SIIM-TR-A-**03-12**

SIIM Technical Report

Robust Inference of Relevant Attributes

by

JAN ARPE AND RÜDIGER REISCHUK

Schriftenreihe der Institute für

Informatik/Mathematik

Serie A

July 3, 2003

Universität zu Lübeck
Technisch-Naturwissenschaftliche Fakultät

Email: arpe@tcs.uni-luebeck.de

Phone: +49-451-7030-412
Fax: +49-451-7030-438

Robust Inference of Relevant Attributes

Jan Arpe[*] **Rüdiger Reischuk**

Institut für Theoretische Informatik, Universität zu Lübeck,
Wallstr. 40, 23560 Lübeck, Germany

arpe/reischuk@tcs.uni-luebeck.de

July 3, 2003

Abstract

Given n Boolean input variables representing a set of attritubes, we consider Boolean functions f (i.e., binary classifications of tuples) that actually depend only on a small but unknown subset of these variables/attributes, in the following called *relevant*. The goal is to determine these relevant attributes given a sequence of examples - input vectors X with their classification $f(X)$. We analyze two simple greedy strategies and prove that they are able to achieve this goal for various kinds of Boolean functions and various input distributions according to which the examples are drawn at random. This generalizes results obtained by Akutsu, Miyano, and Kuhara for the uniform distribution. The analysis also provides explicit upper bounds on the number of necessary examples. They depend on the distribution and combinatorial properties of the function to be inferred.

Our second contribution is an extension of these results to a situation where the examples contain errors, that means a certain number of input bits x_i may be wrong. This is a typical situation, for example in medical research, where not all attributes can be measured reliably. We show that even in such an error-prone situation reliable inference of the relevant attributes can be performed by proving that our greedy strategies are robust even against a linear number of errors.

[*]Supported by DFG research grant Re 672/3.

1 Introduction

In many data mining applications, one is faced with the situation that a binary classification of elements with a large number of attributes only depends on a small subset of these attributes. A central task is then to infer these relevant attributes from a given input sample consisting of a series of examples $X(k) = (x_1(k), \ldots, x_n(k))$ with classification $y(k)$ for $k = 1, 2, \ldots, m$, i.e., one wants to find a set of variables $x_{i_1}, \ldots, x_{i_d}$ such that the sample can be *explained* by a function $f : \{0,1\}^n \to \{0,1\}$ that depends only on these d variables. A function f is said to explain the sample, if $f(x_1(k), \ldots, x_n(k)) = y(k)$ for all k. Moreover, since real data usually contain noise, it is of particular interest to design algorithms that in some sense behave "robustly" with respect to input disturbances.

When inferring relevant attributes, there are two natural questions that can be asked:

1. Given a fixed sample of an unknown concept, what is the minimum number of variables that explain the sample?
2. How many examples does one need to generate in order to find out the actual relevant attributes?

The first question gives rise to an optimization problem introduced in Section 3, whereas the second one can be considered as an algorithmic learning problem. In both cases, however, the key task is to infer relevant variables from a sample (of a given fixed size in the first case, and of variable size in the second case). Thus our goal is to design efficient algorithms that find a small set of variables explaining the input sample.

Akutsu and Bao [3] proposed a greedy algorithm based on a well-known greedy strategy for the SET COVER problem (see Johnson [10]). Akutsu, Miyano, and Kuhara [4] describe an efficient implementation of this approach and give an average case analysis of the algorithm for two special types of functions, namely AND and OR of arbitrary literals, under the uniform distribution of input examples. In Section 4, we simplify the greedy strategy. We call this strategy GREEDY RANKING and show that its performance is similar to the one obtained in [4].

In Section 5, the average case analysis of [4] is generalized in two respects: to a broader class of functions and to weaker assumptions on the input distributions. It turns out that a modification of our approach, namely taking the *smallest* sets of the ranking, may also be useful for some classes of functions and input distributions. We call this strategy MODEST RANKING, since its "modest" behavior of first selecting the smallest sets is in contrast to the greedy strategy of taking the largest sets. We apply these very general results to some typical input distributions and some specific functions of major interest (e.g., monomials, clauses, and threshold functions) in Section 6.

