More Accurate Learning of k-DNF Reference Classes

Brendan Juba
Washington University in St. Louis
bjuba@wustl.edu

Hengxuan Li*
Facebook
lhx@fb.com

Abstract

In machine learning, predictors trained on a given data distribution are usually guaranteed to perform well for further examples from the same distribution on average. This often may involve disregarding or diminishing the predictive power on atypical examples; or, in more extreme cases, a data distribution may be composed of a mixture of individually "atypical" heterogeneous populations, and the kind of simple predictors we can train may find it difficult to fit all of these populations simultaneously. In such cases, we may wish to make predictions for an atypical point by selecting a suitable *reference class* for that point: a subset of the data that is "more similar" to the given query point in an appropriate sense. Closely related tasks also arise in applications such as diagnosis or explaining the output of classifiers. We present new algorithms for computing k-DNF reference classes and establish much stronger approximation guarantees for their error rates.

Introduction

In practice, every document, every scene, or every patient for which we wish to use machine learning to make predictions is atypical in its own way. Fortunately, these idiosyncracies are often irrelevant enough to the task at hand that we can ignore them, and make successful predictions with a relatively simple model. But, the dream of "personalized medicine," for example, is that we can somehow identify when a patient's case is atypical in some significant way. Rosenfeld et al. (2015) showed for example that there are subpopulations with risk factors for gastrointestinal cancer that are not significant risk factors in the overall population. For a given patient, we want to be able to identify what, if any, such subpopulations that patient belongs to that might yield such risk factors. The algorithm of Hainline et al. (2019) (discussed below) finds a subpopulation on which accurate prediction is possible. Thus, in particular, the linear predictors found by their algorithm can make use of these special risk factors for the subpopulation the patient belongs to. Or, in artificial intelligence applications, we may wish to identify when a scene or a conversation is atypical in some significant way, so that we can select similarly atypical examples to use for prediction. These similar examples are called a *reference class* for the example in question. Roth (1995) showed how, by filtering the data used for inference in an appropriate way – thus using an appropriate reference class – classic nonmonotonic reasoning problems may be solved easily.

In other words, a reference class captures the "relevant context" for probabilistic inference. This view of reference classes, and especially their role in artificial intelligence, was largely developed by Kyburg (1974) and Pollock (1990), building on a view of probability originally proposed by Reichenbach (1949). In Kyburg and Pollock's proposals, one should select the most specific available reference class, ruling out disjunctive reference classes.

Bacchus et al. (1996) provide a critical review of this framework (and the decision to disallow disjunctions). One of the main criticisms of the reference-class framework is that it is often unclear how to select an appropriate reference class for inference; making the class too specific may lead us with few prior examples or none at all. Such a problem almost always arises when reference classes are invoked. For example, in Valiant's "neuroidal" cognitive model (Valiant 1994; 2000), the class of scenes to be used to make a prediction (a reference class, essentially) are chosen by the currently "firing" model neurons, which could easily be overly specific. The key question is, how can we find a reference class for which *(i)* we have sufficient data to make an informed prediction, but that is nevertheless *(ii)* sufficiently specific to help inform our prediction?

Juba (2016) and Hainline et al. (2019) proposed a formulation of the reference class selection problem in the context of making certain kinds of predictions. Specifically, Juba considered a variant of reference class selection as a means to perform diagnosis, or more generally abduction: we try to find the *most likely* reference class in our overall data distribution from some specified family of representations such that a property of interest, to be diagnosed, holds. Hainline et al. subsequently sought to find a reference class on which to perform linear regression: they seek a class that is sufficiently likely, and such that the squared-error loss (or some other ℓ_p-loss) is approximately minimized. These works focus on algorithms for finding k-DNF reference classes (ORs of ANDs of at most k Boolean literals—attributes or their negations) since (as we observe here—see Proposition 2) solving such tasks for conjunctions, or any representation

*Work performed in part while affiliated with Washington University in St. Louis

Copyright © 2020, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

that can express general conjunctions, would enable PAC-learning of general DNFs in the standard, "distribution-free" model. Apart from being a notorious open problem, recent work by Daniely and Shalev-Shwartz (2016) shows that algorithms for learning DNF would have further strong consequences. (See also Durgin and Juba 2019.) It follows that among the usual, natural classes of representations, k-DNF is the most expressive class for which we have any hope of solving such problems.

Our contribution We present algorithms for finding k-DNF reference classes that achieve better approximations to the optimal loss than the $O(n^k)$-approximations presented by Juba (2016) and Hainline et al. (2019) based on the "tolerant elimination" algorithm (originally proposed for abduction), respectively for diagnosis and linear regression. Our algorithms are analogues of the improved algorithms for exception-tolerant abduction obtained by Juba et al. (2018) and Zhang et al. (2017). Our algorithms, like these, respectively obtain an $\tilde{O}(r \log\log n)$-approximation when there is an r-term k-DNF, and an $\tilde{O}(n^{k/2})$-approximation in general. We note that the first guarantee is particularly strong when the formula contains few terms. This of particular interest in cases where the formula is to be provided for human inspection, and thus must be small to be useful.

Both of the abduction algorithms are, at their core, based on approximation algorithms for variants of partial set cover. We observe that these algorithms only fail to solve the reference class problem because there is no guarantee that the partial covers found by these algorithms will include the point corresponding to the query. So, we consider two new variants of partial set cover in which there is a distinguished point that must be covered. In the standard, additive-cost variant of partial set cover used by Juba et al., which was first considered by Kearns (1990) and analyzed fully by Slavík (1997), this is almost immediate. But, the variant used by Zhang et al. considers the *ratio* of the cost to the number of elements covered, which is significantly more difficult. The main technical issue is that the sets containing the distinguished point may have a much worse cost ratio than any set the greedy algorithm would normally choose. As a consequence of this, many of the observations used in the analysis of the greedy algorithm for partial cover no longer hold; indeed, a naïve modification of the usual greedy partial set cover algorithm will not achieve the usual approximation ratio. It may be necessary to include sets covering significantly more than the bare minimum number of elements in order to "dilute" the cost of the set containing the distinguished element. Nevertheless, we show how to repair the greedy algorithm and give an analysis.

We stress that Zhang et al. showed that tolerant elimination performed far worse than their abduction algorithm on a synthetic data task, and moreover, Qi et al. (2018) obtained similar findings when using the algorithms to perform a real-world webcam anomaly explanation application. It seems that tolerant elimination fails in all but simple cases where there is a planted solution with a small amount of independent noise. Indeed, tolerant elimination completely failed at anomaly explanation, whereas the algorithm of Zhang et al. was quantitatively competitive with the random forest baseline they considered. We include an empirical demonstration that our large formula algorithm can be used in practice for a similar application, explaining the decisions of classifiers on a given point. We find that our algorithm can compete with the previous algorithms for this task (Ribeiro, Singh, and Guestrin 2016; 2018) while providing theoretical guarantees. Similarly, the small formula algorithm of Juba et al. (2018) was used for the conditional linear regression experiments of Hainline et al. (2019), and found to be effective in practice.

