Metric-based inductive learning using semantic height functions

Zdravko Markov[1] and Ivo Marinchev[2]

[1] Department of Computer Science, Central Connecticut State University
1615 Stanley Street, New Britain, CT 06050, U.S.A.
E-mail: markovz@ccsu.edu
[2] Faculty of Mathematics and Informatics, University of Sofia
5 James Bouchier Str., 1164 Sofia, Bulgaria
E-mail: ivo@fmi.uni-sofia.bg

Abstract. In the present paper we propose a consistent way to integrate syntactical least general generalizations (lgg's) with semantic evaluation of the hypotheses. For this purpose we use two different relations on the hypothesis space – a constructive one, used to generate lgg's and a semantic one giving the coverage-based evaluation of the lgg. These two relations jointly implement a *semantic distance measure*. The formal background for this is a height-based definition of a semi-distance in a join semi-lattice. We use some basic results from lattice theory and introduce a family of language independent coverage-based height functions. The theoretical results are illustrated by examples of solving some basic inductive learning tasks.

1 Introduction

Inductive learning addresses mainly classification tasks where a series of training examples (instances) are supplied to the learning system and the latter builds an intensional or extensional representation of the examples (hypothesis), or directly uses them for prediction (classification of unseen examples). Generally two basic approaches to inductive learning are used. The first one is based mainly on generalization/specialization or similarity-based techniques. This approach includes two types of systems – *inductive learning from examples* and *conceptual clustering*. They both generate inductive hypotheses made by abstractions (generalizations) from specific examples and differ in the way examples are presented to the system (whether or not they are pre-classified). The basic techniques used within the second approach are various kinds of *distances* (*metrics*) over the example space which are used to classify directly new examples (by similarity to the existing ones) or group the examples into clusters.

There exists a natural way to integrate consistently the generalization-based and metric-based approaches. The basic idea is to estimate the similarity between two objects in a hierarchical structure by the distance to their closest common parent. This idea is formally studied within the lattice theory. In ML

this is the well known *least general generalization (lgg)* which given two hypotheses builds their most specific common generalization. The existence of an *lgg* in a hypothesis space (a partially ordered set) directly implies that this space is a semi-lattice (where the lgg plays the role of infimum). Consequently some algebraic notions as finiteness, modularity, metrics etc. can be used to investigate the properties of the hypothesis space. Lgg's exist for most of the languages commonly used in ML. However all practically applicable (i.e. computable) lgg's are based on *syntactical* ordering relations. A relation over hypotheses is syntactical if it does not account for the background knowledge and for the coverage of positive/negative examples. For example dropping condition for nominal attributes, instance relation for atomic formulae and θ-subsumption for clauses are all syntactical relations. On the other hand the evaluation of the hypotheses produced by an lgg operator is based on their coverage of positive/negative examples with respect to the background knowledge, i.e. it is based on *semantic relations* (in the sense of the inductive task). This discrepancy is a source of many problems in ML, where overgeneralization is the most difficult one.

In the present paper we propose a consistent way to integrate syntactical lggs with semantic evaluation of the hypotheses. For this purpose we use two different relations on the hypothesis space – a constructive one, used to generate lgg's and a semantic one giving the coverage-based evaluation of the lgg. These two relations jointly implement a *semantic distance measure*. The formal background for this is a height-based definition of a semi-distance in a join semi-lattice. We use some basic results from lattice theory and introduce a language independent coverage-based height function. We also define the necessary conditions for two relations to form a correct height function. The paper introduces a bottom-up inductive learning algorithm based on the new semantic semi-distance which is used to illustrate the applicability of the theoretical results.

The paper is organized as follows. The next section introduces the basic algebraic notions used throughout the paper. Section 3 introduces the new a heihgt-based semi-distance. Section 4 presents an algorithm for building lattice structures and shows some experiments with this algorithm. Section 5 contains concluding remarks and directions for future work.

2 Preliminaries

In this section we introduce a height-based distance measure on a join semi-lattice following an approach similar to those described in [1] and [5] (for a survey of metrics on partially ordered sets see [2]).

