Proceedings of the Fourteenth International Conference on Principles of Knowledge Representation and Reasoning

Simultaneous Learning and Prediction

Loizos Michael
Open University of Cyprus
loizos@ouc.ac.cy

Abstract

Agents in real-world environments may have only *partial* access to available information, often in an arbitrary, or hard to model, manner. By reasoning with knowledge at their disposal, agents may hope to recover some missing information. By acquiring the knowledge through a process of learning, the agents may further hope to guarantee that the recovered information is indeed correct. Assuming only a black-box access to a learning process and a prediction process that are able to cope with missing information in some principled manner, we examine how the two processes should interact so that they improve their overall joint performance. We identify natural scenarios under which the interleaving of the processes is provably beneficial over their independent use.

Introduction

Consider a medical database, with one row for each patient, and one column for each attribute of interest to some medical doctor. Due to numerous reasons, not every patient-attribute pair is associated with a definite value. Can the missing values be reliably and efficiently predicted / recovered, given the rest of the available information, and without assuming any particular structure in the way information is missing?

It is typical for research in Knowledge Representation and Reasoning (KRR) to assume that an agent (e.g., the medical doctor in our example scenario) has access to some knowledge base of rules that collectively capture certain regularities in the environment of interest. By applying such rules on the available information, the agent draws inferences on those aspects of its available inputs that are not explicitly observed, effectively recovering some of the missing information. Research, then, seeks to identify how to most appropriately represent and reason with rules for this recovery task.

A knowledge base need not comprise only infallible rules, in that rules need not be fully qualified at an individual or local level. Instead, rules could be qualified via the reasoning process, so that they are blocked from drawing an inference they support if and when another piece of evidence supports a contradictory inference, and the latter takes precedence. First, rules could be *endogenously qualified*, if they are explicitly known to be overridden by certain other rules in the knowledge base (e.g., the rule that "high fever during winter suggests a flu infection" could be overridden by another rule that "high fever near a swamp suggests a malaria infection"). Second, rules could be *exogenously qualified*, if they yield in the presence of explicit, but external to the knowledge base, information that contradicts them (e.g., the rule that "pain or pressure in the chest or arms suggests a heart attack" could yield to an explicit observation of a negative test result.

Accounting for the qualifications, the underlying assumption remains that after applying a (sound) reasoning process on the knowledge base and the available information, the drawn inferences should be accepted as correctly recovering the missing information. And they should be, if the knowledge that is available for the agent to reason with is, indeed, appropriate for the environment on which it is being used.

It has often been argued that the appropriateness of knowledge can be guaranteed if (and in many settings, only if) it is acquired through a process of induction over the agent's experiences from its environment (Valiant 2013). Indeed, by a simple, in retrospect, statistical argument (Valiant 1984), rules that have been found to be correct on past experiences are also *Probably* (i.e., except with a small probability with which the agent did not have typical experiences from its environment) *Approximately Correct* (i.e., they will predict correctly on almost all of the agent's future experiences).

To learn knowledge with the aim to recover some missing information in its future partially observable experiences, an agent has access only to such equally partially observable experiences from which it can learn. It has been shown that the PAC learning semantics (Valiant 1984) can be extended to deal with arbitrarily missing information, and that certain PAC learning algorithms can be easily modified to cope with such experiences (Michael 2010), even under the seemingly arduous requirements that we later impose on the guarantees provided by these algorithms. In the interest of generality, we shall not deal in the sequel with the exact details of how such algorithms work. Instead, we shall, henceforth, assume a black-box access to any algorithm for learning from partially observable experiences, and call it the *base algorithm*.

Applying a base algorithm on the database, a medical doctor may seek to learn a predictor P_i for each of the database attributes x_i, and apply these predictors on each row of the database to complete some of the missing information. Due to partial information and imperfect training, the predictors

may sometimes abstain or predict incorrectly when applied, giving rise to a database that is generally more complete, but less sound. The question we seek to answer in this work is this: *Is there a natural and general technique to improve the completeness and soundness of the above naive approach?*

It is important to clarify upfront that the notions of soundness and completeness that are used in this work are akin, but not identical, to those used in KRR research. In the latter, one typically has a semantics that determines, for each input and knowledge base, the inferences that are valid / correct (e.g., by specifying the models of the given input and knowledge base). Then, one may conceivably consider a number of different reasoning processes, and evaluate each one in terms of soundness and completeness thus: a reasoning process is sound if the inferences it produces *agree* with the inferences specified by the semantics, and it is complete if it produces *all* the inferences that are specified by the semantics.

Since knowledge bases in KRR research are typically assumed to truly capture the structure of the environment of interest, a definition of soundness and completeness against a knowledge base is appropriate. When learning is involved, however, the knowledge base is only *approximately* correct. The goal of the combined learning and reasoning processes is, then, to produce inferences not as specified against the agent's own knowledge base, but rather as specified against some "ideal" knowledge base; or in other words, to produce inferences specified by the underlying truth of things.

In effect, this work considers different policies (i.e., ways to apply learned rules) for drawing inferences, and evaluates them in terms of their soundness and completeness. More precisely, we take the stance that soundness should not be compromised (beyond the approximate correctness that cannot be avoided due to learning), and seek to find policies that draw inferences in a manner that improves completeness.

In seeking such policies we ask: When is it meaningful to *learn* the predictors (for each attribute) independently of each other? When is it meaningful to *apply* them independently of each other? Can something be gained by interleaving their learning and their application? We offer formal answers to these questions, and discuss how our results apply when knowledge is acquired through learning in other typical settings, such as supervised classification / regression.

Preliminaries and Notation

Fix a set $\mathcal{A}$ of attributes x_i, with $i \in \{1, \dots n\}$. Denote by $dom[i]$ the domain of attribute x_i, and by $dom[\mathcal{A}]$ the cross product $dom[1] \times \cdots \times dom[n]$ of all attribute domains. A ***complete assignment*** of values to the attributes $\mathcal{A}$ is denoted by $\texttt{asg} \in dom[\mathcal{A}]$, and the value of attribute x_i in $\texttt{asg}$ is denoted by $\texttt{asg}[i]$. To account for partial observability, denote by $dom^*[i]$ the extended domain $dom[i] \cup \{*\}$, and by $dom^*[\mathcal{A}]$ the cross product $dom^*[1] \times \cdots \times dom^*[n]$ of all extended attribute domains; $*$ stands for "don't know". An ***(observed) assignment*** of values to the attributes $\mathcal{A}$ is denoted by $\texttt{obs} \in dom^*[\mathcal{A}]$, and the value of attribute x_i in $\texttt{obs}$ is denoted by $\texttt{obs}[i]$. Our example medical database can, then, be represented simply as a list $\mathcal{S}^*$ of assignments.

Given $\mathcal{S}^*$, we seek a new list $\mathcal{S}'$ of assignments in which some of the $*$ values are replaced with the "actual" (as determined by nature) values, from the domain of the corresponding attributes. That is, given the partial inputs and whatever relevant domain knowledge we have at our disposal, we seek to reason and infer (some of) the information that is not explicitly given in the inputs. For instance, while the height-at-birth attribute of some patient might be missing in medical database $\mathcal{S}^*$, the patient does have some value associated with that attribute, and the goal is to accurately recover it.

Although the recovery process does not have access to the underlying reality, our analysis of the recovery process will make use of it. We shall, henceforth, consider $\mathcal{S}$ to be a list of complete assignments, and assume that $\mathcal{S}^*$ is always such that the j-th observed assignment $\texttt{obs} \in \mathcal{S}^*$ is obtained by applying an unknown fixed process $\texttt{mask}$ on the j-th complete assignment $\texttt{asg} \in \mathcal{S}$. In the general case, the ***masking process*** $\texttt{mask}$ is a (possibly stochastic) mapping from $dom[\mathcal{A}]$ to $dom^*[\mathcal{A}]$. This general treatment captures cases where masking may (arbitrarily) hide or even distort parts of the underlying reality. For this work we shall restrict our attention to masking processes that only hide (without distorting) assignments, by insisting that whenever $\texttt{obs}$ is obtained from $\texttt{mask}(\texttt{asg})$, either $\texttt{obs}[i] = *$ (i.e., information is hidden) or $\texttt{obs}[i] = \texttt{asg}[i]$ (i.e., information is preserved). Such an observed assignment $\texttt{obs}$ shall be said to ***mask*** the complete assignment $\texttt{asg}$, and each attribute $x_i \in \mathcal{A}$ whose value $\texttt{obs}[i]$ is $*$ shall be said to be ***masked*** in $\texttt{obs}$.

