7 Inductive Learning from Partial Examples

The chapter extends the learning task, introduced in Chapter 4, to handle training examples that consist of *partial* truth-assignments, i.e., that only specify the truth values of a subset of the attributes. We will first introduce our model of partial observations, and then, in Section 7.2, consider a spectrum of possible guarantees that algorithms for learning from partial examples might satisfy. We will observe that learning nontrivial properties inherently requires assumptions. We will see in Section 7.3 that as long as the process that produces our observations of the partial examples is sufficiently benign (features *“bounded concealment”*), it is possible to learn some simple classes of representations in a strong sense, but then in Section 7.4, also that some other classes (in particular, decision lists) are no longer properly learnable. Finally, in Section 7.5, we will see that if the observations are still more benign (*“revealing”*) then we can again learn some of our basic classes such as k-CNF and k-DNF formulas. In the next chapter, we will incorporate this partial information model into the combined learning and reasoning task that we introduced in Chapter 6.

7.1 Masking processes.

Learning is about making predictions: a learned rule allows us to predict whether or not a new world has the property of interest. Therefore, learning is only helpful when we do not already know whether or not that property is true: some information about the world must be missing, that is supplied by the rule. Otherwise, the rule’s prediction is useless.

Up until now, those properties have been *“exogenous,”* i.e., not part of the world, not included among the propositional attributes that describe the possible worlds. The examples we used for learning had been supposed to be labeled by an expert, where those labels were not among the attributes we had at our disposal for making predictions in the future. But we might also seek to learn about observable properties of the world from our own experience, for example, that when there is smoke there must be fire, or that if the lights in a shop are off, then the shop is not open for business. These rules are useful because some information about the world is not immediately available to us.

We model these situations by supposing that some of the attributes describing our possible worlds are not provided to us:

Definition 7.1 (Masking functions and partial examples) For possible worlds given by n Boolean attributes, a *partial example* is an element of $\{0, 1, *\}^n$, indicating for each ith attribute whether it is respectively false, true, or *missing*. A partial example $\rho \in \{0, 1, *\}^n$ is *consistent* with a possible world $\vec{w} \in \{0, 1\}^n$ if whenever $\rho_i \neq *$, $\rho_i = w_i$. A *masking function* is a function $m : \{0, 1\}^n \to \{0, 1, *\}^n$ such that $m(\vec{w})$ is consistent with $\vec{w}$, for each possible world $\vec{w}$.

Example 7.1: The original setting of PAC-learning is captured by supposing that the test worlds are obtained from a masking function in which the label attributes are hidden. Let's suppose that the final attribute is the label. Then during training we obtain examples of complete descriptions of the possible worlds, and after learning, we observe partially specified worlds $m_{\text{nature}}(\vec{w})$ such that for $i = 1, \ldots, n-1$, $m_{\text{nature}}(\vec{w})_i = w_i$, and $m_{\text{nature}}(\vec{w})_n = *$. So, the learned rule $h : \{0, 1\}^{n-1} \to \{0, 1\}$ should satisfy $h(W_1 \ldots, W_{n-1}) = W_n$ with high probability for possible worlds $\vec{W}$ drawn from the same distribution. h can still be computed even if W_n is not observed, thus allowing us to predict W_n after training.

As a natural example, suppose we wish to know whether or not a mushroom is poisonous based on its appearance. Naïvely this is not apparent: the property is missing from our observation. But, sometimes we might have access to an expert who tells us whether or not the mushrooms we find are poisonous, and based on the expert's advice, we may be able to learn to classify mushrooms ourselves.

In other cases, parts of the world are missing from our observation because we are situated in the world ourselves:

Example 7.2: We return once again to the box-moving robot introduced in Sections 2.1 and 2.4, and discussed further in Example 3.3. Attributes X_1, X_2, and X_3 corresponded to the robot being in a hallway, first room, and second room respectively, while attributes X_4, X_5, and X_6 corresponded to a box being present, respectively again, in a hallway, first room, and second room. *From the robot's perspective,* when X_1 is true, we may suppose that the robot cannot see whether a box is in the rooms, only the hallway. So, we would obtain a masking function such that in examples where X_1 is true, X_5 and X_6 are missing while X_4 is observed; and similarly, when X_2 is true, X_4 and X_6 are missing since the robot cannot observe what is in the hallway or the second room from the vantage point of the first room, while X_5 is now observed. Observe that a masking function can indeed use some attribute values to determine which other attributes to omit.

In other cases still, the reporting may actually be adversarial:

Example 7.3: Suppose we are conducting a survey, and X_1 is a politically loaded question, in an environment disapproving of the opinion $X_1 = 0$. Then it may be that whenever $X_1 = 0$ for a respondent, the response to X_1 is omitted. We'd then only

obtain examples where $m_{\text{response}}(\vec{x})_1 = 1$, and examples where $m_{\text{response}}(\vec{x}')_1 = *$ (whenever $x'_1 = 0$), *not* examples where $m_{\text{response}}(\vec{x}')_1 = 0$. Here, m_{response} unfortunately omits the value of the first attribute depending on that value itself.

In all of the above examples, which attributes were present, and which were missing, was fully determined by (the values of the attributes describing) the world itself. But in some particularly interesting cases, we might *sometimes* be able to observe the values and at other times not, determined by factors that we have not modeled and cannot predict. As with the possible worlds on which we wish to make our predictions, we will model this variation in missingness as stochastic and independent across examples:

Definition 7.2 (Masking processes and partial observations) A *masking process* is a masking-function valued random variable. We will suppose that we obtain partial examples as follows: there is a random variable $\vec{X}$ taking values in $\{0,1\}^n$ (i.e., over possible worlds), and a masking process M such that $\vec{X}$ and M are independent. Our partial observations are then given by independent draws of the random variable $M(\vec{X})$, i.e., first drawing $\vec{x}$ from the distribution of $\vec{X}$, drawing a masking function m from the distribution of M (independent of $\vec{x}$), and then observing $\rho = m(\vec{x})$. We'll generally suppose that we obtain a sample of partial observations $\rho^{(1)}, \rho^{(2)}, \rho^{(3)}, \ldots$ that are independently drawn from this distribution. The possible worlds $\vec{x}^{(1)}, \vec{x}^{(2)}, \vec{x}^{(3)}, \ldots$ are the respective *ground truth* worlds for these partial observations.

Example 7.4: Suppose that sometimes on our mushroom foraging expeditions an expert is present, and not at other times. We assume that whether or not the expert is present is independent of the mushrooms we encounter, and hence our observations of whether or not the mushrooms we find are poisonous are independent of the fact of the matter. We'd then obtain a masking process $M_{\text{expedition}}$ that with some probability p takes value m_{expert} such that for all $\vec{x}$, $m_{\text{expert}}(\vec{x}) = \vec{x}$ (the expert can tell us everything we wanted to know about our mushrooms) and otherwise, with probability $1-p$, $M_{\text{expedition}}$ takes value m_{nature} as described in Example 7.1. So, X_n – whether or not the mushroom is poisonous – is observed with probability p and missing with probability $1-p$, independent of any of the mushroom's attributes.
Since the distribution of $X_1 \ldots, X_{n-1}$ here is independent of whether or not X_n is observed in our partial observations, We can simply collect all of the examples where X_n is observed, and treat the value of $x_n^{(i)}$ as a label for an example given by $(x_1^{(i)}, \ldots, x_{n-1}^{(i)})$. Then a PAC-learning algorithm provides a rule h such that

$$\Pr[h(X_1 \ldots, X_{n-1}) = X_n | M_{\text{expedition}}(\vec{X}) = *] =$$
$$\Pr[h(X_1 \ldots, X_{n-1}) = X_n | M_{\text{expedition}}(\vec{X}) \neq *] \geq 1 - \epsilon$$

with probability $1-\delta$ given enough such examples, since the distributions

$$\vec{X} | [M_{\text{expedition}}(\vec{X}) = *] \qquad \text{and} \qquad \vec{X} | [M_{\text{expedition}}(\vec{X}) \neq *]$$

are identical by our independence assumption. So, we would be justified in using the learned rule h to decide which ones are edible.

We thus see that the framework of masking processes captures a natural formal setting in which PAC-learning is useful, while testing and training examples are drawn from the same distribution on partial examples. But, it also captures some more interesting settings in which some of the attributes that we might wish to use for prediction are themselves missing:

Example 7.5: Suppose that our observations are given by sensors that communicate wirelessly with a base station. Due to atmospheric conditions or power failures, some of the sensor values may be dropped. (We'll suppose that error-correction is used so that we can treat any received reports as correct readings.) In a particularly simple setting, these failures might be independent of one another. So, for a parameter $\mu \in [0, 1]$, we consider the masking process M_μ such that for each ith attribute, for all $\vec{x}$ $M_\mu(\vec{x})_i = x_i$ with probability μ and $M_\mu(\vec{x})_i = *$ with probability $1 - \mu$, mutually independent from one another (so the missingness pattern on any subset of the attributes is independent of the missingness pattern on any other disjoint subset of the attributes).
In this case, we might wish to learn multiple predictors on different subsets of the attributes, that would sometimes enable us to predict missing values from some others that we have observed. Again, since the distribution of $\vec{X}$ is indepdendent of whether $M_\mu(\vec{X})_i = *$ or not (indeed, as is moreover the joint distribution of X_i and $(M_\mu(\vec{X})_j : j \neq i)$), any rule we learn that predicts X_i from $(M_\mu(\vec{X})_j : j \neq i)$, using values of X_i as labels on the examples where it is observed, will generalize due to our strong independence assumption.

The previous examples featured masking processes that were easy to describe in complete detail. But, much as with our probability distributions on possible worlds, there is in general a complex relationship between the state of the world and what information is available to us; that is, in general we *do not* have simple, accurate models of the masking process, as the next couple of examples illustrate.

Example 7.6: Text, whether a sentence or a paragraph, often gives a partial description of a possible world. *Which* features are specified and which are omitted is generally determined the author's intent; we still suppose that there is a complete possible world that is consistent with the text, but only that a subset of the attributes are specified by the sentence.
For example, a sentence such as "Bob held a barbecue in his backyard." specifies literally that there is someone named Bob who has a backyard, and that in this backyard there is a barbecue (party), for which Bob is the host. Based on our common sense understanding of such concepts, we may presume that there is meat being cooked on a grill over an open flame, that there are other people who are guests in attendance at this party, and that it is not raining. Even this description leaves many

marital status	age	smoking habits	drinking habits	blood pressure	cholesterol level	exercise habits
single	-	-	3/week	130/90	-	-
married	49	cigarettes, 10/day	-	-	high	-
-	32	-	none	-	normal	jogging daily
divorced	-	-	2/day	145/100	-	-
-	68	cigars, 3/day	-		-	none
single	-	none	-	125/80	slightly elevated	-
-	25	-	rarely	-	normal	gym 3×/week

Table 7.1 A portion of a hypothetical data set of patient records. We wish to predict the contents of the black cell.

features of the scene unspecified. But, we may consider a masking process M_{author} that takes a complete description of a scene, and here hides all but the few attributes that capture the literal meaning of "Bob held a barbecue in his backyard." We do not have a simple mathematical description of the masking process any more than we have a simple description of the distribution on possible worlds that the author seeks to describe. Nevertheless, for the purposes of interpreting text, it is useful to consider the relationship between what is written and what is likely true about the world being described by that text. The input here is naturally formulated as a partial example obtained from the masking process M_{author} induced by which attributes are reported in text by the various authors represented in a corpus of text.
Note that this choice of what to report is highly biased: a murder is much more likely to be reported than the fact that individuals in a scene are breathing. If we do not account for this bias in reporting, we may incorrectly conclude that murder is more common than breathing in human society.