Definition 1 (Semi-distance, Quasi-metric). *A semi-distance (quasi-metric) is a mapping $d : O \times O \to \Re$ on a set of objects O with the following properties $(a, b, c \in O)$:*

1. $d(a, a) = 0$ *and* $d(a, b) \geq 0$.
2. $d(a, b) = d(b, a)$ *(symmetry).*
3. $d(a, b) \leq d(a, c) + d(c, b)$ *(triangle inequality).*

Definition 2 (Order preserving semi-distance). *A semi-distance $d : O \times O \to \Re$ on a partially ordered set $(O, \preceq)$ is order preserving iff $\forall a, b, c \in O$: $a \preceq b \preceq c \Rightarrow d(a, b) \leq d(a, c)$ and $d(b, c) \leq d(a, c)$*

Definition 3 (Join/Meet semi-lattice). *A join/meet semi-lattice is a partially ordered set $(A, \preceq)$ in which every two elements $a, b \in A$ have an infimum/supremum.*

Definition 4 (Size). *Let $(A, \preceq)$ be a join semi-lattice. A mapping $s : A \times A \to \Re$ is called a size function if it satisfies the following properties:*

S1. $s(a, b) \geq 0, \forall a, b \in A$ *and* $a \preceq b$.
S2. $s(a, a) = 0, \forall a \in A$.
S3. $\forall a, b, c \in A : a \preceq c$ *and* $c \preceq b \Rightarrow s(a, b) \leq s(a, c) + s(c, b)$.
S4. $\forall a, b, c \in A : a \preceq c$ *and* $c \preceq b \Rightarrow s(c, b) \leq s(a, b)$.
S5. $\forall a, b \in A$. *Let* $c = inf\{a, b\}$. *For any* $d \in A : a \preceq d$ *and* $b \preceq d \Rightarrow s(c, a) + s(c, b) \leq s(a, d) + s(b, d)$.

Theorem 1. *Let $(A, \preceq)$ be a join semi-lattice and s – a size function. Let $d(a, b) = s(inf\{a, b\}, a) + s(inf\{a, b\}, b)$. Then d is a semi-distance on $(A, \preceq)$.*

Proof. 1. d is non-negative by $S1$.
2. $d(a, a) = s(inf\{a, a\}, a) + s(inf\{a, a\}, a) = s(a, a) + s(a, a) = 0$.
3. d is symmetric by definition.
4. We will show that $d(a_1, a_2) \leq d(a_1, a_3) + d(a_3, a_2)$. Let $c = inf\{a_1, a_2\}$, $b_1 = inf\{a_1, a_3\}$, $b_2 = inf\{a_2, a_3\}$, $d = inf\{b_1, b_2\}$. By $S4$ and $S3$ we have $s(c, a_1) \leq s(d, a_1) \leq s(d, b_1) + s(b_1, a_1)$. And by analogy $s(c, a_2) \leq s(d, b_2) + s(b_2, a_2)$. Then $d(a_1, a_2) = s(c, a_1) + s(c, a_2) \leq s(d, b_1) + s(b_1, a_1) + s(d, b_2) + s(b_2, a_2) \leq s(b_1, a_1) + s(b_1, a_3) + s(b_2, a_3) + s(b_2, a_2) = d(a_1, a_3) + d(a_2, a_3)$

A size function can be defined by using the so called *height functions*. The approach of height functions has the advantage that it is based on estimating the object itself rather than its relations to other objects.

