When Redundancy Matters: Machine Teaching of Representations

Cesar Ferri[1], Dario Garigliotti[2*], Jose Hernandez-Orallo[1],
Brigt Håvardstun[2], Jan Arne Telle[2]

[1*]VRAIN, Universitat Politècnica de València, València, Spain.
[2]Department of Informatics, University of Bergen, Bergen, Norway.

*Corresponding author(s). E-mail(s): dario.garigliotti@uib.no;

Abstract

In traditional machine teaching, a teacher needs to teach a concept to a learner by means of a finite set of examples, the witness set. But concepts can have many equivalent representations. This redundancy strongly affects the search space, to the extent that teacher and learner may not be able to easily determine the equivalence class of each representation. In this common situation, instead of teaching concepts, we explore the idea of teaching representations. We work with several teaching schemas that exploit representation and witness *size* (Eager, Greedy and Optimal) and analyze the gains in teaching effectiveness, both theoretically, and also experimentally for languages where redundancy can vary (DNF expressions and Turing-complete P3 programs). Our theoretical and experimental results indicate that there are various types of redundancy, related e.g. to the *spread* of the redundant representations, handled better by the new Greedy schema introduced here than by the Eager schema. For P3 programs we found that witness sets are usually smaller than the programs they identify, which is an illuminating justification of why machine teaching from examples makes sense at all.

Keywords: Machine Teaching, Concept Representations, Redundancy

1 Introduction

In formal models of *machine learning* (ML), there is a concept class C of possible hypotheses, an unknown target concept $c^* \in C$ and training data given by correctly labelled random examples. In formal models of *machine teaching* (MT), a collection of labelled examples, also called a witness set w, is instead carefully chosen by a teacher T so the learner L can identify c^* in C, thus $w = T(c^*)$ and $c^* = L(w)$, see e.g. Goldman and Kearns (1995). Recently, machine teaching has been applied in various fields like pedagogy (Shafto et al., 2014), trustworthy AI (Zhu et al., 2018), reinforcement learning (Zhang et al., 2021), active learning (Wang et al., 2021) and explainable AI (Håvardstun et al., 2023; Yang et al., 2021). Most theoretical models of MT (and of ML) assume that the class C is a set of different concepts; the goal is then to identify or approximate the target concept in the class. In MT, this assumption is broken when concepts are represented in a language that can have several expressions for the same concept, such as "$|x_3| \cdot |x_5| > 0$" and "$x_3 \neq 0 \wedge x_5 \neq 0$". *A teaching model can focus on teaching the first expression while not realising that the second –equivalent– expression may be easier to teach.*

Concepts are actually expressed in a representational language, and most languages have redundancy, with several representations mapping to the same concept; this is a common kind of *language bias*. For some languages, teacher and learner may not know whether two representations are equivalent —it may not even be computable. Hence, we may end up teaching more than one representation of a concept. For example, the 'powers of 2', the 'number of subsets of an n-element set', and the 'number of n-bit binary strings', all denote the same subset of natural numbers. In many practical situations, representations rather than concepts matters, e.g. prompts for answers (Fernando et al., 2023) or programs for behaviours (Finnie-Ansley et al., 2022).

Having redundant representations is common in most languages with which humans and machines represent concepts, from natural languages to artificial neural networks. Redundancy has been vindicated as a way to accomplish resilience - see e.g. Vardi (2022) referring to von Neumann: "Resilience via redundancy is one of the great principles of computer science, which deserves more attention!" - by analysing kinds and levels of redundancy. Hence, studying redundancy, in particular how the relation between representations and concepts affects teaching, has important implications for and beyond teaching. This holds also for machine learning (see, e.g., Hadley and Cardei (1999)), where notions of syntactic, semantic and search bias (any language bias) are explored to make one language more suitable for learning than another (see, e.g., Muggleton and De Raedt (1994), Whigham (1996), and Nédellec et al. (1996)). The issue is important in any area where there is a non-injective correspondence from concepts to representations, such as explainable AI. Several notions of redundancy, density or compactness in representations have been studied (Alpuente et al., 2010; Enflo et al., 1994; Yu, 1988), yet without focus on quantifying redundancy or concentration over equivalence classes, as we define them in this paper.

We explore the idea of teaching *representations*, where the teacher knows a representation and wants the learner to identify it. Different representations r_1 and r_2 can be in the same equivalence class, a concept c, denoted by $[r_1] = [r_2] = c$. We model this much as in the traditional *concept* teaching scenario, but now by a set of representations R expressed in some language, and the teacher $T : R \to W$ being a partial injective mapping of representations to witnesses. We do not require the mapping to be total to accommodate, e.g., the case $|R| > |W|$. As in the classical concept teaching setting, we assume the learner is able to check the consistency between a representation and a given witness.

We assume the learner has simplicity functions on the set of representations and on witnesses. A similar assumption appears in various MT models, e.g., the Preference-Based teaching model assumes an ordering on concepts and the Recursive teaching model assumes a partial ordering on witnesses. Furthermore, in the teacher/learner algorithm of Telle et al. (2019), which we call *Eager*, the learner employs an ordering on concepts called the learning prior, while the teacher wants to minimize the 'teaching size,' which implicitly determines an order on witnesses. In the Eager algorithm, the learner, when given a witness, will employ Occam's razor and guess the simplest consistent representation, and the teacher will always provide the smallest witness that suffices to learn the target representation. Note this implies the learner will learn at most one representation per concept, namely the simplest one. But *the main issue is that if the teacher does not know that r_1 and r_2 are equivalent, and r_1 is simpler than r_2, then the teacher will be looking for a witness set for r_2 that distinguishes it from r_1, something the teacher will never find.* We assume that there is a simplicity function over R that allows the teacher to determine whether a representation is simpler than other, and to use it to choose a representation that is easier to teach. In this work, we operationalize this function by approaching it with a natural size function $s(r)$ defined in terms of the syntax in which a representation r is expressed (e.g. see the respective size functions when defining $\prec_R$ in Sections 4.1 and 4.2).