Relationship to other work Our work thus takes the approaches of Zhang et al. (2017) and Juba et al. (2018) for obtaining better guarantees for the abduction task that does not make reference to a specific query point x^*, and adapts them to solve the reference class search problem. The previous algorithms for finding reference classes, by Juba (2016) and Hainline et al. (2019), only found $O(n^k)$-approximations.

Beyond these directly relevant works, the most similar work to our reference class search algorithms is work on producing "explanations" of anomalies or classifications, see for example the survey by Biran and Cotton (2017) for an overview of such work. While our reference class search algorithms could be used for such problems, most of these works are specific to diagnosing the decisions made by a given classifier based on its structure, and often focus on identifying which attributes were most influential, whereas our formulation and algorithms are generic. The closest is work by Ribeiro et al. (2016), which is not tied to a specific classifier. Given a query point, they sample points nearby in the feature space, and train a simpler classifier on this neighborhood. Needless to say, the semantics of these "explanations" are quite different. The follow-up work by Ribeiro et al. (2018) again samples in the neighborhood of a point of interest, but seeks to optimize precision and coverage objectives that are very similar to ours, using conjunctive rules. Nevertheless, their objective is often not tied to the actual data distribution (but rather, a synthetic "neighborhood" distribution), in contrast to ours, and they do not attempt to provide theoretical guarantees. As we have noted, we believe it is not possible to provide strong theoretical guarantees for conjunctions in general, in contrast to k-DNFs.

Preliminaries

A basic version of the task we consider (from Juba 2016) is as follows. We will work in a PAC-learning style framework, in which the data is drawn i.i.d. from an arbitrary distribution D over n Boolean attributes. (Later we will also extend this to consider regression tasks in which there are additional, real-valued attributes.) We suppose that we are given a query point x^*, for which we are interested in predictions, and a property $c(x)$ that we would like to be true of our reference class. We would then like to find a formula h describing as likely an event as possible under D that, in particular, includes x^*, and such that $c(x)$ is always true when $h(x)$ is. Formally:

Definition 1 *The* (optimal) reference class search *task for a class of representations $\mathcal{H}$ over n attributes is as follows. Given an* observation *example $x^* \in \{0,1\}^n$ and a* query *representation c for some arbitrary distribution D over $\{0,1\}^n$ such that $D(x^*) > 0$, if there exists some $h^* \in \mathcal{H}$ such that $h^*(x^*) = 1$, $\Pr[c(x) = 1 | h^*(x) = 1] = 1$, and $\Pr[h^*(x) = 1] \geq \mu$, using examples drawn from D, find a circuit h in time polynomial in n, $1/\epsilon$, $1/\mu$, and $1/\delta$ such that with probability $1 - \delta$ over the examples, $h(x^*) = 1$, $\Pr[c(x) = 1 | h(x) = 1] \geq 1 - \epsilon$, and $\Pr[h(x) = 1] \geq (1 - \epsilon)\mu$. Such an h is said to be a* reference class *for x^* relative to c. The* learning to abduce *task is the variant of this task in which we do not require either $h^*(x^*) = 1$ or $h(x^*) = 1$ (i.e., in which there is no query point).*

Such reference classes can be used to propose candidate diagnoses as follows: $c(x)$ represents some condition that we wish to diagnose. We typically restrict the attributes that $\mathcal{H}$ may use to avoid trivial diagnoses such as $c(x)$ itself, or we may think of $c(x)$ as being represented by a separate label attribute b. Then the output of reference class search is a formula $h(x)$ such that $h(x)$ is often true, empirically, such that $h(x)$ empirically entails $c(x)$, and such that $h(x)$ is true of x^* in particular. Or, in "learning to reason" (Valiant 1994; Roth 1995; Valiant 2000) it may be used to test if there exists an optimistic "precondition" for reasoning under which $c(x)$ would be true in the specific instance x^*; roughly, whether or not there is a plausible justification for inferring $c(x^*)$.

For example, let's consider the anomaly explanation setting of Qi et al. (2018), which was built on the learning to abduce task. There, an anomaly detector computes a label for images from a webcam indicating whether or not these images are anomalies; Qi et al. considered a vector of meta-data attributes x associated with each image, and searched for rules $h(x)$ (i.e., formulated in terms of these meta-data attributes) that reliably indicated that a μ-fraction of the examples were classified as anomalies. For example, the meta-data attributes may indicate the time of day, weather, or presence or absence of certain kinds of objects in the image, and the "explanations" are formulated in terms of these attributes. In our variant of the task, we may be interested in why a specific image was classified as an anomaly. By presenting the vector of meta-data attributes x^* associated with this image in the reference class task, we then should find some rule $h(x)$ that not only is a reliable indicator that the corresponding images will be labeled as anomalies, but also that the image in question (with metadata x^*) in particular is included among these images.

We will focus on cases where $\mathcal{H}$ is taken to be the class of k-DNFs, ORs of ANDs of at most k "literals," Boolean attributes or their negations. The reason is that it follows from results by Juba (2016) or earlier results by Bshouty and Burroughs (2005) that *any* $\mathcal{H}$ that can represent arbitrary conjunctions (ANDs of literals) gives rise to an intractable problem. Indeed, they show that if the learning to abduce task for conjunctions can be solved in polynomial time, then supervised PAC-learning of (general) DNF can also be solved in polynomial time. In addition to being a notoriously difficult problem, Daniely and Shalev-Shwartz (2016) presented evidence that this task is intractable. (We can obtain slightly stronger consequences for our task, see Durgin and Juba 2019 for details.) These results carry over to reference classes, so we likewise restrict our attention to k-DNFs.

Proposition 2 *Suppose that reference class search can be solved for some $\mathcal{H}$ that can express conjunctions. Then there is a polynomial time algorithm for PAC-learning DNF.*

Proof: We show that the abduction task for conjunctions can be reduced to the reference class task; the claim then follows from the analogous reduction of PAC-learning of DNF to abduction of conjunctions (Juba 2016, Theorem 5).

The reduction is as follows. Given a set of examples $(x, b)^{(1)}, \ldots, (x, b)^{(m)}$ on $\{0,1\}^{n+1}$ (where b indicates the condition to be explained), we construct examples $(y, b)^{(1)}, \ldots, (y, b)^{(m)}$ on $\{0,1\}^{2n+1}$ by taking the first n attributes of $y^{(i)}$ to be identical to the corresponding $x^{(i)}$, and taking the final n attributes to be negations of the first n attributes. Now, we take our distinguished point to be $(1, 1, \ldots, 1)$, and try to solve the reference task class on this input. Suppose the algorithm returns a circuit $C(y)$. We convert this to a circuit $C'(x) = C(x_1, \ldots, x_n, \neg x_1, \ldots, \neg x_n)$, and return this as a solution for (improper) abduction. Note that if the reference class algorithm was "proper," ie., actually returns a conjunction C, then C' is also a conjunction and hence the resulting abduction algorithm is also "proper."