A ***base algorithm*** is one that given access to a list $\mathcal{S}^*$ and an attribute $x_i \in \mathcal{A}$, returns a predictor P_i. A ***predictor*** P_i for x_i is a mapping from $dom[\mathcal{A} \setminus \{x_i\}]$ to $dom[i]$ that is computable in time polynomial in n (i.e., in its input size). A given assignment $\texttt{obs} \in dom^*[\mathcal{A}]$ may not necessarily offer enough information to *uniquely* determine the value of the mapping. In such a case, and only then, P_i abstains from making a prediction. Thus, the predictor P_i is naturally lifted to a mapping from $dom^*[\mathcal{A} \setminus \{x_i\}]$ to $dom^*[i]$ to account for abstaining from predictions when sufficient information is not available to determine a definite value. To reduce notation, we shall write $P_i(\texttt{obs})$ to denote the prediction of P_i on assignment $\texttt{obs} \in dom^*[\mathcal{A}]$, with the implicit understanding that only the values of attributes $\mathcal{A} \setminus \{x_i\}$ are read from $\texttt{obs}$. When P_i abstains due to insufficient information in $\texttt{obs}$ to determine a prediction, then $P_i(\texttt{obs}) = *$.

We shall restrict our attention to ***classical*** (or monotonic) predictors, which we take to mean that when they make a definite prediction on some assignment, they do not change that prediction in the presence of more information.[1] Other than that, we shall treat the base algorithm and the returned predictors as black boxes. It is exactly therein that lies the general applicability of our framework and formal results.

Dealing with Multiple Predictors

Consider a medical database $\mathcal{S}^*$ with multiple rows, and 7 columns corresponding to real-valued patient characteristics (e.g., cholesterol level). Let the complete assignment $\texttt{asg} =$

[1] Although reasoning with non-monotonic rules has a very long and rich history in KRR research (Reiter 1980), the learning of non-monotonic rules has received considerably less attention in Learning Theory research (Schuurmans and Greiner 1994) and the practical deployment of Machine Learning techniques (Mitchell 1997).

$\langle -1, 0, 5, 0, 4, -7, 9\rangle \in \mathcal{S}$ determine the actual values that these characteristics take for patient John Doe, and let the observed assignment $\texttt{obs} = \langle -1, 0, *, 0, 4, *, 9\rangle \in \mathcal{S}^*$ correspond to John Doe's (partial) medical record as found in the medical database $\mathcal{S}^*$. Suppose that a doctor reads in a medical journal that the values of certain patient characteristics can be predicted from the values of the other patient characteristics through a set $\mathbb{P}$ of the following[2] predictors:
(i) P_1 predicts according to rule "$x_3 := |\sqrt{x_5}| + |\sqrt{x_7}|$";
(ii) P_2 predicts according to rule "$x_4 := x_2 \cdot x_6$";
(iii) P_3 predicts according to rule "$x_6 := x_3 - x_5/2$".

The doctor wishes to recover reliably as much of the information missing in $\mathcal{S}^*$, and in John Doe's medical record in particular. To this end, the doctor chooses to apply the predictors in $\mathbb{P}$ *in parallel*, in the following sense: each of the predictors is applied on obs independently of the others, the prediction of each predictor is computed, and only then the predictions are used to complete the information that is missing in John Doe's medical record. By following this process, the doctor computes the predictions of P_1, P_2, P_3 to be, respectively, 5, 0, $*$. Note that predictor P_3 abstains since its prediction cannot be determined without the value of x_3. Incorporating these predictions in obs yields $\texttt{obs}' = \langle -1, 0, 5, 0, 4, *, 9\rangle$, which is sound when compared against the underlying truth asg, and more complete than obs.

Applying predictors in parallel is not the only option. For instance, the doctor could choose to apply the predictors *sequentially*, in the following sense: apply one of the predictors on obs, compute its prediction, modify obs to incorporate that prediction, and repeat the process with another predictor. Unlike the parallel process, the second predictor will now be applied not on obs, but on the new version of John Doe's medical record that has incorporated the prediction of the first predictor. By following this process, the doctor applies P_1 on $\texttt{obs} = \langle -1, 0, *, 0, 4, *, 9\rangle$, and its prediction 5 for x_3 turns obs into $\texttt{obs}' = \langle -1, 0, 5, 0, 4, *, 9\rangle$. Subsequently, the doctor considers P_2, but seing that the value of x_4 is already known in $\texttt{obs}'$, the doctor chooses to let $\texttt{obs}'' = \texttt{obs}'$, effectively ignoring the prediction of P_2. Finally, the doctor applies P_3 on $\texttt{obs}''$, and its prediction 3 for x_6 turns $\texttt{obs}''$ into $\texttt{obs}''' = \langle -1, 0, 5, 0, 4, 3, 9\rangle$.

Evidently, the two ways of applying the very same set of predictors yield different results. Indeed, in the second case, predictor P_3 has access to a value for attribute x_3 at the time of its application, and is able to make a definite prediction. The resulting medical record $\texttt{obs}'''$ is now fully complete, but unsound when compared against the underlying truth asg, since the value of x_6 is predicted, but incorrectly.

A set of predictors may be applied in other more involved ways as well; e.g., first apply $\{P_1, P_3\}$ in parallel, and subsequently apply $\{P_2\}$. It is even possible for the value of some attribute to be predicted on multiple occasions by predictors that are applied, however, at different points in the process. Henceforth, we shall assume that an ordering of predictors is implicitly prescribed by each given $\mathbb{P}$. We shall call $\mathbb{P}$ a ***policy***, and think of it as a sequence of non-empty sets of predictors, so that the predictors in each set are applied in parallel (with each predictor in the set predicting the value of a different attribute), before those in the subsequent set are applied, and so on. We shall write $\mathbb{P}(\texttt{obs})$ to mean the final assignment obtained when predictors in $\mathbb{P}$ are applied on obs in the prescribed ordering. We shall write $\mathbb{P}(\mathcal{S}^*)$ to mean the list with elements in $\{\mathbb{P}(\texttt{obs}) \mid \texttt{obs} \in \mathcal{S}^*\}$.

Note that a prediction $\mathbb{P}(\texttt{obs}) \in dom^*[\mathcal{A}]$ of a policy $\mathbb{P}$ on obs determines values in $dom^*[\mathcal{A}]$ for all attributes $\mathcal{A}$, since every attribute on which a predictor in $\mathbb{P}$ abstains will have its value persist from what is given in $\texttt{obs} \in \mathcal{S}^*$. A policy $\mathbb{P}$ is ***complete on*** $\texttt{obs} \in \mathcal{S}^*$ if $\mathbb{P}(\texttt{obs}) \in dom[\mathcal{A}]$. A policy $\mathbb{P}$ is ***sound on*** $\texttt{obs} \in \mathcal{S}^*$ ***against*** $\texttt{asg} \in \mathcal{S}$ if $\mathbb{P}(\texttt{obs})$ masks asg; soundness concisely states that the resulting value for each $x_i \in \mathcal{A}$ is either $*$, or correct according to the nature-assigned value of x_i in asg. By extension: A policy $\mathbb{P}$ is ***complete on*** a given fraction / sublist of $\mathcal{S}^*$ if it is complete on each obs in that fraction / sublist of $\mathcal{S}^*$. A policy $\mathbb{P}$ is ***sound on*** a given fraction / sublist of $\mathcal{S}^*$ ***against*** $\mathcal{S}$ if it is sound on each obs in that fraction / sublist of $\mathcal{S}^*$ against the asg in $\mathcal{S}$ that corresponds to obs.

We shall employ the notions of soundness and completeness mutatis mutandis on individual predictors as well, by thinking of them as policies that include a single predictor.

Before continuing with the technical analysis of different policies in subsequent sections, certain remarks are in order.

The first remark relates to soundness and completeness and how these notions relate to their typical use in KRR. In addition to the points we have already made in the first section, we note also the quantitative nature of the notions. We will not refer to a policy as being simply sound or complete, but we will quantify the fraction / percentage of the given inputs on which soundness and completeness are achieved.

The second remark relates to how rules are applied. Some form of sequentiality when applying rules seems to resemble proof procedures in KRR research more closely than the parallel application of rules. After all, one could argue, the latter approach seems to unnecessarily waste the opportunity to get more complete inferences by having rules interact. On the other hand, the parallel application of rules seems to be closer to what a lot of Machine Learning research has traditionally done when dealing with multiple predictors; certain exceptions of this trend are discussed in the last section.