Example 7.7: We consider a medical domain, in which a "possible world" corresponds to the state of a patient. Some attributes of each patient may be noted in a medical record, based on which tests were performed and which questions were asked, but this reporting of attributes is generally incomplete, resulting in a data set such as illustrated in Table 7.1, where each row corresponds to a different patient. Often, tests are only performed (and the corresponding data recorded) based on some observed features of the patient's condition; in other cases, a test may be considered

too risky to perform based on the patient's observed condition, and is thus omitted from the record. More generally, what is considered and what is omitted is due to a physician's discretion for a given patient at a given point in time. We may naturally model the process by which a given patient's features are included (or not) in the record by a masking process M_{record} that takes a complete description of the patient's attributes, and selects which ones are actually recorded in the data.
We may be interested in learning to predict the missing features, such as in the black cell in Table 7.1. Although the data is incomplete, we might predict that the patient's blood pressure is high based on the available information in that row. But, this task is complicated by the delicate relationship between reporting and the ground truth. Unfortunately, the lack of independence between the values and the missingness pattern makes it challenging to draw sound conclusions about the missing attributes, much as in the survey example (Example 7.3).

We therefore primarily focus on a setting in which the masking process, like the distribution on possible worlds, is taken to be arbitrary and unknown. This leads to some obvious challenges, such as a masking process that omits *all* of the features on every possible world. Meaningful learning is not possible with such a masking process since the partial examples are identical (empty) for all distributions on possible worlds. In the rest of this chapter, we'll consider some minimal conditions and weaker notions of success, under which learning from partial examples is possible.

7.2 Consistency, accuracy, and concealment.

There are two reasons for which masking all the features on every possible world would prevent meaningful learning: (R1) first, the insufficiency of information in an observed possible world to support the drawing of a definite inference by some particular hypothesis; (R2) second, the insufficiency of statistical evidence in the observed possible worlds to support the learning of a hypothesis that makes accurate predictions.

In a sense, both reasons are beyond the control of a learner, and this should be accounted for in our attempt to model learning in such partially observable situations. Starting with the first reason, we will allow a learned hypothesis to abstain — without a penalty — from making predictions when it can truly not determine a definite prediction from whatever information it has access to.

Definition 7.3 (Abstentions and Consistency Conflicts) Consider a function $h : \{0,1\}^{n-1} \to \{0,1\}$ that we will use to predict the tth attribute of a possible world. We let $\vec{w}_{-t}$ denote the possible world $\vec{w}$ with the tth attribute omitted—thus, a possible world given by $n-1$ attributes, where h is correct on the world $\vec{w}$ if $h(\vec{w}_{-t}) = w_t$. If h is correct on all $\vec{w}$ in the support of a distribution D, we say h *predicts* w_t w.r.t. D. Similarly, for a partial observation $\rho \in \{0,1,*\}^n$, ρ_{-t} omits the tth attribute. We lift the function h to apply on partial observations ρ as follows: If for all possible worlds

$\vec{w}_{-t}$ that are consistent with ρ_{-t} $h(\vec{w}_{-t})$ takes the same value $b \in \{0, 1\}$, $h(\rho_{-t}) = b$; otherwise, $h(\rho_{-t})$ is said to *abstain*. We say that $h(\rho_{-t})$ has a *consistency conflict* with ρ if $h(\rho_{-t})$ does not abstain, $\rho_t \neq *$, and $h(\rho_{-t}) \neq \rho_t$.

Example 7.8: Consider again the medical domain example with the data illustrated in Table 7.1, and assume that our hypothesis h is applied on each row of data to predict each patient's blood pressure. Assume, further, that h critically depends on knowledge of either the exercise habits or the smoking habits of a patient to be able to make a definite prediction. Then, h abstains from making a prediction in the first and fourth rows, and, therefore, does not have a consistency conflict in those cases. Furthermore, and irrespective of the predictions of h, the information on the blood pressure of patients is missing in rows two, three, five, and seven, and hence there is no consistency conflict in those cases also. The only consistency conflict may occur in row six, exactly in the case where h will predict anything other than 125/80.

Consistency conflicts capture situations where what we observe and what we predict for attribute X_t are in explicit disagreement. This evaluation criterion can be used to define a minimal notion of success for learning from partial examples as follows:

Definition 7.4 The *consistent learning* task for the class of representations C using the class $\mathcal{H}$ is the following. For any distribution D such that there is a representation $c \in C$ that predicts w_t w.r.t. D, and any masking process M on possible worlds on n attributes, given a list of partial observations drawn independently from $M(D)$, return a hypothesis $h \in \mathcal{H}$ such that with probability $1 - \delta$,

$$\Pr_{\rho \in M(D)} [h(\rho_{-t}) \text{ does not have a consistency conflict with } \rho] \geq 1 - \epsilon.$$

If $C = \mathcal{H}$, we say that the algorithm is *proper*. If the algorithm succeeds at consistent learning for the class of representations using a number of partial observations that is polynomially in n, $1/\delta$, and $1/\epsilon$ we will say that the algorithm is *sample efficient*, and if the algorithm's running time is polynomial in n, $1/\delta$, and $1/\epsilon$, then we simply say it is *efficient*.

Is this notion of success for learning from partial observations appropriate? First, let us consider whether the requirement for learning is too demanding, given that it should be able to cope with arbitrary masking processes, including possibly adversarial ones. Analogously to our approach for coping with arbitrary distributions over possible worlds, it is also the case here that this requirement is compensated by the assumption that D and M are fixed across training and testing. Thus, exactly in those situations where learning might be challenged because of reason (R2) — too little information during training to acquire sufficient statistical evidence for learning a good hypothesis — the learned hypothesis can abstain more frequently because of reason (R1).

Could it be, on the other hand, that the requirement for learning is too relaxed, and can be met simply by learning a hypothesis that abstains too much to avoid consistency conflicts? By definition, hypotheses cannot abstain freely, but only when their value

cannot be uniquely determined by the possible worlds with which a partial observation is consistent. Thus, the extent to which such a strategy would work is determined by the amount of information that is made available by the specific masking process. Again, this is analogous to the difficulty of learning with an arbitrary distribution over possible worlds, which could vary depending on the specific distribution.

One could argue that consistent learning is the best learning criterion we could hope for, given that the sensory inputs of humans or robots are the only means we have to access our environment, and it is only against these inputs, partial as they might be, that we can empirically evaluate the predictions coming from our learning hypotheses. Nonetheless, sidestepping any such considerations, we can still attempt to define a second notion of success for learning from partial observations, that is closer to what we have considered previously.

Recall Example 7.4, where an external expert is providing labels for some of our learning examples. We had argued earlier that if these labels are provided independently of the possible world that we are partially observing, then the learning goal can be that of accurately predicting the expert's label, even if it is not provided. We can capture this requirement as follows:

Definition 7.5 (Abstentions and Accuracy Conflicts) Consider a function $h : \{0,1\}^{n-1} \to \{0,1\}$ that predicts the tth attribute of a possible world, lifted to apply on partial observations $\rho \in \{0,1,*\}^n$ as in the case of Definition 7.3. $h(\rho_{-t})$ has an *accuracy conflict* with a possible world $\vec{w}$ with which ρ is consistent, if $h(\rho_{-t})$ does not abstain, and $h(\rho_{-t}) \neq \vec{w}_t$.

As for consistency conflicts, we still offer a free pass to a hypothesis that abstains because of reason (R1), and focus, instead, on ensuring that in the remaining cases, the prediction is indeed meaningful. What is meaningful, however, is now more demanding: we expect the hypothesis to agree with information that is not readily available to the learner through the partial observation.

Example 7.9: Consider again the medical domain example as discussed in Example 7.8. As before, h abstains from making a prediction in the first and fourth rows, and, therefore, does not have an accuracy conflict in those cases. An accuracy conflict may occur in row six, exactly in the case where h will predict anything other than 125/80. However, accuracy conflicts may occur also in rows two, three, five, and seven, if the underlying ground truth assigns a value to the blood pressure attribute that differs from the prediction of h.

Although the criterion for determining an accuracy conflict might seem ill-defined — as the learner lacks access to the information that it seeks to predict — this is no more ill-defined than the case of making predictions about the unseen labels of full examples. We could then define a notion of success for learning when dealing with partial examples by insisting that for any distribution D and any masking process M on possible worlds on n attributes, given partial observations drawn independently from

$M(D)$, we return a hypothesis h such that with probability $1 - \delta$,

$$\Pr_{\vec{x} \in D, \rho \in M(\vec{x})}[h(\rho_{-t}) \text{ does not have an accuracy conflict with } \vec{x}] \geq 1 - \epsilon$$

where our algorithm is *sample-efficient* if it satisfies this property while using a number of examples that is polynomial in n, $1/\delta$, and $1/\epsilon$.

To what extent can learning succeed against this more demanding requirement? Recall that our argument that learning against the criterion of consistency conflicts is not too demanding was the connection between reasons (R2) and (R1), with the former making learning harder but implying that the latter would compensate for that hardness. This connection is not of use for accuracy conflicts, as shown in the following result.

Theorem 7.6 (Statistical Indistinguishability in Adversarial Settings) *Consider two hypotheses $h^{(1)}, h^{(2)} : \{0,1\}^{n-1} \to \{0,1\}$ that predict the tth attribute of a possible world, lifted to apply on partial observations. Assume that the hypotheses are neither equivalent, nor the exact negation of one another. Then, for every $\epsilon \in [0,1]$, there exist probability distributions D_1, D_2, and a masking process M, such that:*

1.

$$\Pr_{\vec{x} \in D_1, \rho \in M(\vec{x})}[h^{(1)}(\rho_{-t}) \textit{ does not have an accuracy conflict with } \vec{x}] = 1,$$

and

$$\Pr_{\vec{x} \in D_1, \rho \in M(\vec{x})}[h^{(2)}(\rho_{-t}) \textit{ does not have an accuracy conflict with } \vec{x}] \leq 1 - \epsilon,$$

2.

$$\Pr_{\vec{x} \in D_2, \rho \in M(\vec{x})}[h^{(1)}(\rho_{-t}) \textit{ does not have an accuracy conflict with } \vec{x}] \leq 1 - \epsilon,$$

and

$$\Pr_{\vec{x} \in D_2, \rho \in M(\vec{x})}[h^{(2)}(\rho_{-t}) \textit{ does not have an accuracy conflict with } \vec{x}] = 1,$$

3. $M(D_1) = M(D_2)$, and partial observations only hide x_t with probability ϵ.

