Metrics on Terms and Clauses

Alan Hutchinson

Department of Computer Science, King's College London

Abstract. In the subject of machine learning, a "concept" is a description of a cluster of the concept's instances. In order to invent a new concept, one has to discover such a cluster. The necessary tool for clustering is a metric, or pseudo-metric. Here are presented families of pseudo-metrics which seem well suited to such tasks. On terms and literals, we construct a new kind of metric from the substitutions which arise through subsumption. From these, it is easy to form metrics on clauses, by a technique due to F.Hausdorff. They will be applicable to generalization from sets of ground clauses, to discovery of heuristic guidance for theorem proving, and to inductive logic programming.

We start by describing some pseudo-metrics on terms. They are constructed by means of Plotkin's *least general generalisation* (*lgg*) of two terms (see [6]). A term u is *subsumed* by another, w, if there is a substitution ϑ for which $w\vartheta = u$. The *lgg* of two terms u and v is a term w which subsumes them both, and which is itself subsumed by any other such w' subsuming u and v. If $w\vartheta = u$ and $w\varphi = v$ then the metric is a measure of the complexities of the substitutions ϑ and φ .

Example 1. The *lgg* of the two terms

$$play(Mary, Ann, James, skipping)$$

$$play(Mary, Ann, Ann, skipping)$$

is

$$play(Mary, Ann, x, skipping).$$

The second family contains pseudo-metrics on disjunctive clauses. A clause can be regarded as the finite set of its literal disjuncts. Syntactically, a literal is like a term, so the first metrics can be applied to literals. For each metric on literals, there is an associated *Hausdorff metric* on finite sets of literals. We present a few kinds of situations where these metrics may help.

1 Metrics on Terms

Suppose that S is a real-valued function on substitutions satisfying the following five conditions:

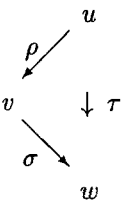
1. for any substitution ϑ, $S\vartheta \geq 0$;

2. $S\varepsilon = 0$ where ε is the identity substitution;
3. for any three terms u, v and w, if

$$\rho : u \longrightarrow v \quad \text{and} \quad \sigma : v \longrightarrow w \quad \text{and} \quad \tau : u \longrightarrow w$$

are all most general, then

$$S\tau \leq S\rho + S\sigma$$



4. under the same assumptions as for 3/,

$$S\sigma \leq S\tau$$

5. if u_1 and u_2 are any two terms which can be unified, and u is formed by unifying them with any substitutions

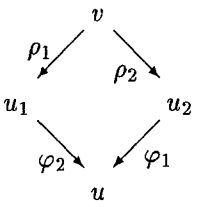
$$\varphi_1 : u_1 \longrightarrow u \quad \text{and} \quad \varphi_2 : u_2 \longrightarrow u$$

and v is a *lgg* of u_1 and u_2, and

$$\rho_1 : v \longrightarrow u_1 \quad \text{and} \quad \rho_2 : v \longrightarrow u_2 \quad \text{are most general,}$$

then

$$S\rho_1 + S\rho_2 \leq S\varphi_1 + S\varphi_2.$$



(Note that u and φ_1 and φ_2 need not be most general.)

Definition 1. In that case, say S is a *size* on substitutions.

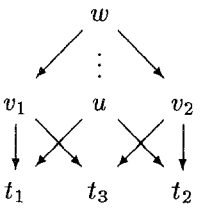
We shall examine a family of examples of sizes when we have seen why they are interesting.

If S is a size on substitutions, and t_1 and t_2 are any two terms, and u is their *lgg*, and $\vartheta_1 : u \longrightarrow t_1$ and $\vartheta_2 : u \longrightarrow t_2$ are most general, then say

$$d_S(t_1, t_2) = S\vartheta_1 + S\vartheta_2.$$

Proposition 2. *The function d_S is a pseudo-metric on the set of terms.*

Proof.
d_S is non-negative, by 1/.
d_S is symmetric, by the form of its definition.
$d_S(t,t) = 0$ for any term t, by 2/.
We show that $d_S(t_1,t_2) \leq d_S(t_1,t_3) + d_S(t_3,t_2)$.
Say u is the *lgg* of t_1 and t_2
v_1 is the *lgg* of t_1 and t_3
v_2 is the *lgg* of t_2 and t_3
w is the *lgg* of v_1 and v_2.