In the sequel we shall identify the reason behind the above discrepancy, which comes down to the fact that applying *learned* (and hence approximately correct) rules naively in some sequential fashion, can reduce soundness. On the other hand, we shall show that an appropriate application of rules in some non-parallel fashion provably improves their combined soundness and completeness. The main question, then, becomes that of finding how to apply rules so that to exploit this improvement, without sacrificing their soundness.

The third remark relates to the qualification of rules. When each rule in a policy is applied, its prediction is used to update information on an attribute only if information on the attribute's value is missing. If an attribute's value is observed, then the prediction will yield to this explicit observation, and be exogenously qualified. It can be easily observed

[2]Predictors utilizing other representations for the rules according to which they make predictions are also possible. In particular, utilizing boolean attributes and a logic-based representation gives rise to rules closer to what a lot of KRR research employs.

that the same treatment accounts for the endogenous qualification of rules as well. If rules that are applied first do not abstain and do make a prediction, then those predictions will complete missing information, which will subsequently prevent the predictions of rules that are applied later from being recorded; in effect, the former rules override the latter rules.

Our investigation herein seeks to learn these rules, and the ordering in which they are applied. Since ordering captures rule qualification, the knowledge bases that we end up learning are ultimately non-classical (i.e., non-monotonic), even though each individual rule is restricted to being classical.

Benefits of Chaining Predictors

Consider a policy $\mathbb{P}$ that prescribes a sequence of d sets of predictors. We shall call d the ***depth of*** $\mathbb{P}$, and the sets the ***layers of*** $\mathbb{P}$, indexed by their order in the sequence. A policy $\mathbb{P}$ is ***flat*** if its depth d is 1, and is ***chained*** if $d \geq 2$; i.e., when at least one predictor in $\mathbb{P}$ is applied *after* some other predictor in $\mathbb{P}$. An ***empty*** policy $\mathbb{P}$ has depth $d = 0$. A policy $\mathbb{P}_1$ is a ***reordering of*** a policy $\mathbb{P}_2$ if the multi-set of the union of the layers of $\mathbb{P}_1$ matches that of $\mathbb{P}_2$; i.e., the policies share the predictors that are being used, but may differ on their depths and on how predictors are assigned to the policy layers.

We proceed to consider whether having chained policies is ever a good idea. In other words, we seek to understand if a form of sequential application of rules, as done in proof procedures in KRR, is useful when it comes to *learned* rules?

The distinction between *learned* and *programmed* rules is critical and should be emphasized. It is trivial to demonstrate the benefits (in terms of completeness) of chaining in knowledge bases with programmed rules: given the rules "if you have fever then you are sick" and "if you are sick then visit a doctor", the inference "visit a doctor" cannot be drawn from the observation "you have fever" without using both rules, since no single rule can bridge the gap between the available observation and the sought inference. However, when rules are learned, there is nothing preventing the learning process from inducing a single rule that bridges this gap (e.g., "if you have fever then visit a doctor"). In theory, then, the learning process itself could render the need for chaining superfluous.

We have revealed in the preceding section that, despite the concern above, chaining remains provably beneficial for knowledge bases with *learned* rules. To establish this claim, one must consider not only the completeness that a knowledge base can achieve (as in the case of programmed rules), but also the soundness. Indeed, the ability of a learning process to *construct* rules (in contrast to being given *prescribed* rules by a process of programming) may trivially lead to perfect completeness by constructing a constant rule for each attribute. It is only by including soundness in the evaluation of a knowledge base that we can demonstrate the benefits of chaining. This evaluation metric of combined soundness and completeness is captured by the following definition.

Definition 1 (Chaining Collapsibility). ***Chaining collapses on $\mathcal{S}^*$ against $\mathcal{S}$*** *if: for every policy $\mathbb{P}$ that is unsound on an ε fraction of $\mathcal{S}^*$ against $\mathcal{S}$, and incomplete on an ω fraction of $\mathcal{S}^*$, there exists a flat policy $\mathbb{P}'$ of predictors that is unsound on an ε' fraction of $\mathcal{S}^*$ against $\mathcal{S}$, and incomplete on an ω' fraction of $\mathcal{S}^*$ such that $\varepsilon' + \omega' \leq \varepsilon + \omega$.*

The sum of the soundness and completeness of a policy $\mathbb{P}$ on $\mathcal{S}^*$ against $\mathcal{S}$ will play an important role in our analysis, and shall be called the ***performance*** of $\mathbb{P}$ ***on $\mathcal{S}^*$ against $\mathcal{S}$***. Chaining is provably beneficial if a situation can be demonstrated where the performance of a particular policy $\mathbb{P}$ cannot be matched by *any* flat policy (not necessarily a reordering of $\mathbb{P}$). We shall demonstrate such a situation next.

Fix an assignment $\texttt{obs} \in \mathcal{S}^*$, a predictor P_t for attribute $x_t \in \mathcal{A}$, and an attribute $x_i \in \mathcal{A}$ that is masked in $\texttt{obs}$. Consider changing the $*$ value of x_i in $\texttt{obs}$ to a value in $dom[i]$. If at least one such value change causes the prediction of P_t to change, then x_i is ***relevant for*** P_t ***w.r.t.*** $\texttt{obs}$. If at least two such value changes cause the prediction of P_t to change in two different ways, then x_i is ***critical for*** P_t ***w.r.t.*** $\texttt{obs}$.

By way of illustration, consider a predictor P_3 that predicts according to the rule "$x_3 := x_1 \cdot x_2$". Given the assignment $\texttt{obs} = \langle *, 2, -1 \rangle$, the prediction of P_3 is $*$. Changing the value of x_1 in $\texttt{obs}$ to any real number causes the prediction of P_3 to change from $*$ to the double of that number. Thus, attribute x_1 is critical (and relevant) for P_3 w.r.t. $\texttt{obs}$.

Lemma 1 (Uniqueness of Critical Attributes). *Consider a predictor P_t for attribute $x_t \in \mathcal{A}$, and an assignment $\texttt{obs}$. Then: If attribute $x_i \in \mathcal{A}$ is critical for P_t w.r.t. $\texttt{obs}$, then no other attribute $x_j \in \mathcal{A}$ is relevant for P_t w.r.t. $\texttt{obs}$.*

Proof. Consider the assignment $\texttt{obs}_{u,v}$ obtained from $\texttt{obs}$ after replacing $\texttt{obs}[i]$, $\texttt{obs}[j]$ respectively with u, v. If x_i is critical for P_t w.r.t. $\texttt{obs}$, then $P_t(\texttt{obs}_{c,*})$, $P_t(\texttt{obs}_{d,*})$, $*$ are distinct for some $c, d \in dom[i]$, $c \neq d$. Assume, by way of contradiction, that x_j is relevant for P_t w.r.t. $\texttt{obs}$. Then $P_t(\texttt{obs}_{*,a}) \neq *$ for some $a \in dom[j]$. Since P_t is classical, $P_t(\texttt{obs}_{c,a}) = P_t(\texttt{obs}_{*,a}) = P_t(\texttt{obs}_{d,a}) = P_t(\texttt{obs}_{d,*}) \neq P_t(\texttt{obs}_{c,*}) = P_t(\texttt{obs}_{c,a})$; a contradiction, as needed. □

Hence, no classical predictor can have its predictions be unilaterally affected / determined by the values of more than one attribute. Although this limitation does not necessarily burden non-classical predictors, Lemma 1 bears a significant weight due to the widespread use of classical predictors in Machine Learning research. To prove that chaining does not collapse, it suffices to demonstrate any particular policy $\mathbb{P}$ that chains its predictors in a manner that effectively simulates non-classical predictors. We present such a policy next.

Consider a list $\mathcal{S}$ of complete assignments, where unbeknownst to a base algorithm, attributes x_1 and x_2 are always assigned values equal to each other, and x_3 is assigned their product. In all assignments in $\mathcal{S}^*$ available to the base algorithm, both x_3 and exactly one of x_1, x_2 are masked. Given *our* knowledge (of $\mathcal{S}$) that x_1 and x_2 share common values, we may expect the base algorithm to return a predictor P_0 for x_3 that always makes correct predictions. Using this predictor P_0, along with a predictor P_1 that predicts according to the rule "$x_1 := x_2$", and a predictor P_2 that predicts according to the rule "$x_2 := x_1$", we would be able to construct a flat policy that would be perfectly sound and complete. Alas, this scenario is not possible, as a consequence of Lemma 1.

Theorem 2 (The Benefits of Chaining). *There exist lists $\mathcal{S}^*$ and $\mathcal{S}$ such that chaining does not collapse on $\mathcal{S}^*$ against $\mathcal{S}$.*

Proof. Consider lists $\mathcal{S}^*$ and $\mathcal{S}$ as in the paragraph above.