Definition 5 (Height). *The function h is called height of the elements of a partially ordered set $(A, \preceq)$ if it satisfies the following two properties:*

1. *For every $a, b \in A$ if $a \preceq b$ then $h(a) \leq h(b)$ (isotone).*
2. *For every $a, b \in A$ if $c = inf\{a, b\}$ and $d \in A$ such that $a \preceq d$* and *$b \preceq d$ then $h(a) + h(b) \leq h(c) + h(d)$.*

Theorem 2. *Let $(A, \preceq)$ be a join semi-lattice and h be a height function. Let $s(a, b) = h(b) - h(a), \forall a \preceq b \in A$. Then s is a size function on $(A, \preceq)$.*

Proof. 1. $s(a, b) = h(b) - h(a) \geq 0$ by $H1$.
2. $s(a, a) = h(a) - h(a) = 0$.
3. Let $a, b, c \in A : a \preceq c, c \preceq b$. Then $s(a, b) = h(b) - h(a) = (h(b) - h(c)) + (h(c) - h(a)) = s(a, c) + s(c, b)$.

4. Let $a, b, c \in A : a \preceq c, c \preceq b$. Then $s(c,b) \leq s(c,b) + s(a,c) = s(a,b)$ by 3.
5. Let $c = inf\{a,b\}$ and $d \in A : a \preceq d$ and $b \preceq d$. Then $s(c,a) + s(c,b) = (h(a) - h(c)) + (h(b) - h(c)) = h(a) + h(b) - 2h(c) = 2(h(a) + h(b)) - h(a) - h(b) - 2h(c) \leq 2(h(c) + h(d)) - h(a) - h(b) - 2h(c) = (h(d) - h(a)) + (h(d) - h(b)) = s(a,d) + s(b,d)$

Corollary 1. *Let $(A, \preceq)$ be a join semi-lattice and h be a height function. Then the function $d(a,b) = h(a) + h(b) - 2h(inf\{a,b\}), \forall a, b \in A$ is a semi-distance on $(A, \preceq)$.*

3 Semantic semi-distance on join semi-lattices

Let A be a set of objects and let $\preceq_1$ and $\preceq_2$ be two binary relations in A, where $\preceq_1$ is a partial order and $(A, \preceq_1)$ is a join semi-lattice. Let also GA be the set of all maximal elements of A w.r.t. $\preceq_1$, i.e. $GA = \{a | a \in A$ and $\neg\exists b \in A : a \preceq_1 b\}$. Hereafter we call the members of GA *ground elements* (by analogy to ground terms in first order logic). For every $a \in A$ we denote by S_a the *ground coverage* of a w.r.t $\preceq_2$, i.e. $S_a = \{b | b \in GA$ and $a \preceq_2 b\}$.

The ground coverage S_a can be considered as a definition of the semantics of a. Therefore we call $\preceq_2$ a *semantic relation* by analogy to the Herbrand interpretation in first order logic that is used to define the semantics of a given term. The other relation involved, $\preceq_1$ is called *constructive (or syntactic) relation* because it is used to build the lattice from a given set of ground elements GA.

The basic idea of our approach is to use these two relations, $\preceq_1$ and $\preceq_2$ to define the semi-distance. According to Corollary 1 we use the syntactic relation $\preceq_1$ to find the infimum and the semantic relation $\preceq_2$ to define the height function h. The advantage of this approach is that in many cases there exists a proper semantic relation however it is intractable, computationally expensive or even not a partial order, which makes impossible its use as a constructive relation too (an example of such a relation is logical implication). Then we can use another, simpler relation as a constructive one (to find the infimum) and still make use of the semantic relation (to define the height function).

Not any two relations however can be used for this purpose. We will show that in order to define a correct semi-distance the two relations $\preceq_1$ and $\preceq_2$ must satisfy the following properties, which we call *coupling*.

Definition 6. *$\preceq_2$ is coupled with $\preceq_1$ if both conditions apply:*

1. *For every $a, b \in A$ such that $a \preceq_1 b$ either $|S_a| \geq |S_b|$ or $|S_a| \leq |S_b|$ must hold. As the other case is analogous without loss of generality we can assume that $\forall a, b \in A$, $a \preceq_1 b \Rightarrow |S_a| \geq |S_b|$.*
2. *$\forall a, b \in A : c = inf\{a,b\}$ and $\exists d = sup\{a,b\}$ one of the following must hold:*
   C1. $|S_d| < |S_a|$ and $|S_d| < |S_b|$
   C2. $|S_d| = |S_a|$ and $|S_c| = |S_b|$
   C3. $|S_d| = |S_b|$ and $|S_c| = |S_a|$