Proof Consider the set W of possible worlds $\vec{w}$ such that $h^{(1)}(\vec{w}_{-t}) = w_t = h^{(2)}(\vec{w}_{-t})$, the set W_1 of possible worlds $\vec{w}$ such that $h^{(1)}(\vec{w}_{-t}) = w_t \neq h^{(2)}(\vec{w}_{-t})$, and the set W_2 of possible worlds $\vec{w}$ such that $h^{(1)}(\vec{w}_{-t}) \neq w_t = h^{(2)}(\vec{w}_{-t})$. By assumption on the relation of the hypotheses, all three sets of possible worlds are nonempty. Let D_1 assign probability $1-\epsilon$ to W and probability ϵ to W_1, and let D_2 assign probability $1-\epsilon$ to W and probability ϵ to W_2. Let the masking process M map every possible world $\vec{w} \in W$ to $\rho = \vec{w}$ (i.e., no feature is missing), and every possible world $\vec{w} \in W_1 \cup W_2$ to ρ such that $\rho_{-t} = \vec{w}_{-t}$ and $\rho_t = *$. The claims follow by noting that $h^{(1)}$ and $h^{(2)}$ do not abstain on any $\rho \in M(D_1) \cup M(D_2)$. □

The result demonstrates that it is not the quantity of missing information that prevents meaningful learning against accuracy conflicts, but rather the adversity with

$\rho \in M(\vec{w})$		prediction for x_6	
observation	probability	$h(\rho_{-6})$	accuracy conflict
$0*10*11$	0.27	0	yes
$*1*0*11$	0.15	0	yes
$*11*0*1$	0.33	0	yes
$0*10*1*$	0.04	$*$	no
$0*1*0*1$	0.21	$*$	no

Table 7.2 Partial observations obtained from the possible world $\vec{w} = 0110011$ by applying a particular masking process M, and the corresponding predictions of hypothesis $h = (x_2 \vee \overline{x_4}) \oplus x_7$ that tries to predict attribute x_6.

which the predicted feature is hidden during training. This is to be expected. We cannot distinguish two concepts – and by extension, we cannot learn a hypothesis to approximate a hidden concept – if whenever we would have observed their points of divergence, this information is withheld. We would expect to require that we can observe *all* such points, at least with some probability, and there does not seem to be a way to circumvent this requirement. This is precisely the motivation behind the next definition and results.

Definition 7.7 (Degree of Concealment) A masking process M is $(1-\eta)$-*concealing* with respect to a class of representations C that predict the tth attribute of a possible world if

$$\eta = \min_{\substack{\vec{x}\in\{0,1\}^n \\ c\in C}} \Pr_{\rho\in M(\vec{x})} [\rho_t \neq * \mid c(\rho_{-t}) \text{ has an accuracy conflict with } \vec{x}].$$

(If accuracy conflicts occur with probability 0, we define the conditional probability to equal 1.)

Example 7.10: As an illustration, consider a particular domain where we are trying to predict the 6th attribute of possible worlds that we partially observe, and our hypothesis is $h = (x_2 \vee \overline{x_4}) \oplus x_7$, where $\oplus$ denotes the "exclusive or" binary operator, and one of the possible worlds is $\vec{w} = 0110011$. The masking process M is such that $\vec{w}$ is mapped to partial observations with the probabilities shown in Table 7.2. The partial observations that give rise to accuracy conflicts are drawn with a total probability of 0.75. Among those, the partial observations in which the attribute x_6 is not hidden are drawn with a total probability of 0.42. Thus, by Definition 7.7, $\eta \leq 0.42/0.75 = 0.56$, and the masking process M is at least 0.44-concealing. Additional formulas and examples may impose extra bounds on η, which, in turn, may increase the concealment degree of M.

Note that Theorem 7.6 holds precisely because it appeals to a 1-concealing masking process. In such scenarios, learning against the metric of accuracy conflicts is information-theoretically impossible, irrespective of how much resources we expend. To reinstate learnability, we need to allow learning resources that depend on the degree of concealment, much analogously to how we allow learning resources to depend on the degree of error ϵ that we are willing to tolerate. We can thus redefine our notion of success for learning when dealing with partial examples as follows:

Definition 7.8 *Bounded-concealment learning* of a class of representations C using the class $\mathcal{H}$ is the following task. For any distribution D and any masking process M on possible worlds on n attributes that is $(1-\eta)$-concealing with respect to C such that some $c \in C$ predicts the tth attribute w.r.t. D, and given partial observations drawn independently from $M(D)$, find a hypothesis $h \in \mathcal{H}$ such that with probability $1-\delta$,

$$\Pr_{\vec{x}\in D, \rho\in M(\vec{x})}[h(\rho_{-t}) \text{ does not have an accuracy conflict with } \vec{x}] \geq 1-\epsilon.$$

If the algorithm satisfies this property when given a number of partial observations that is polynomial in n, $1/\delta$, $1/\epsilon$, and $1/\eta$, we say that the algorithm is *sample efficient*, and if the running time is also polynomial in n, $1/\delta$, $1/\epsilon$, and $1/\eta$ the we simply say that the algorithm is *efficient*. If $C = \mathcal{H}$, then we say that the algorithm is *proper*, and otherwise it is *improper*.

Does this added resource make learning against accuracy conflicts less demanding? And how does it compare to learning against consistency conflicts? The following result provides answers to these questions.

Theorem 7.9 (The Relation of Consistency and Accuracy) *For any masking process M on possible worlds on n attributes that is $(1-\eta)$-concealing with respect to a class of representations C that predict the tth attribute of a possible world,*

- *for every probability distribution D and every $h \in C$, if $\eta \neq 0$ then*

  $$\Pr_{\vec{x}\in D, \rho\in M(\vec{x})}[h(\rho_{-t}) \textit{ does not have an accuracy conflict with } \vec{x}] \geq 1-\epsilon$$

  if

  $$\Pr_{\rho\in M(D)}[h(\rho_{-t}) \textit{ does not have a consistency conflict with } \rho] \geq 1-\eta\cdot\epsilon.$$

- *There exists a probability distribution D_0 and $h^{(0)} \in C$ such that*

  $$\Pr_{\vec{x}\in D_0, \rho\in M(\vec{x})}[h^{(0)}(\rho_{-t}) \textit{ does not have an accuracy conflict with } \vec{x}] \geq 1-\epsilon$$

  only if

  $$\Pr_{\rho\in M(D_0)}[h^{(0)}(\rho_{-t}) \textit{ does not have a consistency conflict with } \rho] \geq 1-\eta\cdot\epsilon.$$

Proof Fix a masking process M. For every representation $h \in C$, every example $\vec{x} \in \{0,1\}^n$, every observation $\rho \in \{0,1,*\}^n$ that is consistent with $\vec{x}$, and every probability distribution D, denote by $\mathrm{cc}(h,\rho)$ the event that $h(\rho_{-t})$ has a consistency

conflict with ρ, and by $\texttt{ac}(h, \vec{x}, \rho)$ the event that $h(\rho_{-t})$ has an accuracy conflict with $\vec{x}$. Clearly, the event $\texttt{cc}(h, \rho)$ holds exactly when the events $\texttt{ac}(h, \vec{x}, \rho)$ and $\rho_t \neq *$ hold simultaneously. In particular, this is true even when $\vec{x}$ is drawn from D and ρ is drawn from $M(\vec{x})$. Thus,

$$\Pr_{\vec{x}\in D, \rho\in M(\vec{x})}[\texttt{cc}(h, \rho)] = \Pr_{\vec{x}\in D, \rho\in M(\vec{x})}[\texttt{ac}(h, \vec{x}, \rho) \wedge \rho_t \neq *].$$

By the law of conditional probabilities, the right hand side of the equation equals

$$\Pr_{\vec{x}\in D, \rho\in M(\vec{x})}[\rho_t \neq * \mid \texttt{ac}(h, \vec{x}, \rho)] \cdot \Pr_{\vec{x}\in D, \rho\in M(\vec{x})}[\texttt{ac}(h, \vec{x}, \rho)].$$

We proceed to derive bounds for the first term of the product above. In the first direction, Definition 7.7 implies that for every example $\vec{x} \in \{0,1\}^n$, and every hypothesis $h \in C$, it holds that:

$$\Pr_{\rho\in M(\vec{x})}[\rho_t \neq * \mid \texttt{ac}(h, \vec{x}, \rho)] \geq \eta.$$

Now, if $\vec{x}$ is drawn from any given probability distribution D, the overall probability that $\rho_t \neq *$ given that $\texttt{ac}(h, \vec{x}, \rho)$ remains lower bounded by η. Thus,

$$\Pr_{\vec{x}\in D, \rho\in M(\vec{x})}[\rho_t \neq * \mid \texttt{ac}(h, \vec{x}, \rho)] \geq \eta.$$

In the other direction, Definition 7.7 implies that there exists an example $\vec{x}^{(0)} \in \{0,1\}^n$, and a hypothesis $h^{(0)} \in C$, such that:

$$\Pr_{\rho\in M(\vec{x}^{(0)})}[\rho_t \neq * \mid \texttt{ac}(h^{(0)}, \vec{x^{(0)}}, \rho)] \leq \eta.$$

Now, if $\vec{x}^{(0)}$ is replaced with an example $\vec{x}$ drawn from the probability distribution D_0 that assigns probability 1 to $\vec{x}^{(0)}$ being drawn, the overall probability that $\rho_t \neq *$ given that $\texttt{ac}(h^{(0)}, \vec{x}, \rho)$ remains upper bounded by η. Thus,

$$\Pr_{\vec{x}\in D_0, \rho\in M(\vec{x})}[\rho_t \neq * \mid \texttt{ac}(h^{(0)}, \vec{x}, \rho)] \leq \eta.$$

The claim follows. □

The first half of the result effectively reduces the goal of learning to make highly-accurate predictions to the goal of learning to make highly-consistent predictions, by exploiting the extra resources afforded by their dependence on $1/\eta$. The second half of the result suggests, further, that this reduction is, in the worst case, an optimal strategy for recovering missing information. And this is the case without requiring knowledge of η, which is, in general, not possible to determine. Consequently, we can restrict our investigation of learnability from partial observations using only the notion of consistency as our learning criterion.

7.3 Examples of learnable functions.

Let us now proceed to establish the existence of certain learnable classes of functions using the notion of consistency as our learning criterion, which allows, in turn, the consideration of partial observations coming from an arbitrary masking process. An immediate consequence of establishing learnability for an arbitrary masking process is that the representations in question are learnable from full examples. The question is what additional requirements need to be satisfied to obtain a sufficient condition. We will employ an analogous version of the consistent representation problem and Occam's Razor from Section 4.1. We leave it as an exercise to the reader to establish these analogous versions for the case of partial observations (Exercise 7.5).

Let us start by considering Algorithm 12, which turns partial observations into full examples by imputing missing values with the value of the attribute to be predicted, if this is available, or otherwise maps the partial observation to the all-zeros example.

