285–318, 1988.

[17] W. Maass and M. K. Warmuth. Efficient learning with virtual threshold gates. *Information and Computation*, 141(1):66–83, 1998.

[18] T. M. Mitchell. Generalization as search. *Artificial Intelligence*, 18(2):203–226, 1982.

[19] D. Roth, M.-H. Yang, and N. Ahuja. Learning to recognize three-dimensional objects. *Neural Computation*, 14(5):1071–1103, 2002.

[20] D. Roth and W. Yih. Relational learning via propositional algorithms: An information extraction case study. In *Proceedings of the 17th International Joint Conference on Artificial Intelligence*, pages 1257–1263, 2001.

[21] L. G. Valiant. Projection learning. *Machine Learning*, 37(2):115–130, 1999.

[22] L. G. Valiant. Robust logics. *Artificial Intelligence*, 117(2):231–253, 2000.