For convenience, write $S(ab)$ for $S\vartheta$ whenever $a\vartheta = b$ and ϑ is most general. By the conditions on the size function S,

$$\begin{aligned} S(ut_1) &\leq S(wt_1) && \text{by 4/} \\ &\leq S(v_1t_1) + S(wv_1) && \text{by 3/} \\ S(wv_1) + S(wv_2) &\leq S(v_1t_3) + S(v_2t_3) && \text{by 5/} \end{aligned}$$

so

$$S(ut_1) + S(ut_2) \leq S(v_1t_1) + S(v_1t_3) + S(v_2t_2) + S(v_2t_3)$$

□

2 Examples of Sizes

It remains to discover a family of size functions. Suppose that each function symbol f is assigned a non-negative real number wt_f called it *weight*. (A constant is just a function symbol of arity zero). For any subsitution ϑ, let

$$S\vartheta = \sum\{wt_f \mid \exists x(x \text{ is a variable and } f \text{ occurs in } x\vartheta)\}.$$

Note that each weight wt_f only occurs at most once in the sum, however often the symbol f may occur in values of ϑ.

Example 2. Suppose that

$$\vartheta = [f(g(a), f(a,b))/x,\ b/y,\ x/z]$$

$$wt_f = 2; \quad wt_g = 0; \quad wt_a = 4; \quad wt_b = 1; \quad wt_c = 6.$$

Then

$$S\vartheta = 2 + 0 + 4 + 1 = 7$$

Proposition 3. *Any such function S is a size.*

Proof.

1. Since wt_f is non-negative for each f, so is $S\vartheta$.
2. $S\varepsilon$ is the empty sum, which is 0.
3. τ is the restriction of $\rho\sigma$ to the free variables of u. Hence every function symbol occurring in values of τ also occurs in values of ρ or σ, so
$$S\tau \leq S\rho + S\sigma.$$
4. Suppose that, for some variable x, a function symbol f occurs in $x\sigma$. Since σ is most general, two cases may arise: either x occurs in u and $x\rho$ is x, or for some variable y occurring in u, x occurs in $y\rho$. If x occurs in u and $x\rho$ is x then $x\tau$ must be $x\sigma$. If x occurs in $y\rho$ where y occurs in u then $y\tau$ must be $y\rho\sigma$. In either event, f occurs in some value of τ. Hence
$$S\sigma \leq S\tau.$$
5. Suppose that f occurs in $x\rho_1$; then, since ρ_1 is most general, x must occur at some place p in v, and since v is least general, the terms $x\rho_1$ and $x\rho_2$ do not begin with the same symbol. If $x\rho_2$ were not a variable then u_1 and u_2 could not be unified; so $x\rho_2$ is some variable, say y. Since $v\rho_1\varphi_2 = v\rho_2\varphi_1 = u$,
$$x\rho_1\varphi_2 = x\rho_2\varphi_1 = y\varphi_1$$
and f occurs in $x\rho_1\varphi_2$ so f occurs in $y\varphi_1$, so $S\rho_1 \leq S\varphi_1$. Similarly, $S\rho_2 \leq S\varphi_2$ and so
$$S\rho_1 + S\rho_2 \leq S\varphi_1 + S\varphi_2.$$

□

3 Metrics on Clauses

As in reference [6], a clause can be regarded as the finite set of its disjuncts, which are atomic formulas and negated atomic formulas.

Example 3. The clause

$$likes(Mary, John) \ \wedge \ likes(John, Ann) \ \wedge \ \neg proud(John) \ \Rightarrow \\ play(Mary, Ann, John, skipping)$$

is equivalent to

$$\neg likes(Mary, John) \ \vee \ \neg likes(John, Ann) \ \vee \ proud(John) \ \vee \\ play(Mary, Ann, John, skipping)$$

so it can be depicted as the set of literals

$$\{\neg likes(Mary, John), \ \neg likes(John, Ann), \ proud(John), \\ play(Mary, Ann, John, skipping)\}.$$

We use the

Lemma (see [5] page 131, problem D). *If (X, d) is any metric space, then the function*

$$d'(A,B) = \max\begin{cases} \max_{x\in A} \min_{y\in B} d(x,y) \\ \max_{y\in B} \min_{x\in A} d(x,y) \end{cases}$$

is a metric on the set of nonempty finite subsets of X. If d is a pseudo-metric then so is d'.

This metric d' is often called the *Hausdorff metric* for d.