When teaching a target it is quite common to give the smallest observation that suffices to learn it, and the learner could actually expect this from the teacher, and thus rule out representations that could have been taught with a witness smaller than what was given. Herein lies the simple idea behind the slight change to Eager into what we call the *Greedy* algorithm. The teacher uses the smallest consistent witness to teach the simplest representation, removes this witness and representation from further consideration, and repeats. Likewise, the learner, when given a witness, will know the teacher behaves in the above way. (See Section 2 for the exact protocol formalizations.)

We also introduce another protocol, *Optimal*, being equal to the graph-theoretical best possible under the minimal constraint that a representation must be taught by a consistent witness and that two distinct representations must be taught by two distinct witnesses. The Section on Formal Comparisons shows several formal results about the strengths of Eager, Greedy and Optimal algorithms. In particular, on concepts taught by both Eager and Greedy, the latter never uses a larger witness, and often a smaller witness. However, Theorem 3 shows that for any concept class, adding redundant representations with small spread in the order, can level out the difference between Eager and Greedy.

In our experiments with these three algorithms, on several representation languages with various kinds of redundancy, the main finding is that *while Greedy needs more witness sets to teach all representations, it uses them more effectively to also teach more concepts, and it does so while teaching several common concepts by a smaller witness.* The experiments were further chosen to answer this question: What characterizes a language where Greedy teaches only a few common concepts by a smaller witness than Eager? We show that this is not directly related to the *amount* of redundancy, but rather to the *spread* of the redundant representations in the order (see Fig. 3). Also, for P3 programs, the witness sets are usually smaller than the programs they identify (see Fig. 4).

Eager and Greedy are two extremes (one teaching a single representation per concept, the other teaching all representations) in a spectrum of approaches that can shed light to important phenomena such as the effect of redundancy in inductive search, the actual bias in size-related priors and the relevance of syntax over semantics in explanations.

2 Machine Teaching definitions

Various models of machine teaching have been proposed, e.g., the classical teaching dimension model (Goldman and Kearns, 1995), the optimal teacher (Balbach, 2008), recursive teaching (Zilles et al., 2011), preference-based teaching (Gao et al., 2017), or no-clash teaching (Fallat et al., 2023). These models differ mainly in the restrictions they impose on the learner and the teacher to avoid collusion. The common goal is to keep the *teaching dimension*, i.e., the size of the largest teaching set, $\max_{c \in C} |T(c)|$, as small as possible.

In all these models, C is a set of concepts, with each $c \in C$ a subset of a domain X. We consider more than one *representation* for the same concept, thus with a set R of representations constituting a multiset of the concepts C that are the subsets of X. The teacher $T : R \to W$ is now a partial injective mapping from representation $r \in R$ to a set of observations $w \in W$ (often $W \subseteq 2^{X \times \{0,1\}}$ so that w is a set of negatively or positively labelled examples from X) and the learner $L : W \to R$ is a partial mapping in the opposite direction. As usual, teacher and learner share the consistency graph G_R on vertex set $R \cup W$ with a representation $r \in R$ adjacent to a witness $w \in W$ (and thus $rw \in E(G_R)$ an edge) if r and w are consistent, i.e., with positive (resp. negative) examples in w being members (resp. non-members) of r. Naturally, r must be consistent with $T(r)$ and $L(w)$ must be consistent with w, and a successful teacher-learner pair must have $L(T(r_1)) = r_2$ such that $[r_1] = [r_2]$. In what follows, we assume L and T use the same representation language and we impose that $L(T(r)) = r$.

From a graph theoretical view, the teacher and learner mappings are matchings in G_R between representations and witnesses. Two vertices are called twins if they have the same set of neighbors. Usually there is a bijection between the equivalence classes of twins of R in G_R, that we denote by R/W, and the set of concepts C. However, if the witness set W is too sparse then R/W may be a coarsening of C.

We will be comparing two models for teaching representations, that we call Eager and Greedy. Both assume some natural size functions on representations R and on witnesses W, and use these to arrive at two total orderings $\prec_R$ on R and $\prec_W$ on W. For example, if the representations in R are expressed in some description language, which can be English, or a programming language, or some set of mathematical formulas, then a representation consists of finite strings of symbols drawn from some fixed alphabet Σ, given with a total order which will be used to derive a lexicographic order on strings over this alphabet, with shorter strings always smaller, sometimes called shortlex. Analogously, we can define a total ordering on witnesses.

Thus, we call the input to these algorithms an ordered consistency graph denoted by the 3-tuple $(G_R, \prec_R, \prec_W)$, and by a vertex (or representation/witness) appearing earlier than another we mean in these orderings. In the Eager model we have $L(w) = r$ for r being the earliest representation (in the order $\prec_R$) consistent with w. The teacher, knowing that the learner behaves in this way, will construct the mapping $T : R \to W$ iteratively as follows: go through W in the order of $\prec_W$, and for a given witness w find the earliest $r \in R$ with $rw \in E(G_R)$, and if $T(r)$ is not yet defined then set $T(r) = w$ else continue with next witness. Telle et al. (2019) use the Eager model to teach programs in the Turing-complete language P3, by witnesses containing specified I/O-pairs. Many P3 programs have equivalent I/O-specifications, and we can observe that in the Eager model only the earliest representation (program) of any concept (I/O-specification) is taught.

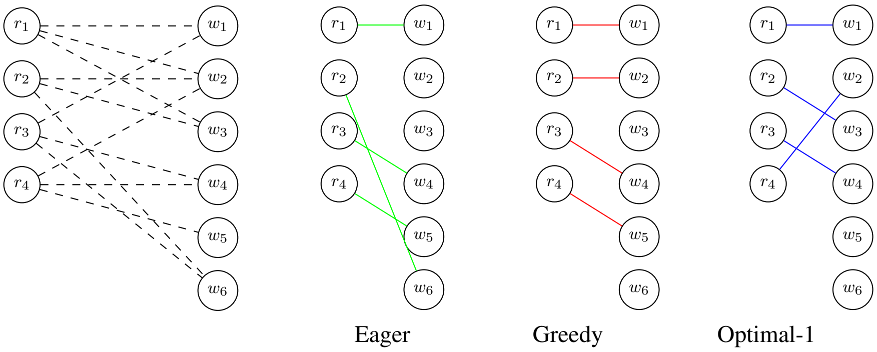

Fig. 1 Simplest possible ordered consistency graph ($\prec_R$ and $\prec_W$ given by the indices; witness w_i has size i) showing that Eager, Greedy and Optimal-1 can require different max teaching sizes (witnesses of size 6, 5 and 4 respectively).