Observe that if there is a conjunction $C^*(x)$ for the abduction task (occupying probability μ and with error rate ϵ), then the following monotone (negation-free) conjunction C^{**} is a solution to the reference class task: if C^* contains x_i, put y_i in C^{**}, and if C^* contains $\neg x_i$, then put y_{i+n} in C^{**}. Since C^{**} is monotone, it is satisfied by $(1, 1, \ldots, 1)$, and it is true on our distribution on y with precisely the same probability and precision as C^* is for the original distribution on x by construction. Thus, assuming the correctness of our algorithm for reference classes, it returns a circuit $C(y)$ solving the reference class task. But again, $C'(x)$ similarly has the same probability and precision as $C(y)$, and hence it is also a solution to the abduction task as needed. ■

We remark that this task remains hard even if we weaken our requirement on h to only have probability polynomially related to h^* in the various parameters.

The basic reference class task may be extended in multiple ways. Juba et al. (2018), for example, consider a partial information version of the task. While it is possible to extend our results to address the partial information analogues of the reference class problems, since the setting is rather involved we leave this extension to the interested reader. A second natural extension, which we will consider, is the *weighted* variant, first introduced by Hainline et al. (2019):

Definition 3 *The* weighted k-DNF reference class search *task is as follows. Given an* observation *example $x^* \in \{0,1\}^n$ and an arbitrary distribution D over $\{0,1\}^n \times [0, b]$ such that $\Pr_D[x^*] > 0$, if there exists some k-DNF h^* such that $h^*(x^*) = 1$, $\mathbb{E}_{(x,w)\in D}[w | h^*(x) = 1] \leq \epsilon^*$, and $\Pr_D[h^*(x) = 1] \geq \mu$, using examples drawn from D, find a circuit h in time polynomial in n, $\max\{b, 1/b\}$,*

$1/\epsilon_0$, $1/\gamma$, $1/\mu$, *and* $1/\delta$ *such that with probability* $1-\delta$ *over the examples,* $h(x^*) = 1$, $\mathbb{E}_D[w|h(x) = 1] \leq \alpha(n, \epsilon_0, b, \mu, \delta, \gamma)) \max\{\epsilon_0, \epsilon^*\}$, *and* $\Pr[h(x) = 1] \geq (1-\gamma)\mu$. *We say that* α *is the* approximation factor *for the* reference class h.

In applications, the weights w represent some loss value, such as prediction error, for including the example in the reference class. Using solutions to the weighted reference class task, one can solve richer tasks such as the reference class regression task introduced by Hainline et al. (2019), below:

Definition 4 Reference class ℓ_p-norm regression *is the following task. We are given a* query point $x^* \in \{0,1\}^n$, *target density* $\mu \in (0,1)$, *ideal loss bound* $\epsilon_0 > 0$ *approximation parameter* $\eta > 0$, *confidence parameter* $\delta > 0$, *and access to i.i.d. examples drawn from a joint distribution over* $(x, y, z) \in \{0,1\}^n \times \{y \in \mathbb{R}^d : \|y\|_2 \leq b\} \times [-b, b]$. *We wish to find* $\hat{a} \in \mathbb{R}^d$ *with* $\|\hat{a}\|_2 \leq b$ *and a* reference class k*-DNF* $\hat{h}$ *such that with probability* $1-\delta$,

1. $\hat{h}(\vec{x}^*) = 1$,
2. $\Pr[\hat{h}(x) = 1] \geq (1-\eta)\mu$, *and*
3. *for a fixed* approximation factor $\alpha > 1$, $\mathbb{E}[|\langle \hat{a}, y\rangle - z|^p | \hat{h}(x) = 1]^{1/p} \leq \alpha \max\{\epsilon^*, \epsilon_0\}$ *where* ϵ^* *is the optimal* ℓ_p *loss* $\mathbb{E}[|\langle a^*, y\rangle - z|^p | h^*(x)]^{1/p}$ *over* $a^* \in \mathbb{R}^d$ *of* $\|a^*\|_2 \leq b$ *and* k*-DNFs* h^* *such that* $h^*(x^*) = 1$ *and* $\Pr[h^*(x) = 1] \geq \mu$.

If we also require both $\hat{a}$ *and* a^* *to have at most* s *nonzero components, then this is the* reference class s-sparse ℓ_p-norm regression *task.*

It follows from the reductions of Proposition 2 and Theorem 7 of Juba (2017) that solving the reference class regression task requires solving the corresponding reference class task; thus, we again must focus on k-DNFs.

Reference classes for small formulas

We first consider the variant of the task in which the k-DNF is known to be small, using at most r terms. Juba et al. (2018) considered abduction for such small formulas, and obtained a $\tilde{O}(r \log\log n)$-approximation to the optimal error rate. We will see that it is easy to extend their approach to obtain such guarantees for reference classes.

The first ingredient in their approach is the sample complexity bound of Haussler (1988) for r-term k-DNFs: he found that only $O(\frac{rk}{\epsilon} \log \frac{n}{\delta})$ examples are statistically necessary; an essentially similar analysis shows that $m = O(\frac{brk}{\epsilon_0 \gamma^2} \log \frac{n}{\delta})$ examples suffice to guarantee an error rate that is competitive with $\epsilon^* \geq \epsilon_0$. But, as also observed by Haussler, actually finding a k-DNF with this fixed number of terms is an NP-hard problem; by allowing a slight increase in the number of terms to $r \log m$ (when an r-term formula exists), and thus using slightly more examples, computationally efficient greedy algorithms can be used. We will also follow this approach; see Juba et al. (2018) for a more careful derivation of the sample complexity bound for the related abduction task. Concretely, $m = O\left(\frac{brk}{\mu\epsilon\gamma^2} \log \frac{n}{\delta} \log\left(\frac{brk}{\epsilon\gamma^2} \log \frac{n}{\delta}\right)\right)$ examples will suffice.

Since the weights are nonnegative, if h^* has error ϵ^*, it follows that every term of h^* must also give error at most $\mu\epsilon^*$ in the distribution. So, any term that has an empirical error rate significantly greater than $\mu\epsilon^*$ cannot be in h^*, and we can safely ignore such terms. Now, Juba et al. treat the problem as an instance of unweighted partial set cover on a universe of size m (i.e., the examples): the sets correspond to the terms with error at most $(1+\gamma)\mu\epsilon^*$, where the set "covers" an example in the training set if it satisfies the corresponding term. Slavík (1997) shows that when we seek to cover a μ-fraction of the universe, the greedy algorithm (which simply chooses the set that covers the most remaining elements until μm elements have been covered) obtains a $H(\mu m)$-approximation to the smallest cover, where $H(i) = \sum_{j=1}^{i} \frac{1}{j}$. Thus, since h^* covers so many elements with a cover of size r, the greedy algorithm finds a cover of size $rH(\mu m) = O\left(r \log\left(\frac{brk}{\epsilon_0\gamma^2} \log \frac{n}{\delta}\right)\right)$. Due to the guarantee that the individual terms have an error rate of at most $\mu\epsilon^*$, the resulting formula has a total error rate of at most $O(rH(\mu m)\mu\epsilon^*)$, and is satisfied with probability at least $(1-\gamma)\mu$ overall; thus, it achieves a $O(rH(\mu m))$-approximation to h^*.