Proof. For any nonempty finite subsets A and B of X, $d'(A,B)$ is clearly well defined; non-negative; symmetric; zero if $A = B$; and, if d is a metric, nonzero if $A \neq B$. If $Y \subseteq X$ and $r > 0$ then say $V_r Y = \{x \in X \mid \exists y \in Y\ d(x,y) \leq r\}$. Observe that

$$d'(A,B) = \min\{r \mid A \subseteq V_r B \ \wedge\ B \subseteq V_r A\}.$$

If C is any other nonempty subset of X, and if

$$C \subseteq V_r A \ \wedge\ A \subseteq V_r C \ \wedge\ C \subseteq V_s B \ \wedge\ B \subseteq V_s C$$

then $\forall x \in A\ \exists z \in C\ (d(x,z) \leq r \ \wedge\ \exists y \in B\ d(z,y) \leq s)$ so $A \subseteq V_{r+s} B$. Similarly, $B \subseteq V_{r+s} A$. Hence if $d'(A,C) \leq r$ and $d'(B,C) \leq s$ then $d'(A,B) \leq r+s$. Hence $d'(A,B) \leq d'(A,C) + d'(B,C)$. □

Example 4. Suppose that (X,d) is the set of integers with the usual distance function, and

$$A = \{2,4,5\} \quad \text{and} \quad B = \{1,2,5,8\}$$

then

$d(x,y)$	$y \in B$:	1	2	5	8	$\min_{y\in B}\ d(x,y)$
$x \in A$:	2	1	0	3	6	0
	4	3	2	1	4	1
	5	4	3	0	3	0
	$\min_{x\in A}\ d(x,y)$	1	0	0	3	

so

$$\max_{x\in A} \min_{y\in B} d(x,y) = 1$$

$$\max_{y\in B} \min_{x\in A} d(x,y) = 3$$

so

$$d'(A,B) = 3.$$

Hausdorff's construction has become popular in the study of fractals.

Syntactically, the only difference between terms and literals is that a term may begin with a function symbol whereas a literal starts with a predicate symbol or a negated predicate symbol. If we treat predicate symbols and negated predicate symbols like function symbols, as in [6], then the metrics d_S on terms can be extended to literals. A clause is a finite set of literals, so the metrics we want on clauses are the Hausdorff metrics of the metrics d_S.

4 Terms and Clauses Containing Variables

These pseudometrics were originally devised for clustering of ground terms and clauses. A referee pointed out that they behave oddly when applied to terms containing variables; for instance, if x and y are variables then

$$d_S(p(x,x),p(x,y)) = 0$$

for *every* size function S. It appears that they may have limited value if one attempts to cluster non-ground clauses. However, there is a solution: one can assign weights to variables, just as one does to constants. Formally, this can be justified, without rewriting all the propositions and proofs, by the following trick.

> Say L is the language in which the clauses are expressed. There is an associated *meta-language* $\mathcal{M}L$ in which x and y, variables of L, are *constants*. (The construction of $\mathcal{M}L$ is too elaborate to describe here, although it is straightforward: see [4].) $\mathcal{M}L$ is another first order language, so all the above constructions can be performed in it.

Thus, if one assigns weights to variables of L, then the functions d_S are still metrics, and they detect the difference between $p(x,x)$ and $p(x,y)$.

Inductive logic programming (ILP) constructs non-ground clauses in which the names of variables are chosen arbitrarily, as long as they are distinct within any one clause. Unless one takes care, this could disrupt the clustering process.

Example 5. The two clauses

$$p(x,y,z) \Rightarrow q(x,y)$$
$$p(x,z,y) \Rightarrow q(x,z)$$

are logically equivalent, but there could be a strictly positive distance between them (in the Hausdorff metric of a suitable meta-level d_S); and they might have different distances from a third clause, such as

$$p(x,y,z) \Rightarrow r(x,y).$$

There is a simple solution to this phenomenon. If C and D are any two terms (or clauses), say

$$A_C = \{C\vartheta \mid \vartheta \text{ is an } \textit{invertable} \text{ substitution}\}$$

so, for instance, when C is $p(x,y,z) \Rightarrow q(x,y)$ then A_C contains

$$p(x,y,z) \Rightarrow q(x,y) \quad \text{and} \quad p(x,z,y) \Rightarrow q(x,z) \quad \text{and} \quad p(z,x,y) \Rightarrow q(z,x).$$

Define A_D similarly; and then let the distance $d(C,D)$ be

$$\max \begin{cases} \max_{x \in A_C} \min_{y \in A_D} d_S(x,y) \\ \max_{y \in A_D} \min_{x \in A_C} d_S(x,y) \end{cases}$$

If all variables have the same weight then, although the sets A_C and A_D are infinite, this is well defined and calculable, and it is a pseudometric, and it is invariant under renaming of variables, and it assigns a strictly positive distance between $p(x,x)$ and $p(x,y)$.