Let policy $\mathbb{P}$ have predictors P_1 and P_2 from above in its first layer, and a predictor P_3 according to the rule "$x_3 := x_1 \cdot x_2$" in its second layer. $\mathbb{P}$ has optimal performance.

Now further specify list $\mathcal{S}^*$ and list $\mathcal{S}$ so that: At least two complete assignments $\mathtt{asg}_1, \mathtt{asg}_2 \in S$ are such that $\mathtt{asg}_1[1] \neq \mathtt{asg}_2[1]$; i.e., two different values in the domain $dom[1]$ of x_1 appear in complete assignments in $\mathcal{S}$. The assignments $\mathtt{obs}_1, \mathtt{obs}_2 \in \mathcal{S}^*$ that correspond to $\mathtt{asg}_1, \mathtt{asg}_2 \in S$ mask attribute x_2. At least one assignment $\mathtt{obs}_3 \in \mathcal{S}^*$ masks attribute x_1, and for each attribute $x_i \in \mathcal{A} \setminus \{x_1, x_2, x_3\}$, it is the case that $\mathtt{obs}_3[i] = \mathtt{obs}_1[i]$; i.e., the "context" defined by the other attributes is the same, although the values of the attributes of interest may differ.

Assume that some flat policy $\mathbb{P}'$ has optimal performance. Then, $\mathbb{P}'$ includes a predictor P_3 for x_3 that always makes a correct definite prediction. Since its predictions on $\mathtt{obs}_1$ and $\mathtt{obs}_2$ differ, then x_1 is critical for P_3 w.r.t. the assignment $\mathtt{obs}_*$ that is obtained from either $\mathtt{obs}_1$ or $\mathtt{obs}_2$ by replacing the value of x_1 with $*$. Necessarily, the prediction of P_3 when applied on $\mathtt{obs}_*$ is $*$. Assignment $\mathtt{obs}_*$ can be obtained also from $\mathtt{obs}_3$ by replacing the value of x_2 with $*$. Since predictor P_3 makes a definite prediction when applied on $\mathtt{obs}_3$, then x_2 is relevant for P_3 w.r.t. the assignment $\mathtt{obs}_*$. A contradiction by Lemma 1, as needed. □

Collectively, the predictors in policy $\mathbb{P}$ given in the proof of Theorem 2 implement a non-classical rule: "$x_3 := (x_1)^2$ if x_2 *is not known*, or $(x_2)^2$ if x_1 *is not known*". The rule's non-classical nature is evidenced by observing that if more information becomes known, the rule may retract a definite prediction it had previously made. As a matter of fact, it can be shown that any "epistemic" rule that makes predictions based not only on what holds, but also on what is *known*, can be simulated by appropriately choosing and chaining classical predictors as part of a certain policy (Michael 2008).

Can the Benefits be Realized?

Chaining predictors in some *appropriate* manner can provably enhance the combined soundness and completeness that is achieved. Are there concrete efficient algorithms that identify such a chaining? We consider two approaches below.

First Learn, Then Chain

In the first approach proceed as follows: First, call the base algorithm on list $\mathcal{S}^*$ to get one predictor P_i for each attribute $x_i \in \mathcal{A}$. Then, order the predictors to obtain a policy $\mathbb{P}$.

Such an approach captures what is implicitly assumed in certain research in KRR: that some process effectively produces the rules in the knowledge base, after which the rules can be applied in any order to draw inferences while safely ignoring the process through which the rules were produced.

Is the approach of first learning and then chaining (FLTC) the predictors ever useful? To answer this question, we need to examine more precisely the guarantees that the base algorithm is assumed to offer on the predictors it returns.

Scenario 1: Guarantees on $\mathcal{S}$

For our first scenario, assume that each P_i is guaranteed to be unsound on only an ε_i fraction of $\mathcal{S}$ against $\mathcal{S}$. Note that the predictive guarantees are w.r.t. the *complete assignments* $\mathcal{S}$, which capture the underlying reality associated with $\mathcal{S}^*$. Thus, soundness is assumed even when the predictors are applied on complete assignments in $\mathcal{S}$ (where, in particular, abstentions are disallowed); a more stringent requirement than the general case where soundness is assumed on $\mathcal{S}^*$ against $\mathcal{S}$. The problem of how such guarantees can be effectively obtained is orthogonal, and is discussed later in this work. Let $\varepsilon \triangleq \sum_{i=1}^{n} \varepsilon_i < 1$. We study the performance of $\mathbb{P}$ on $\mathcal{S}^*$.

Theorem 3 (Soundness in FLTC Setting). *Consider the predictors as described above. Any policy $\mathbb{P}$ that uses these predictors is unsound on at most an ε fraction of $\mathcal{S}^*$ against $\mathcal{S}$.*

Proof. By a union bound, since predictors are classical. □

Theorem 4 (Completeness in FLTC Setting). *Consider the predictors as described above. Let policy $\mathbb{P}$ have n layers, with each predictor appearing in every layer. Let policy $\mathbb{P}'$ be a reordering of $\mathbb{P}$. Then: If $\mathbb{P}$ and $\mathbb{P}'$ are, respectively, incomplete on an ω and ω' fraction of $\mathcal{S}^*$, then $\omega \leq \omega'$.*

Proof. The policy $\mathbb{P} + \mathbb{P}'$ that concatenates the layers of $\mathbb{P}'$ after all the layers of $\mathbb{P}$, is not less complete than $\mathbb{P}'$. It suffices to establish that $\mathbb{P}$ is not less complete than $\mathbb{P} + \mathbb{P}'$. Consider an assignment $\mathtt{obs} \in \mathcal{S}^*$. If $\mathbb{P}(\mathtt{obs}) \in dom[\mathcal{A}]$, then the claim follows. Otherwise, and since $\mathbb{P}$ has n layers, there must exist some layer in $\mathbb{P}$ that does not complete the value of any new attributes (that were not already completed by earlier layers). Since that layer is a superset of each subsequent layer in $\mathbb{P}+\mathbb{P}'$, any subsequent layer will also fail to complete the value of any new attributes. Thus, $\mathbb{P}$ and $\mathbb{P}+\mathbb{P}'$ will abstain on the same attributes. The claim follows. □

Theorems 3–4 offer a first efficient algorithm for chaining predictors, with bounded unsoundness, and optimal, in a defined sense, completeness. However, the assumed guarantees on the predictors might not always be possible to have.

Before moving on, we note that this scenario accounts for what is assumed in some KRR work: each rule truly captures the structure of the environment of interest (i.e., unsound on an $\varepsilon_i = 0$ fraction of $\mathcal{S}$), and so chaining rules in *any* order is justified without cost on the overall soundness (i.e., $\varepsilon = 0$). Further, applying each predictor (in a given knowledge base) as often as possible is optimal in terms of completeness.

Scenario 2: Guarantees on $\mathcal{S}^*$

For our second scenario, assume that each P_i is guaranteed to be unsound on only an ε_i fraction of $\mathcal{S}^*$ against $\mathcal{S}$. The predictive guarantees are w.r.t. the *observed assignments* $\mathcal{S}^*$. Let $\varepsilon \triangleq \sum_{i=1}^{n} \varepsilon_i < 1$. We study the performance of $\mathbb{P}$ on $\mathcal{S}^*$.

Although Theorem 4 holds unchanged, a naive ordering approach compromises Theorem 3 and the soundness of $\mathbb{P}$.

Theorem 5 (Unsoundness in FLTC Setting). *Consider the predictors as described above. Assume that at least one predictor is not constant on all complete assignments. Then: There exists a policy $\mathbb{P}$ (in fact, there exist exponentially in n*

many such policies) that orders these predictors that is unsound on all of $\mathcal{S}^*$ against $\mathcal{S}$, even though some reordering of $\mathbb{P}$ is unsound on at most an ε fraction of $\mathcal{S}^*$ against $\mathcal{S}$.

Proof. Let P_t be a non-constant predictor for $x_t \in \mathcal{A}$. Order $dom[\mathcal{A}]$ as a Gray Code (Knuth 2011). Choose two complete assignments $\mathtt{asg}_1, \mathtt{asg}_2 \in dom[\mathcal{A}]$ that are consecutive in the Gray Code and such that $P_t(\mathtt{asg}_1) \neq P_t(\mathtt{asg}_2)$, and let x_j be the attribute on which the assignments differ. Since P_t does not base its predictions on x_t, then $x_j \neq x_t$, and $\mathtt{asg}_1[t] = \mathtt{asg}_2[t]$. Without loss of generality, let $\mathtt{asg}_1$ be one among the two chosen complete assignments such that $P_t(\mathtt{asg}_1) \neq \mathtt{asg}_1[t]$. Construct the observed assignment $\mathtt{obs}_1$ from $\mathtt{asg}_1$ by replacing $\mathtt{asg}_1[j]$ and $\mathtt{asg}_1[t]$ with $*$.