Input: Partial observations $\rho^{(1)}, \ldots, \rho^{(m)}$ consistent with examples $\vec{x}^{(1)}, \ldots, \vec{x}^{(m)}$ such that $c(\vec{x}^{(j)}_{-t}) = \vec{x}^{(j)}_t$ for some $c \in C \setminus \{\top\}$.
Output: A hypothesis h solving $\text{ConsisPartial}^{mn+p(n)}_{C_n}$.
begin
  `// Complete missing information in attributes`
  for $j = 1, \ldots, m$ **do**
    for $i = 1, \ldots, n$ **do**
      if $\rho^{(j)}_t = *$ **then** $\tilde{\rho}^{(j)}_i = 0$
      if $\rho^{(j)}_t \in \{0,1\}$ *and* $\rho^{(j)}_i = *$ **then** $\tilde{\rho}^{(j)}_i = \rho^{(j)}_t$
      if $\rho^{(j)}_t \in \{0,1\}$ *and* $\rho^{(j)}_i \in \{0,1\}$ **then** $\tilde{\rho}^{(j)}_i = \rho^{(j)}_i$
    end
  end
  `// Learn from complete examples`
  Execute algorithm L to obtain h solving $\text{Consis}^{p(n)}_{C_n}(\tilde{\rho}^{(1)}, \ldots, \tilde{\rho}^{(m)})$
  return h.
end

Algorithm 12: Completion Algorithm for learning from partial observations

We first establish that Algorithm 12 succeeds at learning *monotone* properties, i.e.:

Definition 7.10 We say that a property $c \subseteq \{0,1\}^n$ is *monotone* if for any $\vec{x}$ and $\vec{y}$ such that for all attributes $x_i \leq y_i$, $c(\vec{x}) \leq c(\vec{y})$ (interpreting the truth values $\{0,1\}$ as integers).

We remark briefly that monotone properties are precisely those that can be expressed by De Morgan formulas without using negation; indeed, analogously to Theorem 3.14, we find that the class of *monotone DNF* formulas, given by DNFs that do not use negation, express all such properties. Observe that linear threshold functions (Definition 4.24) that only use nonnegative weights are also monotone.

Theorem 7.11 *For any class C of monotone properties that are learnable from full examples, Algorithm 12 learns C from partial observations.*

Proof We first need to establish that the full examples $\tilde{\rho}^{(1)}, \ldots, \tilde{\rho}^{(m)}$ are still expressing a property from the class C, and thus calling algorithm L is meaningful. This follows directly by observing that for any possible world $\vec{x}$ consistent with $\rho^{(j)}$, $\tilde{\rho}_i^{(j)} \geq x_i$ for all attributes if $\rho_t^{(j)} = 1$, and $\tilde{\rho}_i^{(j)} \leq x_i$ in case $\rho_t^{(j)} = 0$. Hence, for the $\vec{x}^{(j)}$ such that $\rho_t^{(j)} = c(\vec{x}^{(j)})$ for $c \in C \setminus \{\top\}$, by the monotonicity of c, $c(\vec{x}^{(j)}) = c(\tilde{\rho}^{(j)})$. Also, if $\rho_t^{(j)} = *$, then $\rho^{\tilde{(j)}}$ comprises of all 0s. Again by the monotonicity of c and since $c \neq \top$, we have $c(\tilde{\rho}_{-t}^{(j)}) = \tilde{\rho}_t^{(j)}$.

Now, consider the returned hypothesis $h \in C$ that is consistent with the full examples $\tilde{\rho}^{(1)}, \ldots, \tilde{\rho}^{(m)}$, as returned by algorithm L. Since the partial observations $\rho^{(1)}, \ldots, \rho^{(m)}$ are, respectively, consistent with the full examples $\tilde{\rho}^{(1)}, \ldots, \tilde{\rho}^{(m)}$, it follows that for every j, either $h(\rho_{-t}^{(j)}) = *$, or $h(\rho_{-t}^{(j)}) = h(\tilde{\rho}_{-t}^{(j)})$. In either case, h is consistent with the partial observations $\rho^{(1)}, \ldots, \rho^{(m)}$. □

Noting that the exclusion of the tautology from the class is without loss of generality (as it can be easily checked through sampling whether the property to be learned is indeed the tautology), Theorem 7.11 directly implies that the classes of monotone conjunctions, monotone disjunctions, monotone k-CNF, monotone k-DNF, and monotone linear thresholds, among others, known to be learnable in the case of full examples, are also learnable from partial observations, using the notion of consistency as our learning criterion.

We can now attempt to generalize the family of learnable classes by means of a learning reduction between a class to be learned from partial observations, and a class of monotone functions known to be learnable from partial observations. The proof of the result below follows the key ideas of the proof of Theorem 4.14, adapted to deal with the extra nuances brought about by the nature of partial observations. For concreteness, and to emphasize the adaptations that need to be made, our result and proof will be specifically for learning the concept class of halfspaces (non-monotone linear threshold functions, introduced in Definition 4.24). Note that halfspaces (and monotone linear threshold functions) are properly learnable in a full-information setting; see Appendix A.4 for details.

Theorem 7.12 *There is an efficient algorithm that learns halfspaces from partial observations.*

Proof We suppose we have a set of labeled possible worlds with n Boolean attributes $X_1, \ldots, X_n$, and let X_t be the attribute that expresses a concept as a function of the remaining attributes. We will construct a new set of attributes as follows: we create an attribute Y_t, and for each literal ℓ, we create an attribute Y_ℓ; we will map each input partial observation $\rho^{(j)}$ to a new partial observation $\tilde{\rho}^{(j)}$ on these new attributes where $\tilde{\rho}_\ell^{(j)} = \ell(\rho^{(j)})$ (i.e., if $\ell = X_i$, $\tilde{\rho}^{(j)} = \rho_i^{(j)}$ and if $\ell = \neg X_i$, $\tilde{\rho}^{(j)} = *$ if $\rho_i^{(j)} = *$, and $\tilde{\rho}^{(j)} = 1 - \rho_i^{(j)}$ otherwise) and $\tilde{\rho}_t^{(j)} = \rho_t^{(j)}$.

We define a *read-once* monotone linear threshold function to be a monotone linear threshold function in which only one literal in each complementary pair appears (either Y_{X_i} or $Y_{\neg X_i}$ has a zero coefficient). We can find a consistent read-once monotone linear threshold function by solving the following linear program in the variables B, $U_1, \ldots, U_n, V_1, \ldots, V_n$, and $W_1, \ldots, W_n$:

$$\text{minimize} \sum_{i=1}^{n} U_i + V_i \text{ s.t.}$$

$$\sum_{i=1}^{n} U_i y_{X_i}^{(j)} + V_i y_{\neg X_i}^{(j)} + V_i \geq B + 1 \qquad \text{for } j : y_t^{(j)} = 1$$

$$\sum_{i=1}^{n} U_i y_{X_i}^{(j)} + V_i y_{\neg X_i}^{(j)} + V_i \leq B - 1 \qquad \text{for } j : y_t^{(j)} = 0$$

$$U_i \geq 0 \qquad \text{for } i = 1, \ldots, n$$

$$U_i \geq W_i \qquad \text{for } i = 1, \ldots, n$$

$$V_i \geq 0 \qquad \text{for } i = 1, \ldots, n$$

$$V_i \geq -W_i \qquad \text{for } i = 1, \ldots, n$$

(since we can rescale the parameters to obtain a margin of 1, see Appendix A.4 for details) where, recall, there is a polynomial-time algorithm for linear programming. This obtains a read-once monotone linear threshold function $[\sum_{i=1}^{n} u_i Y_{X_i} + v_i Y_{\neg X_i} \geq b - \sum_{i=1}^{n} v_i]$ that agrees with $y_t^{(j)}$ for all respective assignments $\vec{Y} = \vec{y}^{(j)}$ because in any returned solution, $u_i = \max\{w_i, 0\}$ and $v_i = \max\{-w_i, 0\}$ for all i, where at most one of these is nonzero, and again following Appendix A.4, the program is feasible if such a linear threshold function exists. We note that for the linear function c given by $[\sum_i w_i^* X_i \geq b^*]$ there is a read-once monotone linear threshold function that takes the same values on all (complete) possible worlds under our embedding. Indeed, consider $\tilde{c}$ given by

$$\begin{aligned}
\tilde{c}(\vec{Y}_{-t}) &= \left[\sum_{i : w_i^* \geq 0} w_i^* Y_{X_i} + \sum_{i : w_i^* < 0} (-w_i^*) Y_{\neg X_i} \geq b^* + \sum_{i : w_i^* < 0} -w_i^* \right], i.e., \\
&= \left[\sum_{i : w_i^* \geq 0} w_i^* Y_{X_i} + \sum_{i : w_i^* < 0} w_i^* (1 - Y_{\neg X_i}) \geq b^* \right] \\
&= \left[\sum_{i=1}^{n} w_i^* X_i \geq b^* \right] \text{ for } Y_{X_i} = X_i,\ y_{\neg X_i} = 1 - X_i
\end{aligned}$$

We run Algorithm 12 to obtain a read-once monotone linear threshold function $\tilde{h}$ over the attributes $\vec{Y}_{-t}$. We then map $\tilde{h}$ back to a linear threshold function h on the original attributes: for the obtained linear threshold function $[\sum_\ell \tilde{w}_\ell Y_\ell \geq \tilde{b}]$ we put the threshold b equal to $\tilde{b} - \sum_{i=1}^{n} \tilde{w}_{\neg X_i}$ and $w_i = \tilde{w}_{X_i} - \tilde{w}_{\neg X_i}$. Observe that since $\neg X_i = 1 - X_i$ for $X_i \in \{0, 1\}$, for all complete $\vec{x} \in \{0, 1\}^n$, $\tilde{h}(\vec{y}) = h(\vec{x})$ (we have just

collected all of the coefficients); equivalently, observe that these are the values b and $w_1, \ldots, w_n$ obtained in our solution to the linear program above.

For correctness, we first observe that for any full example $\vec{x}^{(j)}$ with which $\rho^{(j)}$ is consistent, it holds that $\ell(\rho^{(j)}) \in \{\ell(\vec{x}^{(j)}), *\}$, which implies that $\tilde{\rho}_\ell^{(j)} \in \{y_\ell^{(j)}, *\}$, and thus that $\tilde{\rho}^{(j)}$ is consistent with $\vec{y}^{(j)}$. Thus, the new partial observations can be viewed as being drawn from a masking process over examples $\vec{y}$. Moreover, we obtain a distribution on complete $\vec{y}$ such that $\vec{y}_t$ is equal to the value of the read-once monotone linear threshold function $\tilde{c}$ over $\vec{y}_{-t}$. This makes it meaningful to run Algorithm 12 to obtain a read-once monotone linear threshold function $\tilde{h}$.