The only problem with the algorithm of Juba et al. for the reference class selection problem is that the formula they construct is not necessarily satisfied by x^*, so it may not be a reference class. We observe that it suffices to substitute an algorithm for partial set cover that is guaranteed to cover a distinguished point x^* for their partial set cover algorithm:

Definition 5 *Given a universe* $U = \{x_1, \cdots, x_m\}$ *of* m *elements, and collection* $S = \{S_1, \cdots, S_N\}$ *of subsets of* U, *the* Partial set cover task with a must-cover element x^* *is to find a subcollection* T *of* S, *such that* T *contains the specified element* x^*, T *covers a* μ*-fraction of* U, *i.e.,* $|\bigcup_{S_i \in T} S_i| \geq \mu|U|$, *and such that* $|T|$ *is minimized.*

Our modification is that after we run the greedy algorithm for partial set cover, if the cover it finds does not include x^*, we include an arbitrary set that covers x^*. Note that since $h^*(x^*) = 1$ and the terms of h^* have empirical error at most $(1+\gamma)\mu\epsilon^*$, such a term must exist. The formula now has at most one more term, so it has at most $rH(\mu m) + 1 = O\left(r \log\left(\frac{brk}{\epsilon_0\gamma^2} \log \frac{n}{\delta}\right)\right)$ terms. By a union bound over the terms, the formula we have constructed therefore has error at most $O((1+\gamma)(rH(\mu m)+1)\epsilon) = \tilde{O}\left((1+\gamma)r \log\left(\frac{brk}{\epsilon_0\gamma^2} \log \frac{n}{\delta}\right)\epsilon^*\right)$. This is our first theorem:

Theorem 6 *There is a polynomial-time algorithm for weighted reference class search when the condition has* r *terms using* $m = \tilde{O}(\frac{brk}{\mu\epsilon\gamma^2} \log \frac{n}{\delta})$ *examples achieving an* $\tilde{O}(r \log m)$*-approximation using a* k*-DNF with* $\tilde{O}(r \log m)$ *terms with probability* $1-\delta$.

Hainline et al. (2019) observe that we can obtain algorithms for reference class regression by simply plugging in algorithms for reference class search into an algorithm that first produces a list of candidate parameter vectors, and then uses the reference class search algorithm to winnow the list down to parameter vectors that provide low loss on a large

subset. Thus, given our algorithm for the weighted reference class search problem with this improved guarantee, we can follow the same strategy to immediately obtain an improved algorithm for reference class regression:

Theorem 7 *For any constant s and $\gamma > 0$, and $m = \tilde{\Theta}\left(\frac{1}{\mu}\left(\frac{brk}{\epsilon_0\gamma^2}\log\frac{nd}{\delta} + \frac{b^{2p}}{\epsilon_0^{2p}}(b^2 + \ln\frac{1}{\delta})\right)\right)$ examples, if there is a r-term k-DNF solution to the s-sparse ℓ_p reference-class regression task, then using our algorithm for reference class search in the algorithm of Hainline et al. (2019) gives an algorithm that runs in polynomial time and solves the task with a $\tilde{O}(r\log m)$-term k-DNF with $\alpha = \tilde{O}(r\log m)$.*

Specifically, the sample complexity bound is obtained from the guarantee of Theorem 6 by replacing δ with $\frac{\delta\gamma}{dm_0}$ where $m_0 = O\left(\frac{1}{\mu}\frac{b^{2p}}{\epsilon_0^{2p}}(b^2 + \log\frac{1}{\delta})\right)$, and adding m_0 to the resulting expression. So the sample complexity bound is actually $O\left(\frac{brk}{\mu\epsilon\gamma^2}\log\frac{ndm_0}{\delta\gamma}\log\left(\frac{brk}{\epsilon\gamma^2}\log\frac{ndm_0}{\delta\gamma}\right) + m_0\right)$ and the approximation ratio α is actually

$$O\left((1+\gamma)r\log\left(\frac{brk}{\epsilon\gamma^2}\log\frac{ndm_0}{\delta\gamma}\log\left(\frac{brk}{\epsilon\gamma^2}\log\frac{ndm_0}{\delta\gamma}\right)\right)\right).$$

Large k-DNF reference classes

We now consider algorithms for finding reference classes that scale better with the number of terms: using the approach of Zhang et al. (2017), we show that it is possible to obtain $\tilde{O}(\sqrt{n^k})$-approximate reference classes. When the reference class contains more than $\sim\sqrt{n^k}$ terms, this is superior to the algorithms of the previous section.

Zhang et al. cast the abduction task as a *red-blue partial set cover* problem with a *ratio objective*, which they solve, building on earlier work by Peleg (2007). In the variant considered by Zhang et al., the universe consists of two types of elements, red elements and blue elements, and the objective is to cover a prescribed number of blue elements, while minimizing the ratio of the number of red elements covered to the number of blue elements covered. The appropriate generalization of this problem to a weighted problem for use in the regression task is by instead assigning each "red" element a weight in the range $[0, b]$. The objective is then to minimize the ratio of the total weight covered to the number of blue points covered, given that the desired minimum number of blue points have been covered. In any case, at its core, Zhang et al's algorithm solves a variant of weighted partial set-cover in which the objective is to minimize the ratio of the total weight to the number of points covered, again subject to the restriction that a sufficient number of points are covered.

Unlike the approach of the previous section, however, the extension is now nontrivial. If we try to modify the definition of the weighted set cover problem along the lines of Definition 5, simply adding the requirement that the distinguished point x^* must be included, a significant complication arises. Natural modifications of the standard greedy algorithm that only add another set with x^* – either the one with lowest effective ratio or the one that optimal cover uses – fail to satisfy the original approximation bound of $3H(\mu m)$. The difficulty is that if the sets containing the distinguished point all have high weight, an optimal cover may need to "dilute" the cost of using one of these sets to cover x^* by covering more than the minimum number of points.

Let's consider an example. Suppose on some universe U we want to cover at least 10 elements while minimizing the cost-to-size ratio. This is a case of the original problem in Zhang et al., and we can construct an example in which the optimal cover has cost 70 and size 100, while the greedy cover has cost 10 and size 10. This is achievable, as discussed in Lemma 11 in the appendix of Zhang et al. Now we want to add a new set to U that only contains x^* and has cost 10000 (x^* is not contained in any other sets). Then we want a cover with at least 10 elements including x^* (the new problem). To satisfy the requirements we must use the new set in our cover though its cost-to-size ratio is large. If we simply add the new set to the original greedy cover, then the ratio of the new greedy cover is $\frac{10010}{11}$, while the cover constructed by adding the new set to the original optimal cover achieves a ratio of $\frac{10070}{101}$. The original approximation bound thus no longer holds.