Let $\mathcal{S}$ include only $\mathtt{asg}_1$, and let $\mathcal{S}^*$ include only $\mathtt{obs}_1$. Then, $P_t(\mathtt{obs}_1) = *$. Choose a predictor P_j for $x_j \in \mathcal{A}$ such that $P_j(\mathtt{obs}_1) = \mathtt{asg}_1[j]$. Choose a predictor P_i for each $x_i \in \mathcal{A} \setminus \{x_t, x_j\}$ such that $P_i(\mathtt{obs}_1) \in \{\mathtt{asg}_1[i], *\}$. It is consistent for all these predictors to have been returned by the base algorithm as assumed in the current scenario. A flat policy of the predictors is sound on all of $\mathcal{S}^*$ against $\mathcal{S}$.

Consider a policy $\mathbb{P}$ out of the exponentially in n many ones that order P_j before P_t. Since $P_j(\mathtt{obs}_1) = \mathtt{asg}_1[j]$, then policy $\mathbb{P}$ will apply P_t on an assignment obtained from $\mathtt{obs}_1$ by replacing $\mathtt{obs}_1[j]$ with $\mathtt{asg}_1[j]$. On this assignment, P_t will predict a value in the domain $dom[t]$ of x_t that differs from $\mathtt{asg}_1[t]$. Therefore, $\mathbb{P}(\mathtt{obs}_1)$ will not mask $\mathtt{asg}_1$, and hence $\mathbb{P}$ is unsound on all of $\mathcal{S}^*$ against $\mathcal{S}$. The flat policy given earlier, which is a reordering of $\mathbb{P}$, is trivially unsound on at most an ε fraction of $\mathcal{S}^*$ against $\mathcal{S}$. □

The severe loss of soundness in the second scenario is not due to the cumulative effect of the unsoundness of individual predictors; that happens in the first scenario. Rather, it is due to chaining *forcing* predictors to make predictions when they would have otherwise abstained. More specifically: The guarantees of each predictor P_i are w.r.t. $\mathcal{S}^*$. Yet, P_i is applied to predict on assignments that are not in $\mathcal{S}^*$, but assignments that have incorporated the predictions of predictors in earlier layers in $\mathbb{P}$. These more complete assignments are the ones that cause P_i to enhance its completeness by abstaining less, at the cost, however, of sacrificing its soundness.

SLAP-ing with Predictors

The naive FLTC approach essentially trades off soundness for completeness. To boost completeness without sacrificing soundness, the predictors need to be applied on assignments w.r.t. which they (will be made to) have guarantees.

The second approach, then, recognizes that learning and prediction cannot proceed independently, and that one needs to simultaneously learn and predict (SLAP) with predictors.

Algorithm 1, henceforth denoted by $\mathcal{L}_{\text{SLAP}}$, achieves the need to SLAP by interleaving learning and prediction to iteratively build a chained policy $\mathbb{P}$. It is given a list $\mathcal{S}^*$ of size m, a non-negative integer d indicating the desired depth of $\mathbb{P}$, and a positive value ε indicating the desired upper bound on the unsoundness of $\mathbb{P}$ on $\mathcal{S}^*$ against an *unspecified* list $\mathcal{S}$.

One should not be surprised with the requirement that algorithm $\mathcal{L}_{\text{SLAP}}$ achieves soundness guarantees on $\mathcal{S}^*$ against an unspecified, and unavailable, list $\mathcal{S}$. Meeting this seemingly inordinate requirement is critically facilitated by the oracle access that algorithm $\mathcal{L}_{\text{SLAP}}$ has to a base algorithm $\mathcal{L}_{\text{base}}$. As in the second scenario of the preceding section, we shall assume that $\mathcal{L}_{\text{base}}$ is able to provide guarantees for individual predictors on $\mathcal{S}^*$ against $\mathcal{S}$. Furthermore, we shall assume that the base algorithm $\mathcal{L}_{\text{base}}$ can be made to reduce the unsoundness of its returned predictors to any specified positive value ε_k, assuming $\mathcal{L}_{\text{base}}$ is given access to list $\mathcal{S}^*$ and is allowed time polynomial in $1/\varepsilon_k$.[3] So, as long as we do not ask for "unreasonably" small (as a function of n and m) values for ε_k, the calls to base algorithm $\mathcal{L}_{\text{base}}$ will not burden the overall computational efficiency of algorithm $\mathcal{L}_{\text{SLAP}}$.

Algorithm 1 Simultaneous Learning and Prediction

input: depth d, n attributes $\mathcal{A}$, soundness error ε, list $\mathcal{S}^*$ of m assignments, base algorithm $\mathcal{L}_{\text{base}}$ as given in text.
Set $\mathcal{S}^*_0$ to equal $\mathcal{S}^*$.
for every $k = 1, 2, \ldots, d$ **do**
  for every attribute $x_i \in \mathcal{A}$ **do**
    Call the base algorithm $\mathcal{L}_{\text{base}}$ on input x_i, $\mathcal{A}$, ε_k, δ_k, and $\mathcal{S}^*_{k-1}$, and let P_i be the predictor that it returns.
  end for
  Let $\mathbb{P}_k$ be a flat policy of the predictors $\{P_i \mid x_i \in \mathcal{A}\}$.
  Let $\mathcal{S}^*_k$ be the list of assignments given by $\mathbb{P}_k(\mathcal{S}^*_{k-1})$.
end for
Set $\mathbb{P}$ so that its j-th layer includes the predictors in $\mathbb{P}_j$.
output: Return the policy $\mathbb{P}$, and terminate.

Given oracle access to base algorithm $\mathcal{L}_{\text{base}}$ with the characteristics described above, algorithm $\mathcal{L}_{\text{SLAP}}$ calls $\mathcal{L}_{\text{base}}$ on $\mathcal{S}^*$ to obtain a predictor for each attribute, and applies these predictors in parallel to update the assignments in $\mathcal{S}^*$. It repeats the process, by calling algorithm $\mathcal{L}_{\text{base}}$ on the new assignments to obtain a new predictor for each attribute, and applying the new predictors in parallel to further update the assignments, and so on for d iterations. Algorithm $\mathcal{L}_{\text{SLAP}}$ keeps track of all predictors $\mathbb{P}_j$ obtained and applied in each iteration, and constructs the policy $\mathbb{P}$ by assigning in its j-th layer the predictors that were used during the j-th iteration.

We have purposefully left unspecified what values would be considered "reasonable" for ε_k, both in the discussion above, and in Algorithm 1. We will examine two scenarios.

For our first scenario, we assume that accessing all assignments in $\mathcal{S}^*$ is conceptually meaningful and computationally plausible; e.g., when seeking to recover missing information in all records of a database. Since time linear in n and m is already needed just for reading all of $\mathcal{S}^*$, it seems reasonable to allow ε_k to be an inverse linear function of n and m.

Observe, then, that setting $\varepsilon_k = 1/2m$ would ensure that each predictor returned by $\mathcal{L}_{\text{base}}$ during the k-th iteration of $\mathcal{L}_{\text{SLAP}}$ would be perfectly sound on $\mathcal{S}^*_{k-1}$, and so would policy $\mathbb{P}_k$. As a result, list $\mathcal{S}^*_k$ would be noiseless, and by induction the returned policy $\mathbb{P}$ would be perfectly sound on $\mathcal{S}^*$.

[3] We have already remarked on the black-box view of base algorithms taken herein, and the emphasis, instead, on their appropriate use. Nonetheless, we remind the reader that such base algorithms can be constructed under typical assumptions (Michael 2010).

For our second scenario, we assume that assignments in $\mathcal{S}^*$ can only be sampled or otherwise only partly accessed; e.g., when seeking to learn predictors from a sample of data with the aim to apply them on future data. Time linear in n is still needed just for reading a single assignment in $\mathcal{S}^*$, but no dependence of the running time on m can be generally assumed.[4] Thus, we shall not allow ε_k to depend on m.

Exactly due to the restricted access assumed on $\mathcal{S}^*$, $\mathcal{L}_{\text{base}}$ may occasionally fail to produce a predictor with the sought unsoundness ε_k. Analogous to our treatment for ε_k, we shall assume that the base algorithm $\mathcal{L}_{\text{base}}$ can be made to reduce the failure probability to any specified positive value δ_k, assuming $\mathcal{L}_{\text{base}}$ is allowed time polynomial in $1/\delta_k$. For reasons same as for ε_k, we shall not allow δ_k to depend on m.