In order to teach all representations we introduce the Greedy model, where we make a slight change to the way the teacher constructs its mapping, as follows: go through W in the order of $\prec_W$, and for a given witness w find the earliest $r \in R$ with $rw \in E(G_R)$ such that $T(r)$ is not yet defined, then set $T(r) = w$ and continue with next witness (if no such r exists then drop this w). Teaching and Learning in the Greedy model is thus done following an order, similar to what happens in the Recursive teaching model of Zilles et al. (2011) (teach sequentially and remove concepts that have already been taught), using also an order on the representations, similar to what is done in the Preference-Based teaching model of Gao et al. (2017) (return the most preferred remaining concept consistent with a given witness).

In classical machine teaching, the goal is to minimize the teaching dimension of a concept class, i.e. to find a legal teacher mapping $T : C \rightarrow W$ where the maximum of $|T(c)|$ over all $c \in C$ is minimized. When there is a total mapping $T : R \rightarrow W$ we do the same for teaching representations, except we use $size(T(c))$ rather than cardinality, as in Optimal-1 defined below. However, a total mapping may not exist, and in that case we in this paper define an alternative measure, as in Optimal-2 below, being the maximum number of representations covered when constrained to partial teacher mappings covering the maximum number of concepts. We thus define the Optimal model in two flavors.

Optimal-1 is the minimum max size witness over all teacher mappings, with the sole restriction that the teacher mapping is injective and assigns a consistent witness to every representation. This value can be computed in polynomial time by a binary search for the smallest k s.t. the induced subgraph G_R^k of G_R, on R and all witnesses of size at most k, has a matching saturating R i.e. containing all of R.

As Optimal-1 is undefined when G_R has no matching saturating R, we define *Optimal-2* as the maximum number of representations covered by any matching that maximizes the number of concepts covered. It is not clear that this number is computable in polynomial time, but for most graphs in the experimental section we could compute it.

3 Formal comparisons

We show several results comparing the performance of the three algorithms Eager, Greedy and Optimal, starting with Fig. 1 giving the simplest possible consistency graph where Eager, Greedy and Optimal-1 behave differently.

The teacher mapping $T : R \rightarrow W$ for Greedy, as defined in the previous section, is computed by an iterative procedure whose outer loop follows the witness ordering (and inner loop representation ordering). Let us call it $T_W : R \rightarrow W$. Consider the alternative $T_R : R \rightarrow W$ which switches these two: go through R in the order of $\prec_R$, and for a given representation r find the earliest $w \in W$ with $rw \in E(G_R)$ such that $T_R^{-1}(w)$ is not yet defined, then set $T_R(r) = w$ and continue with the next representation (if no such w exists then drop this r).

Theorem 1. *The teacher mappings returned by Greedy following $\prec_W$ and the alternative following $\prec_R$ are the same.*

Proof. We prove, by induction on $\prec_W$, the statement (*): "for any $w \in W$ if $T_W(r) = w$ then also $T_R(r) = w$." Let w_1 be the earliest witness. We have $T_W(r') = w_1$ and when assigning T_R clearly w_1 is the earliest neighbor of r' with $T_R^{-1}(w_1)$ undefined so that $T_R(r') = w_1$. For the inductive step, we assume the statement (*) for all witnesses earlier than w, and assume that $T_W(r) = w$. By the inductive assumption (*) we know that no witness earlier than w will by T_R be assigned to r. This means that when assigning $T_R(r)$ then w is the earliest neighbor of r with $T_R^{-1}(w)$ undefined, and thus we have $T_R(r) = w$ as desired. □

Since the Greedy teaching is only a slight change to the Eager teaching, it is easy to see that for any $r \in R$, if Eager assigns $T_E(r) = w_E$ then Greedy will also assign some $T_G(r) = w_G$, and in the order $\prec_W$ we could have w_G earlier than w_E, but not the other way around.

Theorem 2. *On representations/concepts taught by Eager, Greedy will never use a larger witness.*

Proof. This since $T_E(r) = w_E$ implies that r is the earliest representation consistent with w_E, so when the Greedy teacher mapping is computed then at reaching w_E the first representation tried is r, and if $T_G(r)$ has already been defined it will be for an earlier and thus no larger witness. □

We show that for any concept class we can add redundant copies of representations consecutive in $\prec_R$ to make Eager and Greedy perform almost identical.

Theorem 3. *For any concept class R (i.e. each concept has a unique representation) on witness set W and orders $\prec_R, \prec_W$, we can add less than a total of $|W|$ copies of representations to get R', and make $\prec_{R'}$ an extension of $\prec_R$ with all copies consecutive, such that Eager teaches the same on $(G_R, \prec_R, \prec_W)$ as on $(G_{R'}, \prec_{R'}, \prec_W)$ and using the same witnesses as Greedy on $(G_{R'}, \prec_{R'}, \prec_W)$ on all common concepts.*

Proof. Consider the ordered consistency graph $(G_R, \prec_R, \prec_W)$ and for each representation $r \in R$ compute $f_r = |\{w \in W : rw \in E(G_R) \wedge (r'w \in E(G_R) \Rightarrow r \prec_R r')\}|$, i.e. the number of witnesses for which r is the earliest neighbor. Let R' be the representation class corresponding to the ordered consistency graph $(G_{R'}, \prec_{R'}, \prec_W)$ where for each $r \in R$ we add $f_r - 1$ copies of r and extend the order $\prec_R$ to $\prec_{R'}$ by putting all the copies consecutively

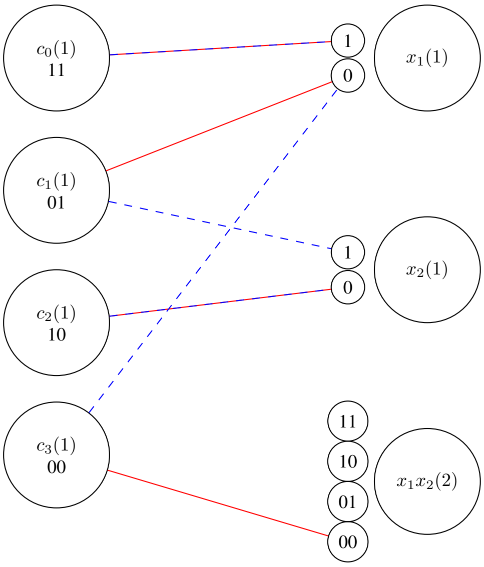

Fig. 2 Comparison of Greedy versus Optimal, when size is cardinality.

right after r, so there are f_r consecutive copies total of r. Note that, as the sum of f_r over all $r \in R$ is $|W|$ (each witness has a single earliest r) we add less than $|W|$ new representations.