To complete the proof we need to show that if h has a consistency conflict with a partial observation $\rho^{(j)}$, then $\tilde{h}$ has a consistency conflict with the partial observation $\tilde{\rho}^{(j)}$. Indeed, if h has a consistency conflict with $\rho^{(j)}$, then $h(\rho_{-t}^{(j)}) \in \{0,1\}$, $\rho_t^{(j)} \in \{0,1\}$, and $h(\rho_{-t}^{(j)}) \neq \rho_t^{(j)}$. If we can establish that $\tilde{h}(\tilde{\rho}_{-t}^{(j)}) = h(\rho_{-t}^{(j)})$, and since $\tilde{\rho}_t^{(j)} = \rho_t^{(j)}$, it follows that $\tilde{h}(\tilde{\rho}_{-t}^{(j)}) \in \{0,1\}$, $\tilde{\rho}_t^{(j)} \in \{0,1\}$, and $\tilde{h}(\tilde{\rho}_{-t}^{(j)}) \neq \tilde{\rho}_t^{(j)}$, as needed. It remains to show, then, that $\tilde{h}(\tilde{\rho}_{-t}^{(j)}) = h(\rho_{-t}^{(j)})$.

We next establish the following auxiliary claim: for every example $\vec{y}$ with which $\tilde{\rho}$ is consistent, there exists an example $\vec{x}$ with which ρ is consistent, and such that $\tilde{h}(\vec{y}_{-t}) = h(\vec{x}_{-t})$. To construct $\vec{x}$, fix in $\vec{x}$ any attributes that have a $\{0,1\}$ value in ρ to have those same values. Then, consider any literal ℓ with a positive coefficient in $\tilde{h}$ such that $\ell(\rho_{-t}) = *$. It follows that ℓ is a literal on a masked attribute in ρ, and that there is a way to fix its value in $\vec{x}$ so that ℓ evaluates to y_ℓ. Any attributes x_i that have only zero coefficients in $\tilde{h}$ may be set arbitrarily, without changing the value of h or $\tilde{h}$. Since we have ensured that at most one of X_i or $\neg X_i$ appeared in $\tilde{h}$, $\vec{x}$ is well-defined, and $\tilde{h}(\vec{y}_{-t}) = h(\vec{x}_{-t})$.

Observe, then, that if $h(\rho_{-t}^{(j)}) \in \{0,1\}$, then $h(\rho_{-t}^{(j)}) = h(\vec{x}_{-t})$ for every example $\vec{x}$ with which $\rho^{(j)}$ is consistent; i.e., the value of h across all such examples $\vec{x}$ is common. From the auxiliary claim above it follows that the value of $\tilde{h}$ across every example $\vec{y}$ with which $\tilde{\rho}$ is consistent will equal this common value $h(\rho_{-t}^{(j)})$. But this common value is, by definition, $\tilde{h}(\tilde{\rho}_{-t}^{(j)})$, as needed. □

7.4 Non-learnability under bounded concealment.

Since our formulation of learning from partial observations insists that learning succeeds across all masking processes, it implies the learnability from full examples as well. Are there, however, cases of concept classes that are learnable from full examples but not from partial observations across all masking processes? Put differently, does the ability to abstain compensate fully for the extra learning demands that coping with missing information brings? Below we show that partial observability does impose some restrictions on learning, be it with respect to the learning criterion of consistency or accuracy (since learnability with the latter implies learnability with the former).

We note upfront that this gap between learnability from full examples and learnability from partial observations does not rely on a high-concealing masking process,

nor on the excess hiding of attributes in each partial observation. Indeed, we will employ masking processes that never hide the attribute to be predicted, and hence are 0-concealing, and only hide up to three attributes in each observation. Our negative results will be with regards to proper learnability, and will involve concept classes that are properly learnable from full examples: parities and monotone-term 1-decision lists.

We will first establish a general result, and then derive our specific negative results as corollaries. As usual, negative learnability results rely on some hardness assumptions. In our case, this hardness assumption is that some NP search problems cannot be solved by randomized polynomial time algorithms.

Theorem 7.13 (Sufficient Condition for Hard Learning Tasks) *Consider* 3*-CNF formulas of size n, and a class* C *of representations on possible worlds over* $n' = \text{poly}(n)$ *attributes, such that* $\log|C|$ *is of size* $\text{poly}(n)$*. Assume that there exists an algorithm that on input a* 3*-CNF formula* χ *of size n runs in time* $\text{poly}(n)$ *and outputs a set of observations* $O(\chi) \subseteq \{0,1,*\}^{n'}$ *such that:*

- *every representation in* C *is evaluatable in time* $\text{poly}(n)$ *on every observation in* $O(\chi)$*;*
- χ *is satisfiable if there exists a representation in* C *that does not have a consistency conflict with any observation in* $O(\chi)$*;*
- χ *is satisfiable only if there exists a representation* $c \in C$*, a probability distribution* D *over all possible worlds in* $\{0,1\}^{n'}$ *for which* $\vec{x}_t = c(\vec{x}_{-t})$*, and a masking process* M *such that* $M(D)$ *is the uniform distribution over* $O(\chi)$*.*

Then, if the class C *is efficiently properly consistently learnable, every NP search problem has a randomized polynomial time algorithm*

Proof Suppose that the class C is efficiently properly consistently learnable by the algorithm L with sample complexity given by the polynomial $m(\cdot,\cdot,\cdot)$ and running in time bounded by the polynomial $q(\cdot,\cdot,\cdot)$. Construct the algorithm L_{sat} as follows:

On input a 3-CNF formula χ of size n, algorithm L_{sat} constructs the set of observations $O(\chi)$. For $\delta = 1/3$ and $\epsilon = 1/2|O(\chi)|$, it produces a sample of $m(n', 1/\epsilon, 1/\delta)$ partial observations by sampling uniformly at random from $O(\chi)$. It then proceeds to simulate algorithm L on these examples with input $\delta = 1/3$ and $\epsilon = 1/2|O(\chi)|$. The simulation is interrupted after $q(n', 1/\delta, 1/\epsilon)$ time-steps. If the simulated algorithm L returns a representation h, algorithm L_{sat} checks and returns whether or not h has a consistency conflict with any observation in $O(\chi)$. If the simulation of algorithm L is interrupted, algorithm L_{sat} returns `false`. In either case, algorithm L_{sat} terminates after returning a truth-value.

We will prove that algorithm L_{sat} runs in time $\text{poly}(n)$ and determines whether a given arbitrary 3-CNF formula χ of size n is satisfiable so that: if χ is unsatisfiable, then algorithm L_{sat} will return `false` with probability 1; if χ is satisfiable, then algorithm L_{sat} will return `true` with probability at least $2/3$.

Assume first that χ is unsatisfiable. By the second condition, every representation in C has a consistency conflict with some observation in $O(\chi)$. Therefore, algorithm

L_{sat} will return `false`, as expected, irrespective of whether algorithm L returns some hypothesis $h \in C$, or its simulation is interrupted.

Assume now that χ is satisfiable. By the third condition, there exist c, D, and M such that we provide the algorithm with $m(n', 1/\epsilon, 1/\delta)$ observations from $M(D)$. By definition of consistent learnability, algorithm L will run in time $q(n', 1/\delta, 1/\epsilon)$, and return, with probability $1 - \delta$, a hypothesis $h \in C$ such that

$$\Pr_{\rho \in M(D)}[h(\rho_{-t}) \text{ does not have a consistency conflict with } \rho] \geq 1 - \epsilon.$$

The algorithm L_{sat} will therefore obtain h. Since ϵ is strictly less than the probability with which any particular observation from $O(\chi)$ is drawn, it follows that, with probability $1 - \delta$, the returned hypothesis $h \in C$ will have no consistency conflict with any observation in $O(\chi)$, and algorithm L_{sat} will verify this and return `true`, as expected. Since $\delta = 1/3$, the probability with which algorithm L_{sat} will correctly report that χ is satisfiable is at least $2/3$.

Since $1/\delta = 3$, $1/\epsilon = 2|O(\chi)| = \text{poly}(n)$, $n' = \text{poly}(n)$, and $\log |C| = \text{poly}(n)$, it follows that $q(n', 1/\delta, 1/\epsilon)$ is $\text{poly}(n)$. The set of observations $O(\chi)$ is constructible in time $\text{poly}(n)$, observations are uniformly samplable from $O(\chi)$ in time $\text{poly}(n)$, and, by the first condition, returned hypotheses are testable for consistency conflicts with observations in $O(\chi)$ in time $\text{poly}(n)$. Hence, algorithm L_{sat} runs in time $\text{poly}(n)$.

Therefore, Algorithm L_{sat} solves an NP-complete search problem using a randomized polynomial time algorithm, and hence every NP search problem may be solved by such algorithms using an appropriate reduction. □

To establish our specific negative learnability results, then, it remains to show how specific concept classes can be used to construct efficiently the observations $O(\chi)$ assumed by Theorem 7.13.

A *parity formula* over a set of attributes is a formula of the form $x_{i_1} \oplus \cdots \oplus x_{i_r}$, where $\oplus$ denotes the "exclusive or" binary operator. (Subsequently, we will use the notation $\bigoplus_{i \in S} x_i$ where $S = \{1_i, \ldots, i_r\}$.) A parity formula evaluates to 1 on an example $\vec{x}$ exactly when an odd number of the formula's attributes are assigned the value 1 in $\vec{x}$. Parity formulas are known to be properly PAC-learnable, albeit using techniques that rely critically on the availability of an explicit set of complete observations. When the labels are corrupted by noise, learning parities is conjectured to be intractable. (It is a close relative of the Learning With Errors problem from Section 4.8.) Interestingly, parity formulas are highly non-monotone, which aligns well with the result on the consistent learnability of concept classes of monotone formulas.

Theorem 7.14 (Intractability of Proper Consistent Learnability of Parities) *If parity formulas are efficiently properly consistently learnable, then every NP search problem has a randomized polynomial time algorithm.*

Proof Let $V = \{v_1, v_2, \ldots, v_n\}$ be the set of variables over which 3-CNF formulas are defined. Construct possible worlds over the $n' = 2n + 1$ attributes $\{x_i^+, x_i^- \mid v_i \in V\} \cup \{x_t\}$, and let C be the set of all parity formulas over the first $2n$ attributes.

For each variable $v_i \in V$, we define $a(v_i) \triangleq x_i^+$ and $a(\neg v_i) \triangleq x_i^-$; by construction, the mapping from the set of literals over V to the set of attributes $\{x_i^+, x_i^- \mid v_i \in V\}$ is bijective. For every 3-CNF formula

$$\chi = \bigwedge_{j=1}^{n'} (l_{j,1} \vee l_{j,2} \vee l_{j,3}),$$

where each $l_{j,k}$ is a literal over V, denote by $O(\chi)$ the set of observations that contains exactly the following:

- for every $i : 1 \leq i \leq n$, the observation $\rho^{var(i)}$ that assigns the value 1 to attributes x_i^+, x_i^-, the value 1 to attribute x_t, and the value 0 to the remaining attributes;
- for every $j : 1 \leq j \leq n'$, the observation $\rho^{cls(j)}$ that assigns the value $*$ to attributes $a(l_{j,1}), a(l_{j,2}), a(l_{j,3})$, the value 1 to attribute x_t, and the value 0 to the remaining attributes.