Given a base algorithm $\mathcal{L}_{\text{base}}$ as discussed above, we can establish that SLAP preserves soundness despite chaining.

Theorem 6 (Soundness in SLAP Setting). *For any inputs and value δ, there exist values ε_k and δ_k such that the policy $\mathbb{P}$ returned by algorithm $\mathcal{L}_{SLAP}$ is, except with probability δ, unsound on at most an ε fraction of $\mathcal{S}^*$ against $\mathcal{S}$. For fixed d, algorithm $\mathcal{L}_{SLAP}$ runs in time polynomial in n, $1/\varepsilon$, $1/\delta$.*

Proof. Let $q(n, 1/\varepsilon_k, 1/\delta_k)$ be a polynomial that determines the running time of the base algorithm $\mathcal{L}_{\text{base}}$ on input $\mathcal{A}$, ε_k, δ_k. It is easy to show that algorithm $\mathcal{L}_{\text{SLAP}}$ runs in time

$$O\left(d \cdot poly(n) \cdot \sum_{k=1}^{d} q(n, 1/\varepsilon_k, 1/\delta_k)\right).$$

Set $\delta_k := \delta/2d$. Then, there is a polynomial $p(n, 1/\delta, d)$ and a constant c such that setting $\varepsilon_k := (\varepsilon/p(n, 1/\delta, d))^{c^{d-k}}$ establishes that the running time of algorithm $\mathcal{L}_{\text{SLAP}}$ is polynomial in n, $1/\varepsilon$, $1/\delta$ for fixed d. Furthermore, the choices of δ_k and ε_k guarantee that, except with probability δ, each call of $\mathcal{L}_{\text{base}}$ will return a sufficiently sound predictor such that whenever it is applied in the context of algorithm $\mathcal{L}_{\text{SLAP}}$, its predictions will be sound (i.e., the improbable event of making unsound predictions will never occur). Overall then, except with probability δ, the above effectively reproduces the noiseless intermediate predictions of the first scenario. □

Theorem 6 is to the SLAP setting what Theorem 3 is to the FLTC setting. Analogously to Theorem 4 for the FLTC setting, one can show that the completeness of algorithm $\mathcal{L}_{\text{SLAP}}$ is optimal for certain natural settings. The settings in question are those where attributes (i.e., notions whose definitions one may wish to learn) are defined in terms of one another in a hierarchical fashion, so that there are no directed cycles in the definitions. We formalize such settings below.

A list $\mathcal{S}$ of complete assignments is ***d-stratified*** if for a partitioning $\mathcal{A}_1, \ldots, \mathcal{A}_d$ of $\mathcal{A}$, and for every $x_i \in \mathcal{A}_j$, there is a predictor P_i sound on all of $\mathcal{S}$ against $\mathcal{S}$ and complete on all of $\mathcal{S}$, that does not base its predictions on $A_j \cup \ldots \cup \mathcal{A}_d$.

We shall assume that we have access to base algorithms, which we shall call ***focused***, that return predictors that do not base their predictions on attributes that are clearly irrelevant due to the stratification of $\mathcal{A}$. Such base algorithms can be constructed either by having direct access to the partitioning if it is known, or by exploiting domain-specific knowledge, attribute-efficient learning, or other relevant techniques.

By way of illustration, let $d = 4$, $x_5 \in \mathcal{A}_2$ and $x_3 \in \mathcal{A}_4$, and $\texttt{obs} = \langle 8, -3, *, 7, *, -2\rangle$. Any predictor P_5 for x_5 returned by a focused base algorithm makes a definite prediction, since none of the attributes $\mathcal{A} \setminus (A_2 \cup \mathcal{A}_3 \cup \mathcal{A}_4)$ is masked. On the other hand, a definite prediction by a predictor P_3 for x_3 might not be possible since the value of x_3 may depend on the value of x_5, which is masked in obs. If we consider, however, a policy $\mathbb{P}$ that chains P_3 after P_5, the prediction of the latter predictor necessarily completes x_5's value before the prediction of the former predictor on x_3, *ensuring* that a definite prediction is made by P_3 for x_3.

We show that algorithm $\mathcal{L}_{\text{SLAP}}$ guarantees predictors with optimal completeness in such stratified settings.

Theorem 7 (Completeness in SLAP Setting). *Consider any integer value d, a d-stratified list $\mathcal{S}$ of complete assignments, and an associated list $\mathcal{S}^*$. Assume that algorithm $\mathcal{L}_{SLAP}$ has oracle access to a focused base algorithm $\mathcal{L}_{base}$ and returns a policy $\mathbb{P}$ of depth d. Then: $\mathbb{P}$ is complete on all of $\mathcal{S}^*$.*

Proof. The claim follows by straightforward induction. □

Theorems 6–7 offer a second algorithm for chaining predictors, with bounded unsoundness, and optimal completeness when in stratified settings. The algorithm is efficient for constant-depth policies. Unlike the first algorithm of Theorems 3–4, the second algorithm is more realistic in assuming a base algorithm with guarantees w.r.t. $\mathcal{S}^*$, instead of $\mathcal{S}$.

Although it remains open whether the efficiency guarantees of Theorem 6 can be extended to hold for super-constant values of d, several natural modifications to algorithm $\mathcal{L}_{\text{SLAP}}$ fail to resolve this question: *(i)* Employing base algorithms that produce predictors by expending sub-linear resources in the inverse error parameter $1/\varepsilon_k$ are excluded (Ehrenfeucht et al. 1989). *(ii)* Base algorithms that *identify* their inaccurate predictions (Rivest and Sloan 1994) employ a strategy that does not scale up to expressive predictors, and fails to work when instead of complete assignments one has access to arbitrary assignments. *(iii)* Training all predictors on a common set of assignments (Valiant 2000) corresponds, effectively, to the FLTC scenario and not to the SLAP scenario considered herein. *(iv)* Appealing to noise-resilient base algorithms also fails,[5] as the unsound predictions of the predictors in earlier layers of $\mathbb{P}$ can be shown (Michael 2008) to correspond to malicious noise for the predictors in later layers of $\mathbb{P}$. Such malicious noise is known to be especially intolerable by base algorithms (Kearns and Li 1993).

Overall, then, evidence suggests that interleaving learning and prediction is a solid approach when seeking to improve

[4] Under this second scenario, algorithm $\mathcal{L}_{\text{SLAP}}$ can no longer be assumed to explicitly compute the list $\mathcal{S}^*_k := \mathbb{P}_k(\mathcal{S}^*_{k-1})$ during the last step of the k-th iteration. This can be easily dealt with, by lazily computing parts of $\mathcal{S}^*_k$ if and when needed for subsequent steps.

[5] We are not suggesting that noise-resilient (or otherwise robust) base algorithms be avoided altogether, only that doing so does not improve asymptotically the achieved chaining depth. One can investigate how the noise-resilience of base algorithms relates to that of $\mathcal{L}_{\text{SLAP}}$ (as done for soundness and completeness). We have not considered noise in this work to avoid obscuring the main points.

completeness without sacrificing soundness. Even in scenarios where d-stratification is not a priori guaranteed, calling $\mathcal{L}_{\text{SLAP}}$ with increasing values for d allows the effective trade-off of time for possibly better, but never worse, performance.

Explicit Completeness

We have established in certain settings *explicit* bounds on the unsoundness of our considered policies, but only a type of optimal, not always specific, degree of completeness. Why not expect algorithms to return policies with *explicit* bounds on their degree of completeness? The answer is two-fold:

First, completeness depends externally on the masking process used to produce list $\mathcal{S}^*$ from list $\mathcal{S}$. In the interest of generality — and this happens to be the case in certain natural settings — no assumptions on how this masking process operates have been (or can be) made. The more information is missing in list $\mathcal{S}^*$ the more a policy is forced to abstain, as long as one insists, as we have done herein, that soundness should not be severely compromised to boost completeness.

Second, even if one can construct efficiently a policy that achieves either perfect soundness or perfect completeness, and even if sufficient information is available in list $\mathcal{S}^*$ for some policy to make perfectly sound and perfectly complete predictions, it is still not possible, in general, to achieve efficiency, soundness, and completeness together. We establish next a barrier to efficiently constructing a policy that simultaneously has soundness that is slightly better than a policy that predicts with constant predictors, and has completeness that is slightly better than a policy that makes no predictions.