For each $w \in W$ define r_w to be the earliest neighbor of w in G_R and let r'_w be the earliest neighbor in $G_{R'}$. For simplicity, when we say $G_R, G_{R'}$ we mean the ordered consistency graphs. We prove the following loop invariant on any witness $w \in W$ by induction over $\prec_W$: "knowing r_w has f_{r_w} witnesses for which it is the earliest, let w be the kth such witness. Both on G_R and $G_{R'}$, if $k = 1$ then Eager will assign r_w to w, and if $k > 1$ Eager will not use w. Greedy on $G_{R'}$ will assign r to w for r being the kth copy of r_w in the $\prec_{R'}$ order." The loop invariant is trivially true for the first witness. For the inductive step, we know r_w is the earliest neighbor of w and that w shares this property with f_{r_w} witnesses. If w is the first of these witnesses, then it is clear that both Eager and Greedy will assign r_w to w, as the loop invariant tells us no earlier witness has been assigned to r_w. If w is the kth copy of r_w for $k \geq 2$ then Eager will not use w, neither on G_R nor $G_{R'}$, while for Greedy on $G_{R'}$ as there are at least k copies of r_w all in consecutive order then Greedy will assign w to the kth copy as it must be available by the loop invariant. We have thus proven the loop invariant, and this implies the statement of the Theorem as it encompasses all representations taught by both algorithms. □

Greedy can be worse than Optimal, when size is cardinality.

In Fig. 2, there are 4 concepts c_0, c_1, c_2, c_3 and 2 ground elements x_1, x_2. Witnesses consist of labelled examples and size is just cardinality (the number of examples in a witness). There are thus 4 witnesses of size 1, and the optimal mapping given by the blue dotted lines uses these 4, but Greedy assigns the red mapping, which uses a witness of cardinality two.

3.1 Optimal-1 can be more than the trivial bound, for size being cardinality

In this subsection we assume the size of a witness is its cardinality (number of examples), as in teaching dimension.

Note that for set of representations R and ground domain X, defining $triv_{unl}(R)$ to be the smallest value k such that $\sum_{i=0}^{k} \binom{|X|}{i} \geq |R|$ then for witnesses on unlabelled examples Optimal-1 must use a witness of cardinality (size) $triv_{unl}(R)$, since the sum is the number of witnesses on unlabelled examples of cardinality up to $triv_{unl}(R)$, and we need that many witnesses to cover R. However, there are concept classes where we need more, e.g. take $X = \{1, 2, 3, 4\}$ and $R = \{\emptyset, 1, 2, 3, 4, 12, 14, 24, 124\}$. Note $|R| = 9$, $|X| = 4$ and thus $triv_{unl}(R) = 2$. In the consistency graph limited to witnesses of cardinality at most 2 we have a vertex for each of the 9 $r \in R$ and 11 vertices corresponding to all subsets of X of size at most 2, but 3 of these latter vertices, corresponding to $\{13, 23, 34\}$, have no neighbors in R and thus no matching saturating R can exist.

Similarly, for witnesses on labelled examples, define $triv_{lab}(R)$ to be the smallest value k such that $\sum_{i=0}^{k} \binom{|X|}{i} * 2^i \geq |R|$. We show a case where Optimal-1 must use a witness of cardinality (size) larger than $triv_{lab}(R)$. Assume a set of concepts R and ground elements $X = \{a, b, c, d, e, f, g\}$. Let $r_1 = \{a, b\}$ and let $r_2 = \{b, c\}$. Now, from the subset $\{d, e, f, g\}$ select 13 unique subsets, and let these be $\{r_3, r_4, ..., r_{15}\}$. By letting $R = \{r_1, r_2, ..., r_{15}\}$ we have $triv_{lab}(R) = 1$. However, the only representations consistent with the witness sets, of cardinality one, on the positive examples $(a, 1), (b, 1), (c, 1)$ are r_1 and r_2. Hence at most two of these three witness sets can be used to teach a representation. With 15 representations, and 15 witness sets of size ≤ 1, and at least one unused witness set, we know we will have to use a witness set of cardinality ≥ 2.

3.2 Greedy can be arbitrarily worse than Optimal-1

We now know that Eager can never be better than Greedy, and that Optimal-1 sometimes leaves small witnesses unused. In the experimental section, we will see that Greedy is often close to Optimal-1 , but we now show that there are also cases where Greedy is far worse than Optimal-1 , in that the mapping it finds will use a witness of larger maximum size.

The most restricted case is when the size of a witness is equal to its cardinality, i.e. equal to the number of examples it contains, so-called teaching dimension, and even then Greedy can do worse than Optimal-1 , as we show in the Appendix. In this restricted case we are not able to show that it can do worse by an arbitrary amount, and must leave this as an open question. However, for some less restricted size functions, when we allow the size of ground elements to vary, we can show the following.

Theorem 4. *When there is a bound on the number of witnesses of any size, then for any t, there exists a size ordered consistency graph where Greedy uses a witness of size G_{max} while Optimal-1 will not use a witness of size larger than O_{max} and where $G_{max} - O_{max} \geq t$.*

Proof. We prove it by construction, for a graph where we have only one representation for each concept. Assume there are at most s witnesses of any size. Let concept ordering be $c_1, c_2, ...c_k$, and let witness ordering be $w_1, w_2, w_3, ...w_{st+k}, ...$ with $s(w_i) \leq s(w_{i+1})$, for any choice of k. Let c_1 be consistent with w_1 and w_2, and let c_2 be consistent with w_1 and

w_{st+k} (but not consistent with any witness in between, which could happen if e.g. there are that many ground elements) and let c_i for $3 \leq i \leq k$ be consistent with $w_i, w_{i+1}, ..., w_{i+st}$. The Optimal matching will assign c_1 to w_2 and c_2 to w_1 and c_i to w_i for any $3 \leq i \leq k$, thus using max size witness with index k. Greedy will assign almost the same matching except it starts with c_1 to w_1 and then c_2 to w_{st+k} since this is the only one available, thus using max size witness with index $st + k$. As we have at most s witnesses of each size, then $s(w_{st+k}) - s(w_k) \geq t$. □

4 Experiments

Table 1 Domain features: 'Witness' is the constraint about the size of each witness set. For P3 we do not know G_R nor C. (*For small-P3 we do not know $|C|$, so we show $|R/W|$.)