We continue to establish that formula χ is satisfiable if (and only if) there exists a parity formula $h \in C$ that does not have a consistency conflict with any observation in $O(\chi)$. Consider any such parity formula h, and let τ_h be the set of literals l over V such that $a(l)$ are included in h. There is a correspondence between truth-assignments for χ induced by τ_h, and a lack of consistency conflicts of h with the observations in $O(\chi)$ that are of the first type above. That is, we say that τ_h *induces a truth-assignment for* χ iff for every $i : 1 \leq i \leq n$, exactly one of v_i and $\neg v_i$ belongs to τ_h. Now, this is equivalent to

$$\begin{aligned} &\forall i : 1 \leq i \leq n, \text{ exactly one of } x_i^+, x_i^- \text{ appears in } h \\ \Leftrightarrow &\forall i : 1 \leq i \leq n, h \text{ does not have a consistency conflict with } \rho^{var(i)}. \end{aligned}$$

A second derivation establishes a correspondence between sets of literals determined by τ_h that would satisfy χ, and the lack of consistency conflicts of h with the observations in $O(\chi)$ that are of the second type:

$$\begin{aligned} &\tau_h \text{ contains at least one literal from each clause of } \chi \\ \Leftrightarrow &\forall j : 1 \leq j \leq n', \text{ at least one of } l_{j,1}, l_{j,2}, l_{j,3} \text{ belongs to } \tau_h \\ \Leftrightarrow &\forall j : 1 \leq j \leq n', \text{ at least one of } a(l_{j,1}), a(l_{j,2}), a(l_{j,3}) \text{ belongs to } h \\ \Leftrightarrow &\forall j : 1 \leq j \leq n', h(\rho_{-t}^{cls(j)}) = * \\ \Leftrightarrow &\forall j : 1 \leq j \leq n', h \text{ does not have a consistency conflict with } \rho^{cls(j)}. \end{aligned}$$

Together, the two derivations imply that τ_h induces a satisfying truth-assignment for χ if (and only if) h does not have a consistency conflict with any observation in $O(\chi)$. This conclusion, along with the bijection property of the mapping, leads to the

following derivation that establishes the claim:

χ is satisfiable
$\Leftrightarrow \exists$ a set of literals over V, τ_h that induces a satisfying truth-assignment for χ
$\Leftrightarrow \exists \tau_h : h$ does not have a consistency conflict with any observation in $O(\chi)$
$\Leftrightarrow \exists h \in C$ that does not have a consistency conflict with any observation in $O(\chi)$.

To conclude the proof it suffices to show that the conditions of Theorem 7.13 are satisfied. Clearly, n' is polynomial in n, and so is $\log |C|$, since there are at most $2^{n'}$ formulas in C. Also, the set of observations $O(\chi)$ is constructible in time polynomial in n. For the first condition of Theorem 7.13, note that each parity formula $h \in C$ can be evaluated on each observation $\rho \in O(\chi)$ in time polynomial in n. Indeed, either an attribute in h is masked in ρ, in which case $h(\rho_{-t}) = *$, or none of the attributes in h is masked in ρ, in which case the number of attributes that are assigned the value 1 in ρ_{-t} determines $h(\rho_{-t})$.

The second condition of Theorem 7.13 follows directly from the last derivation above. For the third condition of Theorem 7.13, assume that χ is satisfiable, and let τ be any set of literals over V that induces a satisfying truth-assignment for χ. Let ϕ_τ be the parity formula $\bigoplus_{l \in \tau} a(l)$. Consider the set of examples that contains exactly the following:

- for every $i : 1 \leq i \leq n$, the example $\vec{x}^{var(i)}$ that assigns the value 1 to attributes x_i^+, x_i^-, the value 1 to attribute x_t, and the value 0 to the remaining attributes;
- for every $j : 1 \leq j \leq n'$, the example $\vec{x}^{cls(j)}$ that assigns the value 1 to the least (under some fixed ordering of the n' attributes) attribute $a(l_{j,k})$ in the set $\{a(l_{j,1}), a(l_{j,2}), a(l_{j,3})\} \cap \{a(l) \mid l \in \tau\}$, the value 1 to attribute x_t, and the value 0 to the remaining attributes.

Define the probability distribution D that returns each of these examples with equal probability. Define the masking process M that maps each example to the observation in $O(\chi)$ with the same superscript with probability 1. Clearly, $M(D)$ is uniform over $O(\chi)$, and $\vec{x}_t = \phi_\tau(\vec{x}_{-t})$ for the parity formula $\phi_\tau \in C$ for all constructed examples. The third condition of Theorem 7.13 follows, and the proof is complete. □

We can also establish the intractability of proper learning of decision lists from partial examples. Recall that we introduced decision lists in Section 4.4, where we showed that these are properly learnable with complete examples. We remark that the algorithm can be easily adapted to properly learn *monotone-term* decision lists, in which it is always the positive attribute value that leads to a decision.

Theorem 7.15 (Intractability of Proper Consistent Learnability of Monotone-Term Decision Lists) *If monotone-term decision list are efficiently properly consistently learnable then every NP search problem has a randomized polynomial-time algorithm.*

Proof Let $V = \{v_1, v_2, \ldots, v_n\}$ be the set of variables over which 3-CNF formulas are defined. Construct possible worlds over the $n' = 2n + 1$ attributes $\{x_i^+, x_i^- \mid v_i \in$

$V\} \cup \{x_t\}$, and let C be the set of all monotone-term decision lists over the first $2n$ attributes.

For each variable $v_i \in V$, we define $a(v_i) \triangleq x_i^+$ and $a(\neg v_i) \triangleq x_i^-$; by construction, the mapping from the set of literals over V to the set of attributes $\{x_i^+, x_i^- \mid v_i \in V\}$ is bijective. For every 3-CNF formula

$$\chi = \bigwedge_{j=1}^{m} (l_{j,1} \vee l_{j,2} \vee l_{j,3}),$$

where each $l_{j,k}$ is a literal over V, denote by $O(\chi)$ the set of observations that contains exactly the following:

- the observation ρ^{zero} that assigns the value 0 to all attributes;
- for every $i : 1 \leq i \leq n$, the observation $\rho^{var(i,1)}$ that assigns the value $*$ to attributes x_i^+, x_i^-, the value 1 to attribute x_t, and the value 0 to the remaining attributes;
- for every $i : 1 \leq i \leq n$, the observation $\rho^{var(i,0)}$ that assigns the value 1 to attributes x_i^+, x_i^-, the value 0 to attribute x_t, and the value 0 to the remaining attributes;
- for every $j : 1 \leq j \leq n'$, the observation $\rho^{cls(j)}$ that assigns the value $*$ to attributes $a(l_{j,1}), a(l_{j,2}), a(l_{j,3})$, the value 1 to attribute x_t, and the value 0 to the remaining attributes.

Without loss of generality, for the remainder of this proof we consider only *read-once* monotone-term decision lists, where no attribute appears in conditions more than once; for any monotone-term decision list that violates this assumption, one may simply drop all but the first condition out of those that contain any particular attribute, and obtain a new monotone-term decision list that respects the assumption and is equivalent to the first one.

We continue to establish that formula χ is satisfiable if there exists a monotone-term decision list $h \in C$ that does not have a consistency conflict with any observation in $O(\chi)$. Consider any such monotone-term decision list $h \in C$, and let τ_h be the set of literals l over V such that h includes a check for $a(l)$ with label 1. We continue to verify that τ_h induces a satisfying assignment for χ by identifying the implications from the assumption on h for each of the four types of observations in $O(\chi)$:

- From observation ρ^{zero}, it follows that the tail of h has label 0.
- For every $i : 1 \leq i \leq n$, from observation $\rho^{var(i,1)}$, we find that h either checks x_i^+ with label 1 or checks x_i^- with label 1, for otherwise these examples are either labeled 0 by any checks for these attributes or reach the tail of h, and therefore by the conclusion above, h would evaluate to 0 on $\rho^{var(i,1)}$.
- For every $i : 1 \leq i \leq n$, from observation $\rho^{var(i,0)}$, it follows that h cannot include checks for *both* of the attributes x_i^+, x_i^- with label 1, for otherwise h would evaluate to 1 on $\rho^{var(i,0)}$. Hence, by the above it contains exactly one of these with label 1, and furthermore must check the other with label 0 before that check with label 1.

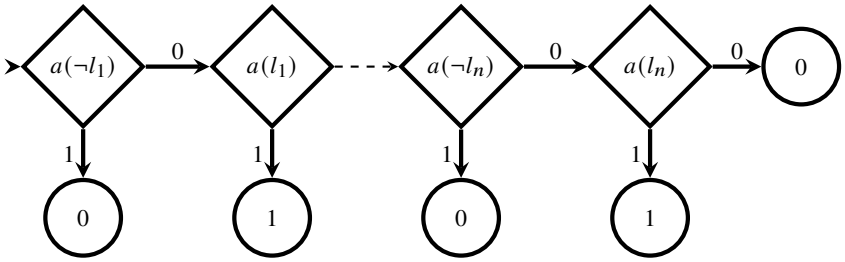


Figure 7.1 The decision list ϕ_τ on V corresponding to the truth assignment that satisfies the literals $l_1, \ldots, l_n$, satisfying χ.

- For every $j : 1 \leq j \leq n'$, from observation $\rho^{cls(j)}$, it follows that h contains a check for at least one of the attributes $a(l_{j,1})$, $a(l_{j,2})$, or $a(l_{j,3})$ with label 1, for otherwise these examples would either be labeled 0 by any checks on these attributes or reach the tail, and by the first conclusion above, h would evaluate to 0 on $\rho^{cls(j)}$.

From the second and third conclusion, τ_h induces a truth-assignment for χ, and from the fourth conclusion, the induced truth-assignment is a satisfying one for χ; thus, χ is satisfiable, as needed.

To conclude the proof it suffices to show that the conditions of Theorem 7.13 are satisfied. Clearly, n' is polynomial in n, and so is $\log |C|$, since there are at most $3^{n'} n'!$ formulas in C. Also, the set of observations $O(\chi)$ is constructible in time polynomial in n. For the first condition of Theorem 7.13, note that each monotone-term decision list $h \in C$ can be evaluated on each observation $\rho \in O(\chi)$ in time polynomial in n. Indeed, for each attribute in h that is assigned the value 1 in ρ one may prune the suffix of h that follows the check in h associated with the attribute and use its label as the tail, without affecting the value of h on ρ. Similarly, for each attribute in h that is assigned the value 0 in ρ one may drop the check of h associated with the attribute, without affecting the value of h on ρ. After the monotone-term decision list h has been thus processed to obtain h', all remaining attributes in h' will be masked in ρ. By construction, $h(\rho_{-t}) = h'(\rho_{-t})$, and clearly $h'(\rho_{-t})$ is 1 if all labels in h' are 1, it is 0 if all labels in h' are 0, and it is $*$ otherwise.