Definition 2 (Properties of Policy-Learning Algorithms). *In what follows, **parameters** are: an integer $n > 1$, a list $\mathcal{S}$ of complete assignments over a set $\mathcal{A}$ of n attributes, and a list $\mathcal{S}^*$ of assignments. A policy-learning algorithm $\mathcal{L}$:*

*(i) is **computationally efficient** if there is a polynomial $r(\cdot,\cdot)$ such that for every choice of the parameters, $\mathcal{L}$ takes $\mathcal{A}$ and $\mathcal{S}^*$ as input, and returns a policy $\mathbb{P}$ in time $r(n, |\mathcal{S}^*|)$;*

*(ii) achieves **non-trivial soundness** if there is a polynomial $s(\cdot)$ such that for every choice of the parameters, $\mathcal{L}$ takes $\mathcal{A}$ and $\mathcal{S}^*$ as input, and returns a policy $\mathbb{P}$ that is sound on an $1/2 + 1/s(n)$ fraction of $\mathcal{S}^*$ against $\mathcal{S}$;*

*(iii) achieves **non-trivial completeness** if there is a polynomial $c(\cdot)$ such that for every choice of the parameters, $\mathcal{L}$ takes $\mathcal{A}$ and $\mathcal{S}^*$ as input, and returns a policy $\mathbb{P}$ that is complete on an $1/c(n)$ fraction of $\mathcal{S}^*$.*

Given these desirable properties for a policy-learning algorithm, the following observations are immediate: The trivial algorithm that returns the empty policy is computationally efficient and perfectly sound. The trivial algorithm that returns a policy comprising one constant predictor for each attribute is computationally efficient and perfectly complete.

Achieving perfect soundness and perfect completeness is easily shown to be impossible if information is hidden adversarially. Assume, therefore, that we are in a non-adversarial setting, where this third combination of two desirable properties is achievable. It it still the case that achieving all three properties simultaneously is impossible, even if only non-trivial soundness and non-trivial completeness is required.

Theorem 8 (Barriers to Explicit Completeness). *Under typical cryptographic assumptions (Kearns and Vazirani 1994), no algorithm that returns policies is computationally efficient and achieves non-trivial soundness and non-trivial completeness. This is so even if there is a perfectly sound and perfectly complete policy, and such policy can be approximated by a (possibly inefficient) PAC learning algorithm.*

Proof. Assume that the claim does not hold for an algorithm $\mathcal{L}$, and consider the polynomials $r(\cdot,\cdot)$, $s(\cdot)$, $c(\cdot)$ from Definition 2. Consider any concept class known not to be weakly PAC-learnable under standard cryptographic assumptions (Kearns and Vazirani 1994). For any function in the class, and every distribution $\mathcal{D}$ over examples, construct a list $\mathcal{S}$ of $m \cdot c(n) \cdot (s(n)+1)$ complete assignments by sampling examples from $\mathcal{D}$ and assigning to x_1 their label according to f. Construct the list $\mathcal{S}^*$ by masking attribute x_1 in each assignment except with probability $1/(c(n) \cdot (s(n)+1))$.

The policy that includes only predictor f for x_1 is sound on all of $\mathcal{S}^*$ against $\mathcal{S}$, and complete on all of $\mathcal{S}^*$. In addition, choosing m to be sufficiently large (but still polynomial in n and the PAC error parameters $1/\varepsilon$ and $1/\delta$), PAC learning can be achieved by finding any function in the concept class that is consistent with assignments in $\mathcal{S}^*$ that are complete.

Running algorithm $\mathcal{L}$ on $\mathcal{S}^*$ for time $r(n, |\mathcal{S}^*|)$ will return a policy $\mathbb{P}$ that is non-trivially complete on $\mathcal{S}^*$, and, therefore, complete on $|\mathcal{S}^*|/c(n)$ assignments, which is more than those already complete in $\mathcal{S}^*$. Then, $\mathbb{P}$ includes a predictor g for x_1 that makes predictions on every assignment in $\mathcal{S}^*$. Policy $\mathbb{P}$ is also non-trivially sound on $\mathcal{S}^*$ against $\mathcal{S}$, and, therefore, sound on at least $(1/2 + 1/s(n)) \cdot |\mathcal{S}^*|$ assignments against $\mathcal{S}$. Ignoring the assignments that are complete in $\mathcal{S}^*$ (and on which the prediction of g is qualified), g predicts correctly with probability $((1/2 + 1/s(n)) \cdot |\mathcal{S}^*| - m)/(|\mathcal{S}^*| - m) \geq 1/2 + 1/2s(n)$. Since $|\mathcal{S}^*|$ is polynomial, so is $r(n, |\mathcal{S}^*|)$, and therefore the concept class is weakly PAC-learnable; a contradiction, and the claim follows. □

Applications and Implications

We have chosen to motivate and present our framework in the context of a single list $\mathcal{S}^*$ of observed assignments that one seeks to complete further. The choice mostly serves to not clatter our central claims: *(i)* chaining predictors is provably beneficial, and *(ii)* the benefits can be realized by (and only by — at least for certain natural strategies) an iterative learning and prediction process. The same ideas and formal results presented herein can be applied to other settings that have been traditionally studied in Machine Learning.

The first point worth making is that the framework applies equally well to the traditional setting of *distinct* training and testing sets. There, $\mathcal{S}^*$ plays the role of the testing set, while another list $\mathcal{T}$ of assignments plays the role of the training set accessed by the base learning algorithm to provide the sought guarantees on each predictor P_i. The only modification needed to the framework is that of giving access to $\mathcal{T}$ instead of $\mathcal{S}^*$ to the learning algorithms. In the first scenario of the FLTC setting, $\mathcal{T}$ is a list of complete assignments drawn from the same distribution as those in $\mathcal{S}$. In the second scenario of the FLTC setting and in the SLAP setting, $\mathcal{T}$ is a list of assignments drawn from the same distribution as those in $\mathcal{S}^*$. Existing models of learning from complete (Valiant

1984; Michael 2011a) or observed (Michael 2010) assignments, show that these choices of $\mathcal{T}$ suffice to efficiently and reliably obtain the guarantees on P_i assumed herein.

A second important point is that the central claims of this work apply also to the traditional supervised classification / regression setting, where certain, possibly continuous and / or unobserved, features are used to inform a prediction on a *single* target attribute $x_t \in \mathcal{A}$, which is observed during training but not during testing. First, note that the proofs of our results, other than those in the SLAP section, go through even if we insist that some target attribute x_t is never observed in the testing set $\mathcal{S}^*$, and always observed in the training set $\mathcal{T}$ that one uses to produce each predictor P_i.

For the results in the SLAP section, we argue as follows: Assume that some target attribute x_t is always observed both in the testing set $\mathcal{S}^*$ and in the training set $\mathcal{T}$. Theorem 6 holds for this choice of $\mathcal{S}^*$. Now, mask x_t in all assignments in $\mathcal{S}^*$ to obtain $\overline{\mathcal{S}^*}$. The change from $\mathcal{S}^*$ to $\overline{\mathcal{S}^*}$ affects only the completeness of $\mathbb{P}$, and hence Theorem 6 still holds. To avoid affecting completeness also, and to ensure that Theorem 7 holds, choose a partitioning with $x_t \in \mathcal{A}_d$, so that no predictor can base its prediction on the value of x_t, and so that the completeness guarantees on $\mathcal{S}^*$ and on $\overline{\mathcal{S}^*}$ coincide.

So, chaining predictors is beneficial for supervised learning, and SLAP helps realize these benefits. Seeking to develop classification / regression algorithms that are robust to missing information is bound to either produce non-classical predictors or result in worse performance than SLAP.

A final point relates to using predictors to complete missing values, but never to override observed values. This view aligns with KRR work that treats observations as indefeasible, and as exogenously qualifying rules (Kakas, Michael, and Miller 2011; Michael 2013b). Although we have chosen to adopt this treatment, we note that predictors that override observed values would also be useful, especially in settings where a prediction might possibly correct an observed noisy value. This choice is inconsequential for our results herein.

Conclusions and Related Work

This work has initiated a formal investigation of interleaving learning and prediction with multiple predictors, examining the guarantees one may get under minimum assumptions.

Among works that interleave learning and prediction, Inductive Logic Programming (Muggleton 1991) typically differs from this work in: *(i)* placing less emphasis on computational efficiency; *(ii)* seeking to fit available data instead of providing statistical guarantees during prediction; *(iii)* producing predictors that never abstain (under the Closed World Assumption) instead of bounding their incompleteness.

Some form of interleaved learning and prediction is found in Transformation-Based Learning (Brill 1993), although little emphasis seems to have been placed on its formal analysis. TBL predictors aim to correct initial heuristic predictions rather than make predictions only when it is justified.