Domain	$\lvert R\rvert$	$\lvert C\rvert$	Witness	$\lvert W\rvert$	Redundancy	Redund. Spread
3-DNF	256	256	$\lvert w\rvert \leq 5$	3,488	0	15.13
3-Term DNF	2,952	246	$\lvert w\rvert \leq 5$	3,488	0.727	13.28
3-Term DNF with permutations	79,158	246	$\lvert w\rvert \leq 5$	3,488	0.9896	11.41
			$\lvert w\rvert = 5$	1,792	0.9896	7.04
3-Term DNF with permuts. and duplicates	158,316	246	$\lvert w\rvert \leq 5$	3,488	0.9948	10.42
P3	$1.9 * 10^9$	?	$s(w) \leq 6$	6,548	?	?
small-P3	1,267	106*	$s(w) \leq 4$	260	0.331	2.97

In this section we observe quantitatively how each teaching protocol behaves in a variety of languages with various types of redundancy. Table 1 summarizes the key features.

We first describe the metrics used, and then the languages. To estimate how redundant a language is, we define *Redundancy* as $1 - Uniqueness$, where $Uniqueness$ is computed as the average for all concepts of $1/|R_c|$, with R_c the set of all equivalent representations for concept c. Formally:

$$\text{Redundancy} = 1 - \frac{1}{|C|} \sum_{c \in C} \frac{1}{|R_c|} \ . \tag{1}$$

From Theorem 3 we suspect that Eager performs well compared to Greedy when, for each witness w, in the set R^i_w of the i earliest representations consistent with w there are few different concepts in R^i_w. Assuming the $\prec_W$ ordering is $w_1, w_2, ...$ we define *Redundancy Spread* as the average number of different concepts in $R^i_{w_i}$. We use $R^i_{w_i}$ for w_i because at most i values of $T : R \rightarrow W$ have been defined when Greedy or Eager consider w_i. Formally, if for each witness w_i we let $r^j_{w_i}$ be the j-th representation consistent with w_i, and recalling that $[r]$ is the concept equivalence class r belongs to, the definition of Redundancy Spread becomes:

$$\text{Redundancy Spread} = \frac{1}{|W|} \sum_{w_i \in W} \left| \left\{ [r^j_{w_i}] : j \in [1 \ldots i] \right\} \right| \ .$$

4.1 Boolean expressions

We experiment with Boolean expressions, since in this domain it is possible to decide whether two representations belong to the same equivalence class, i.e., correspond to the same concept. Furthermore, in this language, we are able to measure the degree of redundancy present in the set of representations. In all our experiments, we consider the ordered alphabet of 3 Boolean variables $\{a, b, c\}$ and every representation in R is expressed in DNF (Disjunctive Normal Form), i.e. OR of AND-terms. Every variable occurs at most once in a term, possibly preceded by the negation operator $\neg$. For example, $(a \wedge b \wedge \neg c)$ is a valid ground term.

Let us first describe the witnesses, which are similar across all experiments. Each example is a pair consisting of a binary string of length three (the input) and a bit (the output), such as $(001, 1)$ indicating that when $a = 0, b = 0, c = 1$ the value of the function should be 1. There are thus 16 witnesses of cardinality one, and $16 * 14/2 = 112$ witnesses of cardinality two, as we do not allow for self-incongruent witnesses and the order of examples is irrelevant. For these experiments we use witnesses of cardinality up to 5, and only positive examples, which implies $|W| = 3,448$.

$\prec_W$: The witnesses in the collection W are ordered by their increasing teaching size. The *size function* $s : W \to \mathbb{N}$ is defined for a witness w as the number of examples in w times 4 (the number of bits) plus the number of 1s in the inputs of the examples. This scheme may be useful for a situation with many variables, where it could be more efficient to just identify the position of the 1s in a long binary string.

In the first experiment, *3-DNF*, we build a scenario without redundancy. We consider all 256 Boolean functions on three variables as the set $R = C$, as follows. A Boolean function ϕ on a, b, c is uniquely defined by a truth table that gives the truth values for the 8 possible truth assignments to the three variables. For any assignment (v_a, v_b, v_c) to (a, b, c) where $\phi(v_a, v_b, v_c) = 1$ we include the corresponding term in the representation r_ϕ of ϕ in R. For instance, if ϕ is logically equivalent to "a and b" then this is represented as $r_\phi = (a \wedge b \wedge c) \vee (a \wedge b \wedge \neg c)$.

In our second experiment, *3-term DNF*, we introduce redundancy in R, and do this in a way that will shrink the concept space to have smaller cardinality $|C| = 246$. Every representation is now a disjunction of up to 3 ground terms, and each term contains one, two or three distinct variables, each possibly negated. For a single term we have six representations of one variable, $12 = 6 * 4/2$ representations of two variables, and 8 representations for three variables, thus 26 in total. For two terms we have $26 * 25/2 = 325$ representations, and for three terms we have $26 * 25 * 24/3! = 2,600$. We also have a single representation (False) with no terms, and thus $|R| = 1 + 26 + 325 + 2,600 = 2,952$.

For the third experiment, *3-term DNF with permutations*, we want to increase Redundancy and decrease Redundancy Spread. The set of representations is similar to 3-term DNF, but here we allow for any permutation of variables within each term. For example, if $(b \wedge \neg c) \in R$, also $(\neg c \wedge b) \in R$. This drastically increases the cardinality of R, while Redundancy also increases substantially and Redundancy Spread decreases. This third experiment is carried out also against a smaller witness set, with witnesses of cardinality 5 only, to see the effects of a large decrease in Redundancy Spread.