5 Applications and Discussion

The metric most commonly used in machine learning is the Hamming distance. This serves well when data are described by attributes with disjoint discrete sets of possible values. It fails to reveal all detail when two different attributes can have the same value. It is also not so useful when each datum is described by an elaborate term in which a subterm can occur twice in different positions.

Example 6. Vere [7] showed how one may construe ground clauses from an instance of apparent causes and their effect. Plotkin [6] showed how one can generalise from a set of similar ground clauses. (See [3] for details.) It is essential to Vere's method that subterms are repeated.

One problem with this process in real situations is that often there is a plethora of possible ground clauses which could be construed and generalised from. In any one case, Vere's method is likely to form ground clauses with irrelevant features. Each example situation will contribute one clause including precisely the relevant features, and several more with irrelevant ones too. If a learner forms clusters in the set of all such ground clauses, using the metrics described here, then those clauses with only attributes common to all situations will form a largest cluster. (There may be several clusters with the maximal number of members, but among them, just one cluster will contain longest clauses.) This is the cluster from which one should generalise.

Example 7. Suppose that a learner should discover heuristic rules to guide theorem proving by natural deduction. A proof is usually found by working back from the target theorem. The hard step is application of the $\Rightarrow E$ rule:

$$\frac{S : p \Rightarrow q \quad S : p}{S : q}$$

When this is used, the root sequent $S : q$ is given and one must choose a suitable p. The key is to choose p so that both p and $p \Rightarrow q$ are provable. This involves finding a p which matches two or more suitable sub-formulas in S.

For a task like this, a Hamming distance would not find suitable clusters of examples from which one could generalise and form useful heuristic rules. A metric based on subsumption, like the ones described above, might serve better.

When extended with weights for variables, the metrics d_S also have potential application to ILP. In the present state of the art, as represented for instance by RIBL [1], an ILP system must overcome noise and redundant features by forming clusters of training instances. Each training instance is a set of literals. The particular similarity measure used for clustering in RIBL depends on a data base in which it finds sets of literals with common terms, rather like those which Vere assembles into clauses.

The similarity measures currently used show scope for improvement. Ideally, similarity (or difference) should be measured by some function which has

a simple general definition and convenient properties, such as a metric. The particular measure encoded into RIBL depends not just on the two examples being compared but also on the data base, which adds complication. If it were replaced with a metric such as some d_S then the resulting program might be more amenable to theoretical analysis and comparisons. It could also incorporate a clustering algorithm such as one of those discussed by Allan Gordon in [2]. RIBL extrapolates by a weighted voting system, based on k nearest neighbours. If instead it extrapolated from a cluster, found by standard methods, then there would be no need for a choice of k. Clustering is still to some extent a black art, because nobody has yet come up with an acceptable formal definition of what is a cluster, but some good work has been done and we may as well take advantage of it.

The metrics d_S are not suitable for all the requirements of RIBL because they are not designed to handle continuous attributes. If two constants a and b are real numbers, then we would prefer $d_S(a, b)$ to be dependent on the difference between a and b, rather than $wt_a + wt_b$. More generally, if terms and clauses are constructed from constants a and b in some metric space (X, δ), then one would like the d_S-distance to depend on $\delta(a, b)$. I hope that the metrics d_S can be so extended.

References

1. W. Emde and D. Wettschereck. Relational instance-based learning. In L.Saitta, editor, *ICML-96*, pages 122–130, Bari, Italy, 1996. Morgan Kaufmann.
2. A. D. Gordon. Hierarchical classification. In P.Arabie, L.J.Hubert, and G. De Soete, editors, *Clustering and Classification*, pages 65–121. World Scientific, 1996.
3. A. Hutchinson. *Algorithmic Learning.* Oxford University Press, 1994, 1995.
4. A. Hutchinson. First order meta theories. *Logic Journal of the IGPL*, 5(1):96–144, 1997.
5. J. L. Kelley. *General Topology.* Springer-Verlag, 1955.
6. G. D. Plotkin. A note on inductive generalization. In B.Meltzer and D.Michie, editors, *Machine Intelligence 5*, pages 153–163. Edinburgh University Press, 1969.
7. S. A. Vere. Induction of relational productions in the presence of background information. In *IJCAI*, volume 1, pages 349–355, 1977. Cambridge, MA.