Rivest and Sloan (1994) learn predictors in a layered manner, with subsequent predictors being trained on the predictions of earlier ones. This resembles the SLAP setting, but their approach does not extend to assignments with missing information. Also, unlike us, they achieve super-constant depth of chaining, at the expense, however, of considering severely restricted classes of predictors. Attempting to integrate their and our approach is certainly worth looking into.

Valiant (2000) considers the parallel learning of predictors before their chaining, as in the first scenario under the FLTC setting. As in our approach, it is the availability of base algorithms with guarantees on complete assignments that allows the decoupling of learning and prediction. Both of our works place emphasis on efficiency and predictive guarantees. The relational learning and the two types of missing information that he employs can be carried over to our framework.

Read et al. (2009) investigate a version of multi-label classification with predictions for a label being available when training predictors for other labels. To account for their suboptimal random ordering of the predictors, they produce ensembles of such orderings. Instead of a distinguished fixed set of labels, we allow predictions on all attributes that may happen to be masked. Instead of an ensemble of orderings, we produce a single universal ordering. Finally, our emphasis is on the formal analysis and understanding of when SLAP is provably useful. To that end, the principled treatment of missing information and abstentions is essential.

Chaining learned predictors differs from learning circuits, neural networks, or other expressive functions, where the values of internal nodes are never observed during the training (or testing) phase. Furthermore, missing information (on internal nodes) is only implicitly completed during prediction, whereas we do so explicitly, and also allow abstentions.

In the spirit of implicit reasoning, Juba (2013) considers a setting of partially observed assignments closely following the one of our work (Michael 2010), and chains learned rules without, however, explicitly representing them. It remains an intriguing question to understand the comparative advantages of the implicit and explicit representations of learned rules, in terms of the performance (soundness and completeness) achieved when they are chained to draw inferences.

On the empirical front, the benefits of chaining have been demonstrated on the massive-scale extraction of commonsense knowledge from raw text (Michael and Valiant 2008) and on other language-related tasks (Doppa et al. 2011). The results obtained in this work can be seen to offer a formal basis for these experimental results, while the latter exemplify the applicability of our framework in real-world settings.

Some arguments and empirical evidence in other contexts also point to the benefits of chaining rules (Dietterich 2000; Valiant 2006), for reasons ultimately related to limitations of the base algorithms: hypothesis class expressivity, statistical and computational constraints, data availability. This work *formally* ties the benefits of chaining to a limitation of rules (cf. Lemma 1), and seeks ways to realize them. Certain ideas from our work have been argued (Michael 2009) to be important in the Recognizing Textual Entailment task (Dagan, Glickman, and Magnini 2005). The design of base algorithms has been considered (Michael 2010), with applications (Michael 2013a), and open problems (Michael 2011b).

The goal of enhancing completeness resembles the goal of "boosting" from Machine Learning, where one seeks to improve a weakly sound predictor. A typical boosting approach (Schapire 1990) proceeds by learning a sequence of predic-

tors, each trained on those assignments on which the preceding predictor was unsound. At this abstract level, the parallels with our work are immediate: SLAP attempts to improve a weakly complete predictor set by learning a sequence of such predictor sets, each trained on those assignments on which the preceding predictor set was incomplete. Of course there are critical differences as well, not the least of which being that an unsound prediction cannot be identified during testing, whereas an incomplete one can. Understanding the parallels and differences in more technical detail may allow boosting techniques to be adapted and adopted for the problem of improving the completeness of predictors.

Regarding discussions pointing out that Machine Learning research has focused on the construction of flat (in our defined sense) predictors (Dietterich 2003), this work offers a glimpse of what one may expect when moving to a chained (and more structured) use of predictors. The work presented herein offers a concrete basis upon which a large scale investigation of this intriguing phenomenon may be carried out.

References

Brill, E. D. 1993. *A Corpus-Based Approach to Language Learning*. Ph.D. Dissertation, University of Pennsylvania, U.S.A.

Dagan, I.; Glickman, O.; and Magnini, B. 2005. The PASCAL Recognizing Textual Entailment Challenge. In *Proc. of 1st PASCAL Challenges Workshop for Recognizing Textual Entailment (RTE'05)*, 1–8.

Dietterich, T. G. 2000. Ensemble Methods in Machine Learning. In *Proc. of 1st International Workshop on Multiple Classifier Systems (MCS'00)*, 1–15.

Dietterich, T. G. 2003. Learning and Reasoning. *Unpublished Manuscript*.

Doppa, J. R.; Nasresfahani, M.; Sorower, M. S.; Irvine, J.; Dietterich, T. G.; Fern, X.; and Tadepalli, P. 2011. Learning Rules from Incomplete Examples via Observation Models. In *Proc. of Workshop on Learning by Reading and its Applications in Intelligence Question-Answering (FAM-LbR/KRAQ'11)*.

Ehrenfeucht, A.; Haussler, D.; Kearns, M. J.; and Valiant, L. G. 1989. A General Lower Bound on the Number of Examples Needed for Learning. *Information and Computation* 82(3):247–261.

Juba, B. 2013. Implicit Learning of Common Sense for Reasoning. In *Proc. of 23rd International Joint Conference on Artificial Intelligence (IJCAI'13)*, 939–946.

Kakas, A.; Michael, L.; and Miller, R. 2011. Modular-E and the Role of Elaboration Tolerance in Solving the Qualification Problem. *Artificial Intelligence* 175(1):49–78.

Kearns, M. J., and Li, M. 1993. Learning in the Presence of Malicious Errors. *SIAM Journal on Computing* 22(4):807–837.

Kearns, M. J., and Vazirani, U. V. 1994. *An Introduction to Computational Learning Theory*. The MIT Press.

Knuth, D. E. 2011. *The Art of Computer Programming, Volume 4A: Combinatorial Algorithms*. Addison-Wesley.

Michael, L., and Valiant, L. G. 2008. A First Experimental Demostration of Massive Knowledge Infusion. In *Proc. of 11th International Conference on Principles of Knowledge Representation and Reasoning (KR'08)*, 378–388.

Michael, L. 2008. *Autodidactic Learning and Reasoning*. Ph.D. Dissertation, Harvard University, U.S.A.

Michael, L. 2009. Reading Between the Lines. In *Proc. of 21st International Joint Conference on Artificial Intelligence (IJCAI'09)*, 1525–1530.

Michael, L. 2010. Partial Observability and Learnability. *Artificial Intelligence* 174(11):639–669.

Michael, L. 2011a. Causal Learnability. In *Proc. of 22nd International Joint Conference on Artificial Intelligence (IJCAI'11)*, 1014–1020.

Michael, L. 2011b. Missing Information Impedements to Learnability. In *Proc. of 24th Annual Conference on Learning Theory (COLT'11), JMLR: Workshop and Conference Proceedings*, volume 19, 825–827.

Michael, L. 2013a. Machines with WebSense. In *Proc. of 11th International Symposium on Logical Formalizations of Commonsense Reasoning (Commonsense'13)*.

Michael, L. 2013b. Story Understanding... Calculemus! In *Proc. of 11th International Symposium on Logical Formalizations of Commonsense Reasoning (Commonsense'13)*.

Mitchell, T. M. 1997. *Machine Learning*. McGraw Hill Series in Computer Science. McGraw-Hill.

Muggleton, S. H. 1991. Inductive Logic Programming. *New Generation Computing* 8(4):295–318.

Read, J.; Pfahringer, B.; Holmes, G.; and Frank, E. 2009. Classifier Chains for Multi-label Classification. In *Proc. of European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML-PKDD'09)*, 254–269.

Reiter, R. 1980. A Logic for Default Reasoning. *Artificial Intelligence* 13(1–2):81–132.

Rivest, R. L., and Sloan, R. 1994. A Formal Model of Hierarchical Concept Learning. *Information and Computation* 114(1):88–114.

Schapire, R. E. 1990. The Strength of Weak Learnability. *Machine Learning* 5(2):197–227.

Schuurmans, D., and Greiner, R. 1994. Learning Default Concepts. In *Proc. of 10th Canadian Conference on Artificial Intelligence (CSCSI'94)*, 99–106.

Valiant, L. G. 1984. A Theory of the Learnable. *Communications of the ACM* 27:1134–1142.

Valiant, L. G. 2000. Robust Logics. *Artificial Intelligence* 117(2):231–253.

Valiant, L. G. 2006. Knowledge Infusion. In *Proc. of 21st National Conference on Artificial Intelligence (AAAI'06)*, 1546–1551.

Valiant, L. G. 2013. *Probably Approximately Correct: Nature's Algorithms for Learning and Prospering in a Complex World*. Basic Books.