Finally, the last Boolean experiment, *3-term DNF with permutations and duplicates*, is similar to the previous one, 3-term DNF with permutations, but here each representation occurs consecutively twice in the ordered collection R, for yet another variation of redundancy.

$\prec_R$: Representations in R are ordered by increasing size $s(r)$ defined as the number of literals in r plus the number of negated variables plus the number of disjunction symbols; e.g., $s((a \wedge b \wedge c) \vee (a \wedge b \wedge \neg c)) = 8$. Two representations of same size are further ordered by the increasing number of terms, and by the $\langle a, b, c, \neg, \vee \rangle$-induced lexicographic order.

4.2 Universal language

We also analyze the new teaching frameworks for the Turing-complete language P3. P3 is a simple string manipulation language inspired by P", introduced by Corrado Böhm ((Böhm, 1964)), the first GOTO-less imperative language proved Turing-complete, i.e., universal. The most popular variant (Brainfuck) has 8 instructions in total. We employ another version called P3, also universal and having just 7 instructions: <>+-[]o. We consider P3-programs that take binary inputs and generate binary outputs. A representation is a P3 program, such as the following one, which performs the left shift operation (e.g., on input 10010 we get output 00101): >[o>]<[<]>o. Regarding the semantics of P3, we refer to the work by Telle et al. (2019).

A witness is a set of pairs of binary strings defining an input and an output, and we use only positive examples. For example, the witness $w = \{\langle 0100, 00 \rangle, \langle 001, \rangle, \langle 00, 00 \rangle\}$ is consistent with any program that on input 0100 outputs 00, on input 001 has no output and on input 00 outputs 00.

$\prec_W$: We use a simple encoding of example sets (the number of bits) as the size function $s(w)$, and break ties in a deterministic way. The witness w above has size $s(w) = 13$.

$\prec_R$: The lexicographic order on instructions <>+-[]o defines the $\prec_R$ length-lexicographic P3 programs ordering.

We perform two experiments, called P3 and small-P3. In P3 we repeat the experiment from Telle et al. (2019). Note that computing equivalence classes of programs is undecidable in general, and we cannot decide if a program enters an infinite loop. We limit the number of timesteps for the computation of each program before breaking. In P3 experiment, the set of representations/programs R is a potentially infinite set, but Eager and Greedy end up never going beyond the $1.9 * 10^9$ first programs. The witness set W contains all sets of size at most 6. As we cannot compute the consistency graph and run any version of Optimal, we conduct another experiment, *small-P3*.

In small-P3 we start with all self-congruent witnesses of size at most 4, and the first 10,000 programs in $\prec_R$. We filter out every witness without consistent program and every program without consistent witness. A set of 1,267 programs is then R, and a set of 260 witnesses is W. We compute the consistency graph G_R and use R/W as the concept class, i.e., considering two programs equivalent if they are consistent with the exact same subset of W.

4.3 Experimental results

In Table 2 we present the results for the experimental domains described in the previous subsections on Boolean Expressions and Universal Language. Additionally, Table 3 reports

Table 2 Results for the teaching frameworks across all domains. $s(w)$ is the size of a witness w; $i(w)$ is the index of witness w in the order $\prec_W$. (*For Optimal-2 and small-P3, the exact number 'Reprs. taught' is unknown.)

Domain	Witness	Algorithm	Reprs. taught	Concepts taught	Max. $\{s(w)\}$	Max. $\{i(w)\}$
3-DNF	$\|w\| \leq 5$	Eager	219	219	30	3,488
		Greedy	256	256	16	328
		Optimal-1	256	256	16	256
3-Term DNF	$\|w\| \leq 5$	Eager	170	170	30	3,466
		Greedy	2,895	246	30	3,481
		Optimal-1	2,952	246	28	2,952
3-Term DNF with permutations	$\|w\| \leq 5$	Eager	170	170	30	3,466
		Greedy	3,488	189	30	3,488
		Optimal-2	3,488	246	30	3,488
	$\|w\| = 5$	Eager	170	170	30	1,770
		Greedy	1,792	177	30	1,792
		Optimal-2	1,792	246	30	1,792
3-Term DNF with permutations and duplicates	$\|w\| \leq 5$	Eager	170	170	30	3,466
		Greedy	3,488	178	30	3,488
		Optimal-2	3,488	246	30	3,488
P3	$s(w) \leq 6$	Eager	2,032	2,032	6	6,512
		Greedy	6,548	?	6	6,548
small-P3	$s(w) \leq 4$	Eager	53	53	4	202
		Greedy	225	65	4	260
		Optimal-2	$\geq$ 214*	106	4	260

on % of cases where Greedy is able to teach concepts with earlier and simpler witnesses than those used by Eager, and Fig. 3 shows that Redundancy Spread to a large degree explains this behavior.[1]

Table 3 Greedy beating Eager: Common concepts with earlier (lower index) and simpler (smaller size) witness.

Domain	Witness	% Witness Index Lower	% Witness Size Smaller
3-DNF	$\|w\| \leq 5$	94.98%	94.06%
3-term DNF	$\|w\| \leq 5$	97.65%	97.06%
3-Term DNF with permutations	$\|w\| \leq 5$	90.59%	88.24%
	$\|w\| = 5$	40.00%	30.00%
3-Term DNF w/ permuts. and duplicates	$\|w\| \leq 5$	84.12%	81.18%
P3	$s(w) \leq 6$	13.98%	6.55%
small-P3	$s(w) \leq 4$	24.53%	18.87%

[1] The materials to reproduce the experimental results are stored in a downloadable directory shared at https://bit.ly/MLJournal-MTOfReprs-Supplementary .