To summarize this example, the set with x^* we added could break the approximation guarantee because the choice of adding that set to the cover does not necessarily satisfy the properties of the greedy choices used by Zhang et al. If the cost-to-size ratio of that set is very large and the size of greedy cover is too small, then the added set will dominate the combined ratio. Thus, moreover, the properties typically used to analyze the greedy algorithm are not satisfied by this problem, and we need a different analysis. We therefore present a modified greedy algorithm to solve our ratio variant of weighted partial set cover with a distinguished point that we must cover. Similar to the previous section, given an algorithm for our new variant of set cover, it is then relatively straightforward to obtain algorithms for our reference class tasks that obtain guarantees that are similar to those of Zhang et al. Thus, the heart of our approach is to solve the following problem.

Definition 8 *Given a universe $U = \{x_1, \ldots, x_m\}$ of m elements with weights $w_1, \ldots, w_m$ such that each $w_j \in [0, b]$, and collection $S = \{S_1, \ldots, S_N\}$ of subsets of U, the* partial weighted set cover task with ratio objective and must-cover element *x^* is to find a subcollection T of S, such that T contains the specified element x^* and T covers a μ-fraction of U, i.e., $|\bigcup_{S_i\in T} S_i| \geq \mu|U|$, minimizing the ratio $\frac{\sum_{S_i\in T} w_i}{|\bigcup_{S_i\in T} S_i|}$.*

We propose Algorithm 1 as a solution.

Theorem 9 *Let $\mathcal{T}$ be a collection of sets $T_1 \ldots, T_d$ on a universe V with corresponding weights $\omega(T_1), \ldots, \omega(T_d)$. Suppose that there is a sub-collection $\mathcal{T}^* \subseteq \mathcal{T}$ such that $T^* = \bigcup_{T\in\mathcal{T}^*} T$ contains at least $\mu|V|$ distinct elements and a specified element x^*, and $\sum_{T\in\mathcal{T}^*}\omega(T) = \omega(\mathcal{T}^*)$. Then Algorithm 1 finds a subcollection $\tilde{\mathcal{T}}$ such that $\bigcup_{T\in\tilde{\mathcal{T}}} T$ also contains at least $\mu|V|$ elements and the specified element x^*, and $\frac{\sum_{T_t\in\tilde{\mathcal{T}}}\omega(T_t)}{|\bigcup_{T_t\in\tilde{\mathcal{T}}} T_t|} \leq 3H(|V|)\cdot\frac{\omega(\mathcal{T}^*)}{|T^*|}$.*

Input: finite set $\mathcal{T} = \{T_1, ..., T_d\}$, costs $\{c_1, ..., c_d\}, \mu \in (0, 1]$, target element x^*
Output: μ-partial cover solution set $\tilde{\mathcal{T}}$
for *every set T_i that contains the target element x^** **do**
  Set $\tilde{\mathcal{T}} = \{T_i\}$, and $T_t = T_t \setminus T_i$ for each $T_t \in \mathcal{T}$ except T_i.
  while $r = \mu m - \left|\bigcup_{t \in \tilde{\mathcal{T}}} T_t\right| > 0$ **do**
    Choose the first $T_j \in \mathcal{T} \setminus \tilde{\mathcal{T}}$ that minimizes $c_t/|T_t|$, for $t \in \mathcal{T} \setminus \tilde{\mathcal{T}}$ and $T_j \neq \emptyset$, and add T_j to $\tilde{\mathcal{T}}$
    for *each $T_t \in \mathcal{T}$ except T_j* **do** Set $T_t = T_t \setminus T_j$;
  end
  while *the cost-to-size ratio $\frac{cost(\tilde{\mathcal{T}})}{|\tilde{\mathcal{T}}|}$ does not increase* **do**
    Choose the first $T_j \in \mathcal{T} \setminus \tilde{\mathcal{T}}$ that minimizes $c_t/|T_t|$, for $t \in \mathcal{T} \setminus \tilde{\mathcal{T}}$ and $T_j \neq \emptyset$, and add T_j to $\tilde{\mathcal{T}}$
    for *each $T_t \in \mathcal{T}$ except T_j* **do** Set $T_t = T_t \setminus T_j$;
  end
end
Return $\tilde{\mathcal{T}}$ with the smallest approximation ratio $\frac{cost(\tilde{\mathcal{T}})}{|\tilde{\mathcal{T}}|}$

Algorithm 1: Partial Greedy Algorithm

Proof: Suppose some optimal set cover OPT contains the set T' that covered the target point x^*. Because we iterate through all the sets that contain x^*, we must have chosen T' as the first set to be added to $\tilde{\mathcal{T}}$ in some iteration. Let's consider this iteration only. For $O = OPT - T'$, we propose the following lemma:

Lemma 10 *If the cost to size ratio of a greedy cover G is bounded by $3H(m - |T'|)$ times of the cost to size ratio of another cover O on $U \setminus T'$ (both covers cover the arbitrary required number of elements), and the other cover O is no more than 3 times larger than the greedy cover G, then the cost to size ratio of $G \cup T'$ is bounded by $3H(m - |T'|)$ times of the cost to size ratio of another cover $O \cup T'$ on U.*

The proof of Lemma 10 is deferred to the appendix.

What remains to be done is to find a greedy cover that satisfies $|O| \leq 3|G|$. Suppose we could invoke the algorithm of Zhang et al. with target size $|O|$ on $U - T'$. Although we do not know $|O|$, if we keep adding elements following the algorithm, then it must be able to return a greedy cover containing more than or equal to $|O|$ elements at some step, where the greedy ordering to add sets guarantees this set cover will obtain an approximation ratio of $3H(|O|)$. Because $|O| \leq m - |T'|$, the existence of a qualifying greedy cover also implies that the greedy cover that achieves the minimum ratio during the run of greedy algorithm must also be bounded by the approximation ratio of $3H(m - |T'|)$.

The reason that we can stop adding sets when the cost-to-size ratio $\frac{cost(\tilde{\mathcal{T}})}{|\tilde{\mathcal{T}}|}$ begins to increase is as follows. The ratio of the greedy cover is a weighted average of each chosen set's "effective" ratio $\frac{c_j}{|T_j|}$ at the iteration where we choose T_j to add to the cover. So, at this point the greedy ratio will only increase when we add more sets to the cover; otherwise it means we added a set with smaller "effective" ratio after a set with larger "effective" ratio, which contradicts our greedy ordering. (Note that the effective ratios of each set T_t only increase as $\mathcal{T}'$ covers more elements.) Then, even though we forced T' to be the first chosen set, which may violate the greedy ordering, the combined ratio will only have a unique local minimum (thus also the global minimum), which appears right before the combined ratio begins to increase by its minimality. We compute the combined cost-to-size ratio every time we add a set to the greedy cover after we have covered $\mu m - |T'|$ elements in $U \setminus T'$. Once we find that adding another set will increase the combined ratio, it means we are right at the global minimum, that must also achieve a $3H(m - |T'|)$ approximation bound, so we can accept that set cover.

Because $3H(m - |T'|)$ is always at most $3H(m)$, we achieve a $3H(m)$ approximation ratio as claimed. ■

Partial red-blue set cover with a must-cover element and ratio objective

In this section, we introduce the *partial red-blue set cover problem with a must-cover element*, a natural variant of red-blue set cover. We will show how our algorithm for our weighted partial set cover problem can be used to adapt the algorithms for the previous variants of red-blue set cover to solve this new variant. Subsequently, we will present our guarantees for finding reference classes.