The second condition of Theorem 7.13 follows directly from the derivation above. For the third condition of Theorem 7.13, assume that χ is satisfiable, and let τ be any set of literals over V that induces a satisfying truth-assignment for χ. Let ϕ_τ be the corresponding decision list shown in Figure 7.1, that for each $l \in \tau$ first checks $a(\neg l)$ with label 0 and then checks $a(l)$ with label 1, and finally labels the tail 0. By construction, for every $v_i \in V$, the checks x_i^+, x_i^- appear in ϕ_τ, with the attribute appearing second with a corresponding value 1 being the one associated with the truth-value of v_i as determined by τ. Consider the set of examples that contains exactly the following:

- the example $\vec{x}^{zero}$ that assigns the value 0 to all attributes;

- for every $i : 1 \leq i \leq n$, the example $\vec{x}^{var(i,1)}$ that assigns the value 1 to the single attribute $x_i^{\pm}$ in the set $\{x_i^+, x_i^-\} \cap \{a(l) \mid l \in \tau\}$, the value 1 to attribute x_t, and the value 0 to the remaining attributes;
- for every $i : 1 \leq i \leq n$, the example $\vec{x}^{var(i,0)}$ that assigns the value 1 to attributes x_i^+, x_i^-, the value 0 to attribute x_t, and the value 0 to the remaining attributes;
- for every $j : 1 \leq j \leq m$, the example $\vec{x}^{cls(j)}$ that assigns the value 1 to all attributes in the set $\{a(l_{j,1}), a(l_{j,2}), a(l_{j,3})\} \cap \{a(l) \mid l \in \tau\}$, the value 1 to attribute x_t, and the value 0 to the remaining attributes.

Define the probability distribution D that returns each of these examples with equal probability. Define the masking process M that maps each example to the observation in $O(\chi)$ with the same superscript with probability 1. Clearly, $M(D)$ is uniform over $O(\chi)$, and $\vec{x}_t = \phi_\tau(\vec{x}_{-t})$ for the monotone-term decision list $\phi_\tau \in C$ for all of the constructed examples. The third condition of Theorem 7.13 follows, and the proof is complete. □

The intractability result of Theorem 7.15 provides yet another indication that learnability from partial observations is harder than learnability from full examples. We remark that since decision lists may be expressed as linear threshold functions (see Exercise 7.6) which are learnable by Theorem 7.12, these are *improperly* learnable from partial examples. It remains open, however, whether parities can be learned improperly.

7.5 Witnessed evaluation and learning complex functions.

We thus see that the kinds of representations that can be learned under the minimal assumption of bounded concealment is unfortunately rather limited in comparison to those that can be learned with complete information. This naturally raises the question or whether there some still-reasonable stronger assumptions that restore our ability to learn, for example, k-CNF representations.

In the analysis of our algorithm for learning monotone representations, we made essential use of the fact that the restriction of a monotone function is only unsatisfiable if a sufficient set of attributes is set to false. Obviously, for a k-CNF, this is not true: we can have an unsatisfiable k-CNF in which no variables are given values, e.g., we know the set of all clauses of size k is unsatisfiable. If only we could guarantee that enough attributes were set to determine that some particular clause of the CNF were actually false, we could carry the argument through. We actually encountered a suitable approach previously, partial evaluation (Definition 5.10), which leads naturally to the following definition:

Definition 7.16 We say that a De Morgan formula φ is *witnessed true* on a partial assignment ρ if $\varphi|_\rho = \top$. Similarly, we say that φ is *witnessed false* on ρ if $\varphi|_\rho = \bot$. Finally, we say that φ's *value is witnessed* on ρ if φ is either witnessed true or witnessed false on ρ.

Observe that since $\top$ and $\bot$ evaluate to 1 and 0, respectively, in all possible worlds, if a formula φ is witnessed on a partial example $\rho = m(\vec{x})$ for a masking function m on some possible world $\vec{x}$, it must be that φ indeed has the respective truth value on $\vec{x}$: ρ is consistent with $\vec{x}$, and so when we fix the remaining attributes to take the same values as in $\vec{x}$, we obtain $\varphi|_{\vec{X}=\vec{x}} = \top|_{X_i=x_i : i \text{ not set in } \rho} = 1$ or $\bot|_{X_i=x_i : i \text{ not set in } \rho} = 0$. Note, moreover, that a CNF is witnessed false if and only if one of its clauses has all of its literals observed false.

Example 7.11: Recall the clausal encoding of a STRIPS planning domain, described in Section 5.6: each action has a set of precondition and effect literals, such that if the action is taken, the precondition literals must have been satisfied in the "pre-state" (at the current time step t) and the effect literals must be satisfied in the "post-state" (at the next time step, $t + 1$), and all other literals must take the same value in the post-state as in the pre-state, which is guaranteed by "frame axioms". We have a set of attributes indicating which action is taken at a given step t, one per action. For the purposes of learning the dynamics of a STRIPS planning domain, it is natural to consider an attribute space in which we only consider pairs of possible worlds at two consecutive time-steps, t and $t + 1$ for some t, since the world is deterministic, and the state at step $t + 1$ is entirely determined by the state at step t and the action taken. As we noted in Example 7.2, it is natural to suppose that the observations of an agent in such an environment are local, i.e., the agent can only "see" its immediate environment and, additionally, it knows it is nowhere else but its present location, and it knows which action it is taking. Let's examine what kind of clauses are witnessed true based on these assumptions.

We first note that since we have assumed that the agent knows which actions it is taking and which actions it is not taking, every precondition and effect clause for an action j' that the agent is not taking is witnessed true, by the literal $\neg A_{j'}$ in the respective clause; this only leaves the precondition and effect clauses for the relevant action j. These clauses are witnessed true if and only if the environment attribute literals are observed: for j to be taken, each of the respective literals (at time t for the preconditions and time $t + 1$ for the effects) is true in the environment, so as long as they are observed, the respective clauses are indeed witnessed true. Hence, if the observed attributes are those that are visible to the agent, then these clauses are witnessed if and only if the attributes used in the preconditions and effects of an action are visible whenever the action may be taken. Note that this would be the case for our box-moving robot: the robot can pick up or put down a box only if it is in the same room as the robot, and hence visible to it.

The frame axioms, however, are a different story. Recall that for a given literal ℓ, if $\mathcal{A}_\ell$ denotes the set of actions that have ℓ as an effect, then the frame axiom for ℓ is given by the clause

$$\bigvee_{j \in \mathcal{A}_\ell} A_{j,t+1} \vee \ell_t \vee \neg \ell_{t+1}$$

i.e., either ℓ is an effect of the action taken to reach step $t+1$, or ℓ was already true at step t, or else it is (it remains) false at step $t+1$. Note that we have supposed that the agent cannot observe attributes describing the state of objects in other rooms. The effects of the actions available to the robot cannot affect the state of these objects either; hence, for the literals ℓ describing these objects, the actions available to the robot are not in $\mathcal{A}_\ell$, so the literals corresponding to these actions cannot witness the frame axiom: the first group of literals are all false. Furthermore, again, neither ℓ_t nor ℓ_{t+1} are observed—hence, the frame axiom is *not* witnessed in general. As a consequence, the definitions we consider here won't enable us to learn the frame axioms. Recall that the frame axiom expresses the common sense law of inertia in this domain; we will need some other mechanism to provide such inferences.

Another view of what is crucial about the definition of witnessed values is that unlike satisfiability, witnessing is locally determined. Indeed, as we saw, the partial example must actually indicate a falsified literal in each clause. As we'll see, it turns out that this additional information is enough for us to determine values for the features in more general enriched feature spaces that are not read-once. That is, it will be sufficient for the masking process to satisfy the following, stronger requirement:

Definition 7.17 We say that a masking process M is η-*revealing* with respect to a distribution D on labeled possible worlds and class of De Morgan formulas C if for all $c \in C$,

$$\Pr_{m\in M,(\vec{x},b)\in D}\left[[c(\vec{X}) = B]|_{m(\vec{x},b)} = \bot \middle| c(\vec{x}) \neq b\right] \geq \eta$$

where $c(\vec{X}) = B$ is an abbreviation for the formula $[c(\vec{X}) \Rightarrow B] \wedge [B \Rightarrow c(\vec{X})]$.

That is, if $c(\vec{x}) \neq b$ (for $(\vec{x}, b)$ from D), with probability at least η either:

- $b = 1$ and $c(\vec{X})|_{m(\vec{x},b)} = \bot$ or
- $b = 0$ and $c(\vec{X})|_{m(\vec{x},b)} = \top$.

Recall that we used the following strategy for learning k-CNF/k-DNF: we took our algorithm for finding a consistent conjunction or disjunction (respectively), and applied it on a transformed set of attributes, in which we took the values of all possible small-width disjunctions/conjunctions as our features. That is, we viewed these formulas as being obtained by a *base* classifier – the single, large conjunction or disjunction for which we solve consistency – composed with members of a small set of possible *features*. If the distribution on partial examples is revealing with respect to the class of compositions, a sufficient collection of feature maps will be determined to rule out wrong candidates:

Theorem 7.18 *Let C be a class of De Morgan formulas obtained by composing a member of a class $\mathcal{B}$ of monotone De Morgan formulas, for which we can solve* $\text{Consis}_{\mathcal{B}}^{s(n,m)}$ *in polynomial time, with a class of De Morgan formulas $\mathcal{F}$. Suppose M is a masking process that is η-revealing with respect to a distribution D on labeled possible worlds for C. Then there is a algorithm that, given $m \geq \Omega(\frac{1}{\eta\epsilon}(s(|\mathcal{F}|, m) +$*

$\log \frac{1}{\delta}))$ *partial examples with labels from some* $c \in C$*, runs in time polynomial in* n*,* m*, and* $\sum_{f \in \mathcal{F}} |f|$ *and returns a representation* $h \in C$ *such that* $\Pr_{(x,b)\in D}[h(\vec{X}) = B] \geq 1 - \epsilon$ *with probability* $1 - \delta$*.*

Proof As sketched above, we construct a new distribution on possible worlds with features indexed by $\mathcal{F}$; given a partial example ρ on the original set of features with observed label b, we construct a transformed partial example τ indexed by $\mathcal{F}$ such that $\tau_f = 1$ if $f|_\rho = \top$, $\tau_f = 0$ if $f|_\rho = \bot$, and $\tau_f = b$ otherwise. This transformation may be computed in time $O(\sum_{f\in\mathcal{F}} |f|)$ per example, and creates examples of size $|\mathcal{F}| \leq \sum_{f\in\mathcal{F}} |f|$. Since M is η-revealing, if $\Pr[\tilde{c}(\vec{X}) \neq c(\vec{X})] \geq \epsilon$, $\Pr[\tilde{c}|_\rho \in \{\top, \bot\}$ and $\tilde{c}|_\rho \neq c|_{\vec{X}=\vec{x}}] \geq \eta\epsilon$, where observe, since $\tilde{c}|_\rho$ has its value witnessed, in particular τ is sufficient to determine the value of the base formula of $\tilde{c}$ (or else it would not be witnessed). Thus, in $\frac{\ln 2}{\eta\epsilon}(s(|\mathcal{F}|, m) + \log_2 \frac{1}{\delta})$ examples, the probability of such base classifiers remaining solutions is at most $\frac{\delta}{2^{s(|\mathcal{F}|,m)}}$, and these may be output with total probability at most δ. Moreover, as in the proof of Theorem 7.11, the base formula of c also has label b on τ, so it remains a solution to $\text{Consis}_{\mathcal{B}}^{s(|\mathcal{F}|,m)}$. The algorithm will thus obtain such a representation with probability at least $1 - \delta$ as claimed. □

In particular, since the set of terms/clauses of width k on n attributes is $O(n^k)$ and the consistency problem for monotone disjunctions and conjunctions are solvable by Elimination in linear time, we indeed find that k-CNF and k-DNF are learnable for revealing masking processes, in time similar to the time to learn them in the usual full-information setting.