After these investigations we turn to the "real case" of samples that contain partial errors. In Section 7, we assume that some bits of an input vector $x_1, \ldots, x_n$ may be wrong. More precisely, for each attribute X_i there is a certain (generally unknown) error probability δ_i that the entry x_i in an input vector is flipped. Of course, we have to assume that errors of different attributes occur independently. It will be shown that one can tolerate a certain (small) constant fraction of such erroneous bits and still infer the relevant attributes successfully with high probability. The only condition required is that all $\delta_i \leq \delta$, where $\delta > 0$ is some small fraction that only depends on combinatorial properties of the function f to be inferred and the probability distribution according to which the samples are generated. This is done by proving that our greedy strategies are quite fault-tolerant in such a circumstance, thus they can infer relevant attributes *robustly*. In addition, the number of examples that are necessary in case of disturbed inputs only grows by a constant factor.

Finally, in Section 8, we consider the parity function for which ranking strategies do not work. A different approach is presented to infer relevant variables in this case as well.

Inferring relevant attributes is related to finding *association rules* (which are also called *functional dependencies* or *functional relations*) – a well-studied problem (e.g., see [2, 11]). Note that in the variant considered in this paper, the target attribute Y is fixed as in [3, 4]. The goal of efficiently inferring concepts with many irrelevant attributes (so-called *attribute-efficient learning*) has attracted much attention in the past (e.g., see Blum et al. [7] and Valiant [12]). However, most authors consider the *mistake-bounded model* in which an algorithm gets one example at a time and computes a hypothesis after each new example. The algorithm has to correct its last hypothesis if it disagrees with the new example. In this setting, one tries to minimize the number of examples for which the current hypothesis turns out to be wrong. However, it may take arbitrarily long until the eventually correct hypothesis is reached. In this paper, our goal is much harder: From a randomly selected sample of small size, we want to compute only a single hypothesis that should be correct with high probability — a variant of the finite learning model.

Our noise model is similar to the *product random attribute noise* introduced by Goldman and Sloan [9] which has been applied to the PAC learning model.

2 Preliminaries

A *concept* is a Boolean function $f : \{0,1\}^n \to \{0,1\}$, a *concept class* is a set of concepts. For a subset $I = \{i_1, \ldots, i_d\}$ of $\{1, \ldots, n\}$, we define the projection

$$\Pi_I : \{0,1\}^n \to \{0,1\}^d$$

by $\Pi_I(x_1, \ldots, x_n) = (x_{i_1}, \ldots, x_{i_d})$. A concept $f : \{0,1\}^n \to \{0,1\}$ *depends* on variable x_i, if the two $(n-1)$-ary subfunctions $f_{x_i=0}$ and $f_{x_i=1}$ with variable x_i fixed to 0 and 1 respectively are not identical. If f depends on x_i, then attribute x_i is called *relevant* for f, otherwise *irrelevant*. We denote the set of relevant (resp. irrelevant) variables/attributes by $V^+(f)$ (resp. $V^-(f)$). If f is clear from the context, we just write V^+ and V^-. Thus for $I = \{i_1, \ldots, i_d\}$, f depends only on attributes with indices in I iff f can be written as $f = \tilde{f} \circ \Pi_I$ for some $\tilde{f} : \{0,1\}^d \to \{0,1\}$. Let us call the unique function $\tilde{f}$ corresponding to the (minimal) number d of relevant attributes the *base function* of f.

An *example* is a vector $(x_1, \ldots, x_n; y) \in \{0,1\}^{n+1}$. It is an *example for* f, if $y = f(x_1, \ldots, x_n)$. The values of $x_1, \ldots, x_n$ are called *variable* or *attribute assignments*, whereas the value for y is called a $label$. A sequence $(x_1(k), \ldots, x_n(k); y(k))_{k=1,\ldots,m}$ of examples for f is called a *sample for f of size m*, and f is said to *explain* the sample. A *sample T* is a sequence of examples such that there exists some f that explains the sample. If f depends only on variables from the set $\{x_{i_1}, \ldots, x_{i_d}\}$, then we also say that these variables *explain* T. Given a sample T of an unknown function f, our goal is to infer a set of explaining attributes for T. Note that a sample may not contain enough information in order to correctly infer the relevant atrributes of f. Thus, a worst case scenario, in which an adversary selects the examples, does not make sense in our model of finite learning. Instead, one has to assume that examples are generated by a random process, and the best one can hope for is a tight average case analysis.