Definition 11 *Consider a finite universe U comprised of two disjoint sets, of* red *elements R and* blue *elements B. We let β denote the number of blue elements. We suppose that we are given a collection $\mathcal{S}$ of d sets $S_1, \ldots S_d$ that are subsets of U.*

For any sub-collection $\mathcal{S}' \subseteq \mathcal{S}$, let $U(\mathcal{S}')$ denote $\bigcup_{S_i \in \mathcal{S}'} S_i$, $B(\mathcal{S}')$ denote $U(\mathcal{S}') \cap B$ and $R(\mathcal{S}')$ denote $U(\mathcal{S}') \cap R$. The goal of the partial red-blue set cover task with a must-cover element and ratio objective *is to choose a $\mathcal{S}' \subseteq \mathcal{S}$ that covers a special element x^* in addition to at least μ fraction of all the elements of B while minimizing $|R(\mathcal{S}')|/|B(\mathcal{S}')|$, i.e., the number of red elements in $\mathcal{S}'$ relative to the number of blue elements.*

As described in the paper, we modify Zhang et al's Algorithm 3, LOW_DEG, to use Algorithm 1 instead of the standard greedy algorithm, and to fit the new approximation ratio. Specifically, we set $Y = \sqrt{\frac{d}{H(\beta)}}$ in this case, yielding Algorithm 2.

Following essentially the same argument as Lemma 2 of Zhang et al., we derive the following lemma. The only difference is we used the new approximation ratio of $3H(\beta)$ instead of $3H(\mu\beta)$.

Lemma 12 (Zhang et al. Lemma 2) *In step 2, the partial set cover algorithm yields an approximation ratio of $\Delta(\mathcal{S}) \cdot 3H(\beta)$ where $\Delta(\mathcal{S}) = \max_{r_i \in R} |S \in \mathcal{S} : r_i \in S|$ is the* "maximum degree" *of red elements (to sets in $\mathcal{S}$).*

Zhang et al's Algorithm 4 (Low Deg Partial 2), which simply searches for the best value of X, remains the same. Therefore, we propose Theorem 13 for our final guarantee for the algorithm on our special partial red-blue set cover problem.

Input: finite set $\mathcal{S} = \{S_1, ..., S_d\}$, $\mu \in (0, 1]$, integer X, must-cover point x^*
Output: μ-partial cover solution $\tilde{\mathcal{S}}_X$ and corresponding error rate $\tilde{\epsilon}$
Discard sets in $\mathcal{S}$ that contain more than X red elements: set $\mathcal{S}_X \leftarrow \{S_i \in \mathcal{S} : R(S_i) \leq X\}$.
If $\frac{|B(\mathcal{S}_X)|}{|B|} < \mu$, then return FAIL ▷ $\mathcal{S}_X$ is not feasible
Set $Y = \sqrt{\frac{d}{H(\beta)}}$
Identify the high degree red elements: R_H is the set of red elements contained in more than Y members of $\mathcal{S}_X$.
Discard elements of R_H in $\mathcal{S}_X$ to obtain $\mathcal{S}_{X,Y}$
Apply Algorithm 1 to the set cover instance obtained by setting the weight of each $T_i \in \mathcal{S}_{X,Y}$ to $R(T_i)$ (with the same μ and x^*), and obtain a solution $\tilde{\mathcal{S}}_{X,Y}$ for it.
Add the dropped red elements back to obtain the corresponding result $\tilde{\mathcal{S}}_X$.
For the set of blue elements $\tilde{B}$ and red elements $\tilde{R}$ respectively covered by $\tilde{\mathcal{S}}_X$, calculate the error rate $\tilde{\epsilon} = \frac{|\tilde{R}|}{|\tilde{B}|}$ and return it and $\tilde{\mathcal{S}}_X$.

Algorithm 2: Low Deg Partial(X)

Theorem 13 (Zhang et al. Theorem 4; Peleg Theorem 3.5) *Low Deg Partial 2 solves the partial red-blue set cover problem with must-cover element, with an approximation ratio of* $4\sqrt{d \cdot H(\beta)}$.

The proof of Theorem 13 is virtually identical to the proof of Theorem 4 of Zhang et al. (and the proof Theorem 3.5 of Peleg 2007), with $3H(\beta)$ replacing the original $3H(\mu\beta)$ approximation ratio.

We now observe that, given an appropriate number of examples, our algorithms for our variant of the partial red-blue set cover problem can be used to find reference classes.

Theorem 14 (Zhang et al. Theorem 5) *Suppose we are given* $m = \Theta(\frac{1}{\gamma^2\mu\epsilon_0}(n^k + \log\frac{1}{\delta}))$ *examples. Then our algorithm can be used to solve the reference class search task in time polynomial in* m *and* n^k *with approximation ratio* $O(\sqrt{n^k \log m}) = O(\sqrt{n^k \log \frac{n+\log 1/\delta}{\gamma\mu\epsilon_0}})$.

The proof of this theorem is almost identical except for the change of a $3H(\beta)$ approximation ratio for our algorithm. Similarly, by plugging our reference class algorithm in to the algorithm by Hainline et al. (2019), we can obtain an algorithm for reference class regression with the same guarantee as for conditional regression:

Theorem 15 *For any constant* s *and* $\gamma > 0$*, and* $m = \tilde{\Theta}\left(\frac{1}{\mu}\left(\frac{b^3}{\epsilon_0\gamma^2}(n^k + \log\frac{d}{\delta}) + \frac{b^{2p}}{\epsilon_0^{2p}}(b^2 + \ln\frac{1}{\delta})\right)\right)$ *examples, the algorithm of Hainline et al. (2019) modified to use the reference-class algorithm of Theorem 14 runs in polynomial time and solves the conditional* s*-sparse* ℓ_p *regression task with* $\alpha = O(\sqrt{n^k \log m})$.

As with Theorem 7, for $m_0 = O\left(\frac{1}{\mu}\frac{b^{2p}}{\epsilon_0^{2p}}(b^2 + \log\frac{1}{\delta})\right)$, Theorem 15 actually requires $O\left(\frac{b^3}{\gamma^2\mu\epsilon_0}(n^k + \log\frac{dm_0}{\delta\gamma}) + m_0\right)$ examples, and actually obtains an approximation ratio of $O\left(\sqrt{n^k \log\left(\frac{b}{\gamma\mu\epsilon_0}(n + \log\frac{dm_0}{\delta\gamma})\right)}\right)$.