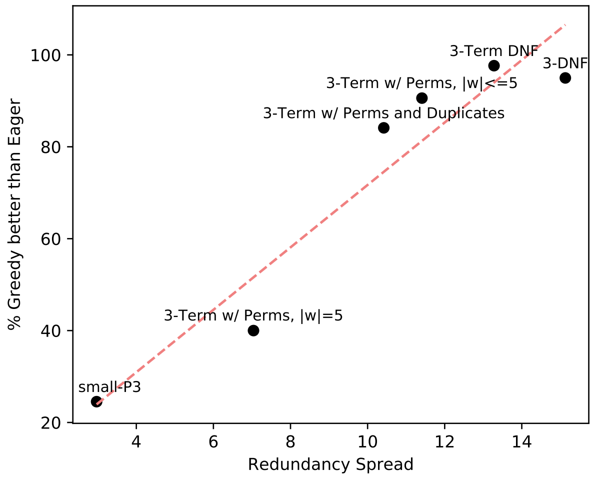

Fig. 3 Relation between *Redundancy Spread* and *% Greedy better than Eager*. Data from Tables 1 and 3. The red line is the best fit linear approximation.

.

As Table 2 shows, for most of the Boolean experiments, Greedy manages to teach as many representations as the amount of witness sets available. We can observe that Greedy is always able to teach more concepts than Eager, a method specialized in teaching unique concepts. Greedy teaches many more concepts and representations often by making use of witnesses that are larger or later in the order. Yet, even in scenarios without redundancy, such as 3-DNF, where Eager might be considered suitable, Greedy still teaches some more concepts, with earlier and smaller witnesses. Greedy performs almost as good as Optimal-1 and Optimal-2 in terms of the number of representations taught, and the largest size and highest index of witnesses used. However, Optimal-2 (used rather than Optimal-1 whenever not all of R can be covered) always covers a higher concept number than Greedy. Figure 3 shows the correlation between the percentage of cases where Greedy outperforms Eager and the Redundancy Spread, and confirms the latter as a strong metric for predicting these performance comparisons.

Analyzing now the universal language domain, we see that Greedy allows to teach over three times more (almost five for small-P3) programs than Eager. For P3 we do not have the consistency graph and thus cannot compute any version of Optimal. Quite remarkably, in Fig. 4 we see that for P3 programs the witness sets of Greedy are usually smaller than the programs they identify. A similar behavior for fewer programs was noted for Eager by Telle et al. (2019). The number of programs taught by the greedy algorithm is significantly higher (see the numbers of programs in the legend). When we focus on the intersection of programs that are taught by the two strategies (Fig. 5), we appreciate how the greedy algorithm is able to teach concepts using witness sets of smaller size.

For small-P3 we use R/W as a proxy for C and note that Optimal-2 is able to cover all these 106 concepts, but seemingly at the price of fewer representations (214 versus 225 for Greedy). However, note that our implementation of Optimal-2 is actually only able to optimize the number of concepts covered. The further maximization of representations that Optimal-2 requires may in fact be NP-hard, but we must leave this as an open problem. On

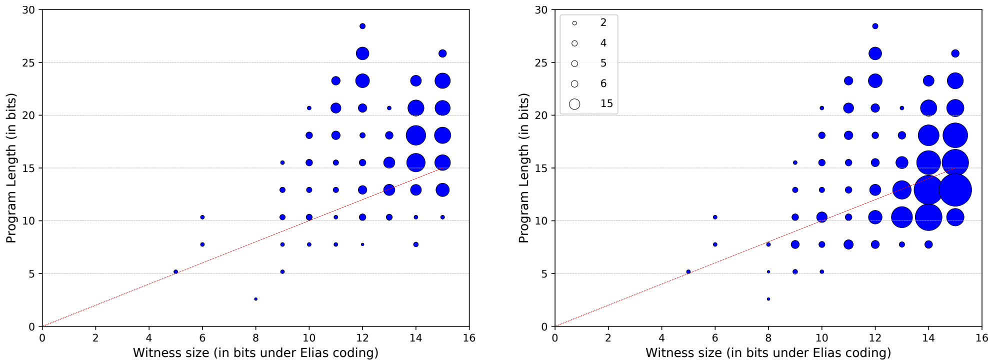

Fig. 4 Eager (left) versus Greedy (right): Program length versus witness size, using Elias coding (Elias, 1975). Circles above the unit diagonal denote witness smaller than program, with size of circle corresponding to the number of programs.

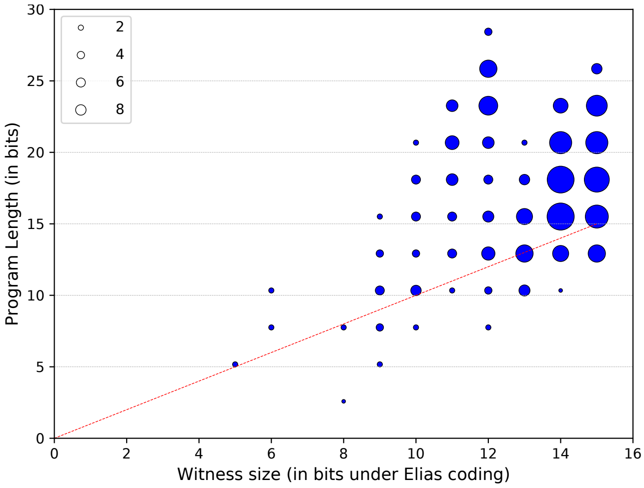

Fig. 5 Program length versus witness size, using Elias coding (Elias, 1975), for the programs taught by both Eager and Greedy, such that $s(Elias(w)) \leq 15$. Circles above the unit diagonal denote witness smaller than program, with size of circle corresponding to the number of programs.

concepts common to Eager and Greedy, the latter finds smaller witnesses on a relatively low percentage of the total for Small-P3 (see Fig. 3) but even more so for P3. Since for P3 there is no consistency graph, we cannot compute Redundancy Spread, but we hypothesize that the small values of 13.98% and 6.55%, still in favor of Greedy, are due to an even lower Redundancy Spread of its consistency graph.

5 Discussion

The notion of redundancy we explore is very old, dating back to numeral systems (e.g., IIII and IV being two alternative representations of the number 4 in Roman numerals). Leibniz was interested in prime decomposition as a tool for his *characteristica universalis* in which the representations and concepts could have a bijective mapping. In numeration systems, this is possible. Indeed, it was Böhm himself (the 'father' of P3) who proved that every natural number has a unique representation in bijective base-k ($k \geq 1$) (Böhm, 1964). However, we also know that in many other languages, such as Turing-complete languages, this is not possible. And even for some other formal languages for which the equivalence classes can be calculated effectively, using a canonical representative for each concept would lead to languages that are very cryptic, such as a numbering of an enumeration —a coding— of all the equivalence classes. This is the first time, to our knowledge, that the problem of teaching has been formally studied without assuming the bijection of representations and concepts, or even assuming that teacher and learner can calculate the equivalence classes.