A sample is stored in a matrix each line of which represents one example. We denote such matrices by

$$T = (X; y) = \begin{pmatrix} x_1(1) & \ldots & x_n(1) & | & y(1)) \\ \vdots & \ddots & \vdots & & \vdots \\ x_1(m) & \ldots & x_n(m) & | & y(m) \end{pmatrix} \in \{0,1\}^{m \times (n+1)} ,$$

where X is the submatrix consisting of the variable assignments in the examples, and y is the column vector containing the labels of the examples.

A sample T may contain a certain combination of attributes several times. Then, of course, it is necessary that for $k \neq l$ the following implication holds:

$$X(k) = X(l) \quad \Longrightarrow \quad y(k) = y(l) . \tag{1}$$

This will always be assumed in the error-free case. Indeed, if (1) does not hold for some k and l, then by definition, T is not a sample. In the noisy case, however, it may well be that different combinations of attributes yield different labels, but due to false measurements of the attributes, the values for $x_1, \ldots, x_n$ all look the same.

We assume that the examples of a sample T are drawn according to a fixed probability distribution $p : \{0,1\}^n \to [0,1]$, and we say that T is *generated according to p*. If f depends on the variables $x_{i_1}, \ldots, x_{i_d}$, then we denote by $\hat{p} : \{0,1\}^d \to [0,1]$ the induced probability distribution on the relevant attributes, i.e.,

$$\hat{p}(x_{i_1}, \ldots, x_{i_d}) = \sum_{x_i \in \{0,1\} \text{ for } x_i \in V^-} p(x_1, \ldots, x_n) .$$

Definition 1 Let $(X; y) \in \{0,1\}^{m \times (n+1)}$ be a sample. The corresponding *functional relations graph* is a bipartite labeled graph defined as follows. The vertices are $\{1, \ldots, m\}$, the edges are

$$S = \{\{k, l\} \mid y(k) \neq y(l)\} .$$

Each edge $\{k, l\}$ is labeled by the set of variables x_i such that $x_i(k) \neq x_i(l)$. The set of edges with a label containing variable x_i is denoted by

$$S_i = \{\{k, l\} \in S \mid x_i(k) \neq x_i(l)\} .$$

Proposition 1 *Let $T = (X; y) \in \{0,1\}^{m\times(n+1)}$ be a sample and $I = \{i_1, \dots, i_d\} \subseteq \{1, \dots, n\}$. Then the following statements are equivalent:*

(a) $x_{i_1}, \dots, x_{i_d}$ *explain* T.

(b) For each pair $k, l \in \{1, \dots, m\}$ *such that* $y(k) \neq y(l)$ *there exists* $r \in \{1, \dots, d\}$ *such that* $x_{i_r}(k) \neq x_{i_r}(l)$.

(c) $S = S_{i_1} \cup \dots \cup S_{i_d}$.

Proof: "(a) ⇒ (b):" Let $f : \{0,1\}^n \to \{0,1\}$ be a concept depending on $x_{i_1}, \dots, x_{i_d}$ and explaining T. Let $k, l \in \{1, \dots, m\}$ with $y(k) \neq y(l)$. Then

$$\begin{aligned}\tilde{f}(x_{i_1}(k), \dots, x_{i_d}(k)) &= f(x_1(k), \dots, x_n(k)) \\ &\neq f(x_1(l), \dots, x_n(l)) \\ &= \tilde{f}(x_{i_1}(l), \dots, x_{i_d}(l)) ,\end{aligned}$$