Example application: explaining classifiers

Our reference classes can serve as high-precision model-agnostic "explanations," along the lines of the *anchors* constructed by Ribeiro et al. (2018). These are intended to help a human user better understand the behavior of a relatively opaque classifier representation such as gradient boosted trees or neural nets, by generating an explanation of what points the classifier labels similarly to a point of interest. In this application, we seek a rule that (a) predicts that a given classifier will produce a given label with high precision, (b) predicts, in particular, that a given point of interest will receive the label given by the classifier, and (c) is easy for a human user to interpret. If we use the condition that a given classifier produces a given label as the condition for the reference class, this is an instance of our reference class search problem. Specifically, recall that coverage is essentially our μ and precision is essentially the error rate conditioned on the rule being satisfied, just as in our reference classes. To satisfy the final requirement, Ribeiro et al. produce conjunctive rules, ANDs of some Boolean attributes derived from the features of examples in the domain. We produce a k-DNF representation (ORs of ANDs of k Boolean attributes). DNFs have widely been presumed to be easy to interpret (Hayes and Shah 2017; Wang et al. 2015; Hauser et al. 2010). A recent study suggests that while DNFs are not *uniquely* easy to interpret, they at least satisfy many of the necessary properties needed for interpretability, as long as the formulas are small (Booth, Muise, and Shah 2019).

We now compare the performance of our algorithms for this task to the beam-search algorithm for conjunctive conditions proposed by Ribeiro et al. In particular, one evaluation considered by Ribeiro et al. evaluates the average precision and coverage achieved when generating these explanations for points drawn from a validation set. We compared the performance of our algorithm to the method proposed by Ribeiro et al. for one dataset (Lending) and the same settings they used in this particular experiment. In particular, we use our method to find a k-DNF reference class of a data point as an explanation for that data point, with coverage that matches that of the (conjunctive) anchor rule discovered by Ribeiro et al., and we compare the precision of the two rules. In the interest of achieving a balance between expressive power, comprehensibility, and running time, we only considered 3-DNFs rules for our method. Code for the experiments can be found at https://github.com/lihengxuan-wustl/Refclass-KDNF.

We note that conjunctive rules, like our k-DNFs, can only be written in terms of Boolean conditions, and we use the same transformation of the real-valued and categorical features to Boolean attributes as used by Ribeiro et al. In particular, categorical features are represented as literals asserting that the categorical attribute takes a specific value. After this transformation, the Lending data set has 36 Boolean

Table 1: Average precision and coverage of explanations

	Precision			Coverage		
	anchor	**lime-t**	**3-dnf**	**anchor**	**lime-t**	**3-dnf**
lr	95.6	81	82.6	10.7	21.6	33.6
gbt	96.2	81	82.5	9.7	20.2	35
nn	95.6	79.6	83.5	7.6	17.3	21

features. Since it is meaningless for more than one of these binary features that belong to the same original categorical feature to appear in the same term, we modified our algorithm to only use terms that contain at most one literal for each categorical feature. An example of such a 3-DNF explanation on the Lending dataset is:
(loan history= $>$10 years and Employment= Employed and total payment= $<$10000) or (grade=A and purpose=car and term=60 months) or (fico range=800 and Loan status=Fully paid and home ownership=mortgage) $\Rightarrow$ Good Loan

Thus, using the same settings as Ribeiro et al. we split the Lending dataset into three parts: a training set with 5635 examples, and a validation set and test set of 1134 examples each. On the training set, we trained three different models: logistic regression (lr), 400 gradient boosted trees (gb) and a multilayer perceptron with two layers of 50 units each (nn). Then we utilized these models to predict each point in the validation set to generate a corresponding $c(x)$. For each data point in the validation set, we set it as the distinguished point x^* for our reference class, and ran our algorithm (Algorithm 2) using the training set to produce a 3-DNF reference class. We evaluated the precision and coverage of each 3-DNF on the (held-out) test set and averaged them to get the reported values, shown in Table 1.

Discussion

We observe that our large 3-DNF algorithm performs better than LIME (Ribeiro, Singh, and Guestrin 2016) (the baseline chosen by Ribeiro et al.) both in terms of coverage and precision. Comparing to Anchors, while achieving nearly $3\times$ coverage, the precision of 3-DNF falls behind more than 10%. Although we set the target coverage for 3-DNF to be the same as Anchors, the precision keeps improving as terms are added, yielding the results above. On the other hand, the performance of the 3-DNFs is guaranteed by our bound, but Anchors (using conjunctions) cannot have such a theoretical guarantee as discussed earlier. In any case, the representations are not strictly comparable, and conjunctions might be better suited to some distributions, and 3-DNFs to others.

A downside of our large 3-DNF algorithm (compared to Anchors and LIME) is that it is much more computationally expensive. Our single-threaded Cython implementation takes about two days to compute a reference class on this data set. But, the outer loop that searches over terms that contain the target element x^* in Algorithm 1 can be safely run in parallel. (The iterations do not depend on each other.) Similarly, the runs of Algorithm 2 for different values of X in Low Deg Partial 2 can also be safely run in parallel.[1] We thus expect that the time to compute a reference class for such a parallelized variant of the algorithm should decrease roughly linearly with the number of cores available.

If a more computationally efficient method is needed, we note that the much simpler and faster algorithm we gave for small k-DNFs may be more appropriate. In experiments with a similar, simple greedy baseline, Zhang et al. (2017) found that their analogous algorithm for abduction gave only a small improvement. Also, Hainline et al. (2019) used the actual method of Juba et al. (2018) in their experiments, and found it to be effective in practice. Thus, the small k-DNF algorithm may present a more appealing trade-off.

[1] We also can only consider X of the form $\lceil (1+\gamma)^i \rceil$ for $i = 1, 2, \ldots, \log |R|$.

Directions for future work

One undesirable property of our algorithm for finding large reference classes is that it simply enumerates the choices of sets containing x^*, and runs a greedy algorithm for each of them. If the number of sets is large, this may substantially increase the running time. It is natural to conjecture that it is possible to simply make a *greedy* choice of set covering x^* before the greedy algorithm would terminate, and then continue the same as our algorithm given this choice. This would eliminate the overhead from enumerating the sets containing x^*, but this algorithm seems to be significantly more difficult to analyze.

Furthermore, similar to the abduction task, we have no idea whether or not the approximation ratios we achieve are anywhere near optimal. The difficulty is partially that the problem has an "agnostic improper learning" formulation, which makes it very challenging to analyze.

Acknowledgements

BJ was supported by NSF award CCF-1718380. This work was partially performed while BJ was visiting the Simons Institute for the Theory of Computing. We thank our reviewers for their constructive comments.