Corollary 2. *Every partial order relation is coupled with itself.*

Theorem 3. *Let A be a set of objects and let $\preceq_2$ and $\preceq_1$ be two binary relations in A such that $\preceq_2$ is coupled with $\preceq_1$. Then there exists a family of height functions $h(a) = x^{-|S_a|}$, where $a \in A$, $x \in \Re$ and $x \geq 2$.*

Proof. 1. Let $a, b \in A$, such that $a \preceq_1 b$. Then by the definition of coupling $|S_a| \geq |S_b|$ and hence $h(a) \leq h(b)$.

2. Let $a, b \in A : c = inf\{a, b\}$ and $\exists d = sup\{a, b\}$.
   (a) Assume that $C1$ is true. Then $|S_d| < |S_a|$ and $|S_d| < |S_b| \Rightarrow |S_a| \geq |S_d| + 1$ and $|S_b| \geq |S_d| + 1 \Rightarrow -|S_a| \leq -|S_d| - 1$ and $-|S_b| \leq -|S_d| - 1$. Hence $h(a) + h(b) = x^{-|S_a|} + x^{-|S_b|} \leq x^{-|S_d|-1} + x^{-|S_d|-1} = 2x^{-|S_d|-1} \leq x.x^{-|S_d|-1} = x^{-|S_d|} = h(d) \leq h(c) + h(d)$.
   (b) Assume that $C2$ is true. Then $|S_d| = |S_a|$ and $|S_c| = |S_b|$. Hence $h(a) + h(b) = h(c) + h(d)$.
   (c) Assume that $C3$ is true. Then $|S_d| = |S_b|$ and $|S_c| = |S_a|$. Hence $h(a) + h(b) = h(c) + h(d)$.

4 Experiments

To illustrate the theoretical results we use an algorithm that builds a join semi-lattice G, given a set of examples GA (the set of all maximal elements of G). The algorithm hereafter referred to as *MBI (Metric-based Bottom-up Induction)* is as follows:

1. Initialization: $G = GA$, $C = GA$;
2. If $|C| = 1$ then exit;
3. $T = \{h | h = lgg(a_1, a_2) : a_1, a_2 \in C$ and $d(a_1, a_2) = min\{d(b, c) | b, c \in C\}\}$;
4. $DC = \{h | h \in C$ and $\exists h_{min} \in T : h_{min} \preceq_2 h\}$;
5. $C = C \setminus DC$;
6. $G = G \cup T$, $C = C \cup T$, go to step 2.

There is a possible modification of this algorithm. In Step 3 instead of all, only one minimal element h from T can be used. With this modification the algorithm has a polynomial time complexity $O(n^3)$. A disadvantage of this modification is that some useful generalizations can be missed. Therefore in the practical implementations we augment the algorithm with another distance or heuristic measure used to select one of all minimal elements of T which possibly leads to the most useful generalizations.

Further in this section we discuss some experiments with the MBI algorithm with two different representation languages – atomic formulae and Horn clauses.

4.1 Atomic formulae

The algebraic properties of the language of first order atomic formulae are studied by Reynolds in [8], where he shows that the set of atoms with the same

functors and arity form a *join semi-lattice* (or *complete lattice* when the language is augmented by adding a 'universal atom' and a 'null atom'). In this framework we use $\preceq_1 = \preceq_2 = \theta$-subsumption and by Corollary 2 we have that θ-subsumption is coupled with itself.

Figure 1 shows the top portion of the lattice G built by the algorithm, where GA consists of the 61 *positive* examples of the well-known MONK1 [9] database (the training sample) represented as atoms. Note that the produced lattice can be used both for concept learning (it contains the target hypothesis monk(A,A,_,_,_,_) or monk(_,_,_,_,red,_)) and for conceptual clustering since the classifications of the examples are not used (the negative examples are skipped).