The more realistic view of teaching adopted in this paper modifies some key elements of the traditional machine teaching setting and gives more relevance to the representation language. Given the high predictiveness for some teaching indicators, the new metric of redundancy spread, introduced in this paper, could be used to anticipate what language representations are better than others for teaching and learning. The size of representations becomes a natural way of imposing an order on them, which can be used in the teaching protocols. Some of the protocols seen here, such as Greedy, are computationally expensive. However, none of them requires the calculation of equivalence classes, which may also be expensive –or undecidable. The key finding is that *Greedy, a protocol designed to teach all representations, ends up teaching more concepts than Eager, which aims at only teaching concepts.* Our results reveal a wide spectrum between these two protocols, and future work could explore other teaching protocols for representations that are somewhat in between, or have better theoretical or statistical properties.

Declarations

Funding.

This work was funded by the Norwegian Research Council grant 329745 "Machine Teaching for Explainable AI," CIPROM/2022/6 (FASSLOW) funded by Generalitat Valenciana, the EC H2020-EU grant agreement No. 952215 (TAILOR), and Spanish grant PID2021-122830OB-C42 (SFERA) funded by MCIN / AEI / 10.13039 / 501100011033 and "ERDF A way of making Europe."

Data availability.

The materials to reproduce the experimental results are stored in a downloadable directory shared at https://bit.ly/MLJournal-MTOfReprs-Supplementary .

References

Alpuente, M., Comini, M., Escobar, S., Falaschi, M., and Iborra, J. (2010). A compact fixpoint semantics for term rewriting systems. *Theoretical Computer Science*, 411(37):3348–3371.

Balbach, F. J. (2008). Measuring teachability using variants of the teaching dimension. *Theoretical Computer Science*, 397(1-3):94–113.

Böhm, C. (1964). On a family of Turing machines and the related programming language. *ICC Bulletin*, 3(3):187–194.

Elias, P. (1975). Universal codeword sets and representations of the integers. *IEEE Trans. Inf. Theory*, 21(2):194–203.

Enflo, P., Granville, A., Shallit, J., and Yu, S. (1994). On sparse languages l such that ll= σ. *Discrete Applied Mathematics*, 52(3):275–285.

Fallat, S., Kirkpatrick, D., Simon, H. U., Soltani, A., and Zilles, S. (2023). On batch teaching without collusion. *Journal of Machine Learning Research*, 24:1–33.

Fernando, C., Banarse, D., Michalewski, H., Osindero, S., and Rocktäschel, T. (2023). Promptbreeder: Self-referential self-improvement via prompt evolution. *CoRR*, abs/2309.16797.

Finnie-Ansley, J., Denny, P., Becker, B. A., Luxton-Reilly, A., and Prather, J. (2022). The Robots Are Coming: Exploring the Implications of OpenAI Codex on Introductory Programming. In Sheard, J. and Denny, P., editors, *ACE '22: Australasian Computing Education Conference, Virtual Event, Australia, February 14 - 18, 2022*, pages 10–19. ACM.

Gao, Z., Ries, C., Simon, H. U., and Zilles, S. (2017). Preference-based teaching. *Journal of Machine Learning Research*, 18:31:1–31:32.

Goldman, S. A. and Kearns, M. J. (1995). On the complexity of teaching. *Journal of Computer and System Sciences*, 50(1):20–31.

Hadley, R. F. and Cardei, V. C. (1999). Language acquisition from sparse input without error feedback. *Neural Networks*, 12(2):217–235.

Håvardstun, B. A. T., Ferri, C., Hernández-Orallo, J., Parviainen, P., and Telle, J. A. (2023). XAI with machine teaching when humans are (not) informed about the irrelevant features. In Koutra, D., Plant, C., Rodriguez, M. G., Baralis, E., and Bonchi, F., editors, *Machine Learning and Knowledge Discovery in Databases: Research Track. ECML PKDD 2023*.

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

Nédellec, C., Rouveirol, C., Adé, H., Bergadano, F., and Tausend, B. (1996). Declarative bias in ILP. *Advances in inductive logic programming*, 32:82–103.

Shafto, P., Goodman, N. D., and Griffiths, T. L. (2014). A rational account of pedagogical reasoning: Teaching by, and learning from, examples. *Cognitive Psychology*, 71:55 – 89.

Telle, J. A., Hernández-Orallo, J., and Ferri, C. (2019). The teaching size: Computable teachers and learners for universal languages. *Machine Learning*, 108(8-9):1653–1675.

Vardi, M. Y. (2022). *Efficiency vs. Resilience: Lessons from COVID-19*, pages 285–289. Springer International Publishing, Cham.

Wang, C., Singla, A., and Chen, Y. (2021). Teaching an active learner with contrastive examples. In Ranzato, M., Beygelzimer, A., Dauphin, Y. N., Liang, P., and Vaughan, J. W., editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 17968–17980.

Whigham, P. A. (1996). Search bias, language bias, and genetic programming. *Genetic Programming*, 1996:230–237.

Yang, S. C.-H., Vong, W. K., Sojitra, R. B., Folke, T., and Shafto, P. (2021). Mitigating belief projection in explainable artificial intelligence via bayesian teaching. *Scientific reports*, 11(1):1–17.

Yu, S. (1988). Can the catenation of two weakly sparse languages be dense? *Discrete applied mathematics*, 20(3):265–267.

Zhang, X., Bharti, S. K., Ma, Y., Singla, A., and Zhu, X. (2021). The sample complexity of teaching by reinforcement on q-learning. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 10939–10947. AAAI Press.

Zhu, X., Singla, A., Zilles, S., and Rafferty, A. N. (2018). An overview of machine teaching. *arXiv preprint arXiv:1801.05927*.

Zilles, S., Lange, S., Holte, R., and Zinkevich, M. (2011). Models of cooperative teaching and learning. *Journal of Machine Learning Research*, 12(Feb):349–384.