Chapter notes

The model of partial observations presented in Section 7.1 was essentially first proposed by Rubin (1976), who was primarily concerned with identifying conditions under which, for example, naïve averages of the observed values are accurate estimators. The approach to learning from partial examples we present in Section 7.2 is from Michael (2010). A related model, in which the masking is required to reveal enough attribute values to determine the label, was considered earlier by Goldman et al. (2003). (Goldman et al. actually furthermore assume the ability to query the label of arbitrary examples, i.e., given "membership queries.") The Completion Algorithm for learning monotone classes (Algorithm 12/Theorem 7.11) and the results on the intractability of proper learning from partial observations in Section 7.4 are also from Michael (2010). We note that there is a gap in the proof of Corollary 4.8 in this paper, on learning read-once k-CNF formulas, etc.: for k-CNF, the reduction requires properly learning a subset of monotone conjunctions in which certain pairs of attributes are mutually exclusive, which unfortunately is intractable. It is an open problem whether read-once k-CNF are learnable in this model. Theorem 7.12 on learning halfspaces, which uses the observation that we can always obtain such halfspaces (by collapsing the coefficients of literals), is new. The results of Section 7.5 are also previously unpublished.

Exercises

7.1 *Computing partial evaluations.* Let the attributes be

S = "there is smoke", F = "there is fire"
A = "the alarm sounds", B = "the battery is charged",
T = "the sprinkler turns on".

A partial example is a vector in $\{0, 1, *\}^5$, where $*$ means that the value of the attribute is not observed. Consider the following partial examples:

$$\rho^{(1)} = (1, *, 1, 1, *),$$
$$\rho^{(2)} = (0, *, *, 1, 0),$$
$$\rho^{(3)} = (*, 0, 1, *, *),$$
$$\rho^{(4)} = (1, *, *, 0, *),$$

where the coordinates are ordered as (S, F, A, B, T). For each formula below and each partial example $\rho^{(j)}$, say whether the formula is forced true, forced false, or undetermined by the partial example. A formula is forced true if it is true in every complete world consistent with the partial example, and forced false if it is false in every complete world consistent with the partial example.

$\varphi_1 = S \wedge B$, $\varphi_2 = S \vee F$,
$\varphi_3 = (S \wedge B) \supset A$, $\varphi_4 = A \wedge \neg B$,
$\varphi_5 = (F \vee S) \wedge \neg T$.

(a) Fill in a table with rows $\rho^{(1)}, \ldots, \rho^{(4)}$ and columns $\varphi_1, \ldots, \varphi_5$, using the entries 1, 0, and $*$ to indicate whether the formula in that column is, respectively, forced true, forced false, or undetermined on the partial example in that row.
(b) For one entry marked $*$, give two complete worlds consistent with the partial example: one in which the formula is true and one in which it is false.
(c) For one formula that is forced true on some partial example, explain why every completion of that partial example satisfies the formula.
(d) Explain why treating every not-observed value $*$ as 0 can give the wrong answer for partial evaluation. Give an example from the above.

7.2 *Setting up a masking process.* Continue with the fire-alarm domain from Exercise 7.1. Consider the following three masking functions: $m_{\text{full}}(x) = x$ (as an element of $\{0, 1, *\}^5$). The masking function m_{visible} reveals S, A, and T, but hides F and B. The masking function m_{alarm} reveals all attributes when $A = 1$, but hides F and B when $A = 0$, while still revealing S, A, and T. Let M be the masking process given by

$$M = \begin{cases} m_{\text{full}} & \text{with probability } 1/2, \\ m_{\text{visible}} & \text{with probability } 1/3, \\ m_{\text{alarm}} & \text{with probability } 1/6 \end{cases}.$$

(a) For a vector $(s, f, a, b, t) \in \{0, 1\}^5$, write the value of $m_{\text{visible}}(s, f, a, b, t)$ explicitly as a vector in $\{0, 1, *\}^5$.

(b) For vectors $(s, f, 0, b, t) \in \{0,1\}^5$ and $(s, f, 1, b, t) \in \{0,1\}^5$ (i.e., where $a = 0$ and $a = 1$, respectively), write the respective values of $m_{\text{alarm}}(s, f, a, b, t)$ explicitly as vectors in $\{0, 1, *\}^5$.
(c) For the complete world

$$\vec{x} = (1, 0, 1, 1, 0),$$

list all partial examples that can be produced by the masking process M, and give the probability of each.
(d) Repeat part (c) for the complete world

$$\vec{x}' = (1, 0, 0, 1, 0).$$

(e) For each of the two worlds $\vec{x}$ and $\vec{x}'$, compute the probability that the F attribute is observed.
(f) Explain why this is still a valid masking process even though the probability that the attribute F is observed can depend on the value of the attribute A.

7.3 *Masking processes capture all partial observation distributions.* In our partial information framework, we made the mathematically convenient assumption that the masking process is independent of the distribution on possible worlds. In this exercise, we will see that this is essentially without loss of generality. Consider any pair of jointly distributed random variables (R, D) where R takes values in $\{0, 1, *\}^n$ and D takes values in $\{0,1\}^n$, such that in all $(\rho, \vec{x})$ in the support of (R, D), ρ is consistent with $\vec{x}$. Prove that there is a masking process M such that independently sampling $\vec{x}$ from D and m from M yields $(m(\vec{x}), \vec{x})$ that is distributed identically to (R, D).

7.4 *Simulating nondeterministic functions with partial examples.* In this exercise, we see how learning problems that appear nondeterministic from partial examples can arise from deterministic target attributes over richer possible worlds. Let $X_1, \ldots, X_n$ be the observed Boolean attributes, and let X_t be a target attribute.
(a) A *switching concept* is specified by a set of Boolean functions

$$c_1, \ldots, c_k : \{0,1\}^n \to \{0,1\}.$$

For each possible world $\vec{a} \in \{0,1\}^n$, the label is equal to $c_j(\vec{a})$ for some $j \in \{1, \ldots, k\}$, but the index j is not observed.
Show how to simulate a switching concept using partial examples. More precisely, for a given distribution D on the n attributes $X_1, \ldots, X_n$, and any combination of choices of $j(\vec{a})$ for each $\vec{a} \in \{0,1\}^n$, introduce additional hidden Boolean attributes, define a distribution D' over complete possible worlds, define a masking process M, and define a deterministic Boolean function for X_t over all attributes, such that after applying M, the partial examples have the behavior of the switching concept. Your distribution D' should satisfy that

$$\Pr_{\vec{x} \in D}[X_1 = x_1, \ldots, X_n = x_n] = \Pr_{\vec{x} \in D'}[X_1 = x_1, \ldots, X_n = x_n]$$

and the value of the hidden attributes conditioned on the event that $(X_1, \ldots, X_n)$ take the value $\vec{x}$ should be deterministic (i.e., take a fixed value with probability 1).

Your construction should ensure that the observed attributes $X_1, \ldots, X_n$ and the target attribute X_t are revealed, but the information determining which concept c_j is active in each example is not observed. Be sure to indicate a class of functions C' that contains the function determining the value of X_t, given that $c_1, \ldots, c_k \in C$ for some class of Boolean functions C. (C' indeed should depend on C.)

(b) A *p-concept* is given by a function

$$p : \{0,1\}^n \to [0,1],$$

where on each example, the target attribute X_t is independently equal to 1 with probability $p(\vec{a})$, given that $\vec{X} = \vec{a}$. Show how to simulate a p-concept using partial examples. Assume, for simplicity, that there is an integer k such that

$$p : \{0,1\}^n \to \left\{0, \frac{1}{k}, \frac{2}{k}, \ldots, 1\right\}.$$

For any distribution D on $\vec{X}$, introduce additional hidden Boolean attributes that are independent of $\vec{X}$ and define a masking process M such that for every $p \in \mathcal{P}$ for some class of functions $\mathcal{P}$ taking $\{0,1\}^n \to \left\{0, \frac{1}{k}, \frac{2}{k}, \ldots, 1\right\}$ there is a deterministic Boolean function for X_t over all attributes, such that after applying M,

$$\Pr[X_t = 1 \mid \vec{X} = \vec{a}] = p(\vec{a}),$$

for every $\vec{a} \in \{0,1\}^n$. Be sure to indicate a class of functions C' that contains the function determining the value of X_t. (C' indeed should depend on $\mathcal{P}$.)

(c) In both constructions, explain why the value of X_t is deterministic in the complete possible world, even though it may appear nondeterministic from the partial examples alone. What information is hidden by the masking process, and why would revealing it remove the apparent nondeterminism?

7.5 *Consistent learning and Occam's razor for partial examples.* Recall that for a class of representations C on n attributes and representation length bound B, the search problem Consis_C^B is to find a representation $c \in C$ that fits a given set of labeled examples. Occam's Razor (Theorem 4.2) shows that if B is sufficiently small compared to the number of examples, any solution to Consis_C^B is a satisfactory solution to the learning problem using representations from C. Define a search problem that generalizes Consis_C^B to a set of labeled *partial* examples, and prove the analogue of Occam's Razor for your generalized search problem, that for a sufficiently large sample size m relative to B, any solution c to your problem meets the criteria of Definition 7.4.

7.6 *Improper learning of decision trees from revealing masking processes.* In Exercise 4.4, we developed algorithms for learning decision trees from complete information. Here, we'll generalize the methods to learn from partial information. In the case of partial information, unlike complete information, we *do not* have a (proper) learning algorithm for decision lists, so we will need to use the connection to linear threshold functions established in Exercise 4.5, which are learnable.

(a) Our definition of revealing masking processes was based on a definition of witnessing values under partial evaluation (Definition 7.16) that was tied to De Morgan

formulas. Here, we wish to use linear threshold functions as our basic classifier, which are *not* simple De Morgan formulas. Propose a generalization of Definition 7.16 to linear threshold functions, such that the linear threshold function is witnessed true/false (respectively) if the function is true/false (respectively) in every possible world consistent with the partial example. Prove that your generalization satisfies these conditions and is computable in polynomial time.

(b) Prove that for the generalization of revealing distributions using your notion of witnessed values from part (a), degree-d polynomial threshold functions (for a fixed value of d) on n attributes are learnable from η-revealing masking processes with polynomial sample complexity in n^d, $1/\epsilon$, $1/\eta$, and $\log 1/\delta$.

(c) Using part (b) and the results of Exercise 4.4(b) and Exercise 4.5(a), prove that rank-k decision trees are (improperly) learnable from revealing masking processes.