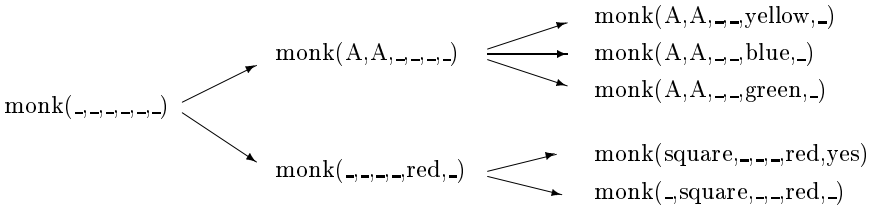

Fig. 1. Hypotheses for the MONK1 problem built by the MBI algorithm.

In more complex domains however the standard version of the algorithm performs poorly with small sets of randomly selected examples. In these cases we use the augmented version of the algorithm with a syntactic distance measure to choose one element of T in Step 3. In this way we avoid the random choice and allow "cautious" generalizations only. Further heuristics can be used for this purpose, especially in the case of background knowledge.

4.2 Horn clauses

Within the language of Horn clauses the MBI algorithm can be used with the *θ-subsumption-based lgg* (the constructive relation $\preceq_1$) and *logical implication* for the semantic relation $\preceq_2$. Under θ-subsumption as partial order the set of Horn clauses with same head predicates forms a semi-lattice. Furthermore, it can be shown that logical implication is coupled with θ-subsumption which makes the use of our algorithm well founded. Figure 2 shows the complete lattice build by the algorithm with 10 instances of the *member* predicate.

A major problem in bottom-up algorithms dealing with lgg_θ of clauses is the *clause reduction*, because although finite the length of the lgg_θ of n clauses can grow exponentially with n. Some well-known techniques of avoiding this problem are discussed in [3]. By placing certain restrictions on the hypothesis language the number of literals in the lgg_θ clause can be limited by a polynomial function independent on n. Currently we use *ij-determinate* clauses in our experiments (actually 22-determinate).

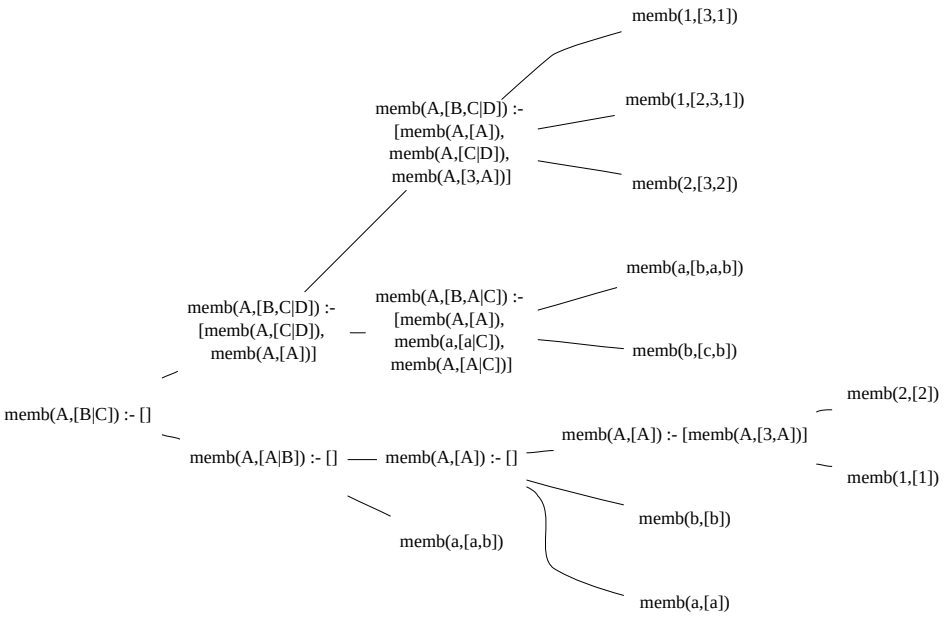

Fig. 2. ILP hypotheses for the instances of the *member* predicate.