Appendix: proof of Lemma 10

We will need the following lemmas used by Zhang et al. (2017):

Lemma 16 (Lemma 8, Zhang, Mathew, and Juba 2017) *There is an optimal μ-partial cover of U in which only the final set in any greedy ordering may contain more than $\mu|U|$ elements, and the collection of all prior sets covers fewer than $\mu|U|$ elements.*

Lemma 17 (Lemma 2, Slavík 1997) *If $\{A_1, \ldots, A_\ell\}$ is an optimal μ-partial cover of U, the greedy algorithm obtains a cover of cost at most $\sum_{s=1}^{\ell-1} w(A_s)H(|A_s|) + w(A_\ell)H(\min\{\lceil \mu|U| \rceil, |A_\ell|\})$.*

We are given

$$\frac{w(G)}{|G|} \leq 3H(m - |T'|)\frac{w(O)}{|O|} \tag{1}$$

and

$$|O| \leq 3\,|G|\,. \tag{2}$$

We want to prove

$$\frac{w(T')+w(G)}{|T'|+|G|} \leq 3H(m-|T'|)\frac{w(T')+w(O)}{|T'|+|O|}.$$

Clearing denominators, this is equivalent to

$$\begin{aligned} &w(T')\,|T'| + w(T')\,|O| + w(G)\,|T'| + w(G)\,|O| \leq \\ &\;3H(m-|T'|)w(T')\,|T'| + 3H(m-|T'|)w(T')\,|G| \\ &\;+ 3H(m-|T'|)w(O)\,|T'| + 3H(m-|T'|)w(O)\,|G|\,. \end{aligned}$$

First, $w(T')\,|T'| \leq 3H(m-|T'|)w(T')\,|T'|$, so we can reduce the above to proving

$$\begin{aligned} &w(T')\,|O| + w(G)\,|T'| + w(G)\,|O| \leq \\ &\quad 3H(m-|T'|)w(T')\,|G| + 3H(m-|T'|)w(O)\,|T'| \\ &\quad + 3H(m-|T'|)w(O)\,|G|\,. \end{aligned}$$

Clearing denominators in inequality 1, we get

$$w(G)\,|O| \leq 3H(m-|T'|)w(O)\,|G|\,. \tag{3}$$

Given the terms from inequality 3, we only need to prove

$$\begin{aligned} &w(T')\,|O| + w(G)\,|T'| \leq \\ &\quad 3H(m-|T'|)w(T')\,|G| + 3H(m-|T'|)w(O)\,|T'|\,. \end{aligned}$$

Multiplying both sides of inequality 2 by $w(T')$, we have

$$w(T')\,|O| \leq 3w(T')\,|G|$$

which can be further loosened to get

$$w(T')\,|O| \leq 3H(m-|T'|)w(T')\,|G|\,. \tag{4}$$

Also, combining Lemma 16 and Lemma 17, we get

$$w(G) \leq H(m-|T'|)w(O)$$

so we have

$$w(G)\,|T'| \leq 3H(m-|T'|)w(O)\,|T'|\,. \tag{5}$$

Combining inequalities 4 and 5 completes the proof.

References

Bacchus, F.; Grove, A. J.; Halpern, J. Y.; and Koller, D. 1996. From statistical knowledge bases to degrees of belief. *Artificial Intelligence* 87:75–143.

Biran, O., and Cotton, C. 2017. Explanation and justification in machine learning: A survey. In *IJCAI-17 Workshop on Explainable AI (XAI)*.

Booth, S.; Muise, C.; and Shah, J. 2019. Evaluating the interpretability of the knowledge compilation map: Communicating logical statements effectively. In *Proc. 28th IJCAI*, 5801–5807.

Bshouty, N. H., and Burroughs, L. 2005. Maximizing agreements with one-sided error with applications to heuristic learning. *Machine Learning* 59(1–2):99–123.

Daniely, A., and Shalev-Shwartz, S. 2016. Complexity theoretic limtations on learning DNF's. In *Proc. 29th COLT*, volume 49 of *JMLR Workshops and Conference Proceedings*. 1–16.

Durgin, A., and Juba, B. 2019. Hardness of improper one-sided learning of conjunctions for all uniformly falsifiable CSPs. In *Proc. 30th ALT*, volume 98 of *PMLR*. 369–382.

Hainline, J.; Juba, B.; Le, H. S.; and Woodruff, D. P. 2019. Conditional sparse ℓ_p-norm regression with optimal probability. In *Proc. 22nd AISTATS*, volume 89 of *PMLR*, 369–382.

Hauser, J. R.; Toubia, O.; Evgeniou, T.; Befurt, R.; and Dzyabura, D. 2010. Disjunctions of conjunctions, cognitive simplicity, and consideration sets. *Journal of Marketing Research* 47(3):485–496.

Haussler, D. 1988. Quantifying inductive bias: AI learning algorithms and Valiant's learning framework. *Artificial Intelligence* 36:177–221.

Hayes, B., and Shah, J. A. 2017. Improving robot controller transparency through autonomous policy explanation. In *Proc. 12th HRI*, 303–312.

Juba, B.; Li, Z.; and Miller, E. 2018. Learning abduction under partial observability. In *Proc. 32nd AAAI*, 1888–1896.

Juba, B. 2016. Learning abductive reasoning using random examples. In *Proc. 30th AAAI*, 999–1007.

Juba, B. 2017. Conditional sparse linear regression. In *Proc. 8th ITCS*, volume 67 of *LIPIcs-Leibniz International Proceedings in Informatics*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.

Kearns, M. J. 1990. *The computational complexity of machine learning*. MIT press.

Kyburg, H. E. 1974. *The Logical Foundations of Statistical Inference*. Dordrecht: Reidel.

Peleg, D. 2007. Approximation algorithms for the label-covermax and red-blue set cover problems. *J. Discrete Algorithms* 5:55–64.

Pollock, J. L. 1990. *Nomic Probabilities and the Foundations of Induction*. Oxford: Oxford University Press.

Qi, D.; Arfin, J.; Zhang, M.; Mathew, T.; Pless, R.; and Juba, B. 2018. Anomaly explanation using meta-data. In *Proc. WACV'18*, 1916–1924.

Reichenbach, H. 1949. *Theory of Probability*. Berkeley, CA: University of California Press.

Ribeiro, M. T.; Singh, S.; and Guestrin, C. 2016. Why should I trust you?: Explaining the predictions of any classifier. In *Proc 22nd KDD*, 1135–1144.

Ribeiro, M. T.; Singh, S.; and Guestrin, C. 2018. Anchors: High-precision model-agnostic explanations. In *Proc. 32nd AAAI*, 1527–1535.

Rosenfeld, A.; Graham, D. G.; Hamoudi, R.; Butawan, R.; Eneh, V.; Kahn, S.; Miah, H.; Niranjan, M.; and Lovat, L. B. 2015. MIAT: A novel attribute selection approach to better predict upper gastrointestinal cancer. In *Proc. DSAA*, 1–7.

Roth, D. 1995. Learning to reason: the non-monotonic case. In *Proc. 14th IJCAI*, volume 2, 1178–1184.

Slavík, P. 1997. Improved performance of the greedy cover algorithm for partial cover. *Information Processing Letters* 64(5):251–254.

Valiant, L. G. 1994. *Circuits of the Mind*. Oxford: Oxford University Press.

Valiant, L. G. 2000. A neuroidal architecture for cognitive computation. *J. ACM* 47(5):854–882.

Wang, T.; Rudin, C.; Doshi-Velez, F.; Liu, Y.; Klampfl, E.; and MacNeille, P. 2015. Or's of and's for interpretable classification, with application to context-aware recommender systems. *arXiv preprint arXiv:1504.07614*.

Zhang, M.; Mathew, T.; and Juba, B. 2017. An improved algorithm for learning to perform exception-tolerant abduction. In *Proc. 31st AAAI*, 1257–1265.