5 Conclusion

The algebraic approach to inductive learning is a very natural way to study the generalization and specialization hierarchies. These hierarchies represent hypothesis spaces which in most cases are partially ordered sets under some generality ordering. In most cases however the oredirings used are based on *syntactical* relations, which do not account for the background knowledge and for the coverage of positive/negative examples. We propose an approach that explores naturally the semantic ordering over the hypotheses. This is because although based on syntactic lgg it uses a semantic evaluation function (the height function) for the hypotheses. Furthermore this is implemented in a consistent way through a height-based semi-distance defined on the hypothesis space.

As in fact we define a new distance measure our approach can be also compared to other metric-based approaches in ML. Most of them are based on attribute-value (or feature-value) languages. Consequently most of the similarity measures used stem from well known distances in feature spaces (e.g. Euclidean distance, Hamming distance etc.) and vary basically in the way the weights are computed. Recently a lot of attention has been paid to studying distance measures in first order languages. The basic idea is to apply the highly successful instance based algorithms to relational data using first order logic descriptions. Various approaches have been proposed in this area. Some of the most recent ones are [1, 4, 6, 7]. These approaches as well as most of the others define a simple metric on atoms and then extend it to sets of atoms (clauses or models) using

the Hausdorff metric or other similarity functions. Because of the complexity of the functions involved and the problems with the computability of the models these approaches are usually computationally hard. Compared to the other approaches our approach has two basic advantages. First, it is language independent, i.e. it can be applied both within propositional (attribute-value) languages and within first order languages and second, it allows consistent integration of generalization operators with a semantic distance measure.

We consider the following directions for future work. Firstly, particular attention should be paid to the clause reduction problem when using the language of Horn clauses. Other lgg operators, not based on θ-subsumption should be considered too.

The practical learning data often involve numeric attributes. In this respect proper relations, lgg's and covering functions should be investigated in order to extend the approach for handling numeric data.

Though the algorithm is well founded it still uses heuristics. This is because building the complete lattice is exponential and we avoid this by employing a hill-climbing strategy. It is based on additional distance measures or heuristics used to select the best lgg among all minimal ones (Step 3 of the algorithm). Obviously this leads to incompleteness. Therefore other strategies should be investigated or perhaps the semantic relation should be refined to incorporate these additional heuristics.

Finally, more experimental work needs to be done to investigate the behavior of the algorithm in noisy domains.

References

1. A. Hutchinson. Metrics on terms and clauses. In M. van Someren and G. Widmer, editors, *Machine Learning: ECML-97*, volume 1224 of *Lecture Notes in Artificial Intelligence*, pages 138–145. Springer-Verlag, 1997.
2. B. Monjardet. Metrics on partially ordered sets – a survey. *Discrete Mathematics*, 35:173–184, 1981.
3. S. Muggleton. Inductive logic programming. In S. Muggleton, editor, *Inductive Logic Programming*, pages 3–28. Academic Press, 1992.
4. S.-H. Nienhuys-Cheng. Distance between herbrand interpretations: a measure for approximations to a target concept. Technical Report EUR-FEW-CS-97-05, Erasmus University, 1997.
5. N. Pelov. Metrics on terms and clauses and application in inductive machine learning. Master's thesis, University of Sofia, 1997.
6. J. Ramon, M. Bruynooghe, and W. V. Laer. Distance measure between atoms. Technical Report CW 264, Katholieke Universiteit Leuven, 1998.
7. J. Ramon, M. Bruynooghe, and W. V. Laer. A framework for defining a distance between first-order logic objects. Technical Report CW 263, Katholieke Universiteit Leuven, 1998.
8. J. C. Reynolds. Transformational systems and the algebraic structure of atomic formulas. *Machine Intelligence*, 5:135–152, 1970.
9. S. B. Thrun et al. The MONK's problems - a performance comparison of different learning algorithms. Technical Report CS-CMU-91-197, Carnegie Mellon University, Dec. 1991.