so there has to be at least one $r \in \{1, \dots, d\}$ such that $x_{i_r}(k) \neq x_{i_r}(l)$.
"(b) ⇒ (c)": This implication holds by definition of the sets S_i.
"(c) ⇒ (a):" Suppose that $S = S_{i_1} \cup \dots \cup S_{i_d}$. We define a concept $f : \{0,1\} \to \{0,1\}$ as follows. For $z = (z_1, \dots, z_n)$ such that there exists $k \in \{1, \dots, m\}$ with $(z_{i_1}, \dots, z_{i_d}) = (x_{i_1}(k), \dots, x_{i_d}(k))$, we define $f(z) = y(k)$. Otherwise, we can define $f(z)$ arbitrarily, say $f(z) = 0$. We have to show that this is well-defined: Assume that there are $k, l \in \{1, \dots, m\}$ such that $z = (x_{i_1}(k), \dots, x_{i_d}(k)) = (x_{i_1}(l), \dots, x_{i_d}(l))$ but $y(k) \neq y(l)$. Then $\{k, l\} \in S_{i_r}$ for some $r \in \{1, \dots, d\}$, and thus $x_{i_r}(k) \neq x_{i_r}(l)$ which is a contradiction. Hence f is well-defined. By definition of f, T is a sample for f, and f only depends on variables in $\{x_{i_1}, \dots, x_{i_d}\}$, so these variables explain T. □

3 Approximability

Consider the following optimization problem:

INFERENCE OF RELEVANT ATTRIBUTES (INFRA)	
Instance:	sample $T = (X; y) = (x_1(k), \dots, x_n(k); y(k))_{k=1,\dots,m} \in \{0,1\}^{m\times(n+1)}$
Solution:	a function $f : \{0,1\}^n \to \{0,1\}$ such that T is a sample for f (i.e., $y(k) = f(x_1(k), \dots, x_n(k))$ for all $k \in \{1, \dots, m\}$)
Measure:	$\lvert V^+(f)\rvert$
Goal:	minimize $\lvert V^+(f)\rvert$

Note that in order to find a small set of explaining attributes for an INFRA instance, we do not have to explicitly define a corresponding concept f, but it is enough to find a set of attributes $x_{i_1}, \dots, x_{i_d}$ such that for $k, l \in \{1, \dots, m\}$ with $y(k) \neq y(l)$ there exists $r \in \{1, \dots, d\}$ with $x_{i_r}(k) \neq x_{i_r}(l)$ by Proposition 1.

In order to obtain results on the approximability of INFRA, we introduce the well-studied SET COVER problem:

SET COVER	
Instance:	universe $U = \{1,\ldots,m\}$, subsets $S_1,\ldots,S_n \subseteq U$ such that $S_1 \cup \ldots \cup S_n = U$
Solution:	a set of indices $I = \{i_1,\ldots,i_d\}$ such that $S_{i_1} \cup \ldots \cup S_{i_d} = U$
Measure:	$\lvert I\rvert$
Goal:	minimize $\lvert I\rvert$

Note that Proposition 1 yields a reduction from INFRA to SET COVER. Based on this fact, Akutsu and Bao [3] have proved the following theorem.

Theorem 1 ([3]) INFRA *can be approximated in polynomial time within a factor of* $2\ln m + 1$.

The next claim is a slightly stronger version of Theorem 8 in [3], since we consider the special case of INFRA for *Boolean* functions.

Proposition 2 SET COVER *is reducible to* INFRA *via a polynomial time computable approximation factor preserving reduction.*

Proof: Let $U = \{1,\ldots,m\}$ and $S_1,\ldots,S_n \subseteq U$ with $S_1 \cup \ldots \cup S_n = U$ be a SET COVER instance. We will describe the INFRA instance T with n variables and $m+1$ examples to which the given instance is mapped. For $i \in \{1,\ldots n\}$ and $k \in \{1,\ldots,m\}$, define $x_i(k) = 1$, if $k \in S_i$, and $x_i(k) = 0$ otherwise. Finally, let $x_i(m+1) = 0$ for all $i \in \{1,\ldots,n\}$, $y(k) = 0$ for all $k = 1,\ldots,m$, and $y(m+1) = 1$. Now it is easy to see that for an arbitrary set of indices $\{i_1,\ldots,i_d\} \subseteq \{1,\ldots,n\}$, it holds that $S_{i_1} \cup \ldots \cup S_{i_d} = U$ iff $x_{i_1},\ldots,x_{i_d}$ explain T:
"$\Rightarrow$": Let $k,l \in \{1,\ldots,m+1\}$ with $y(k) \neq y(l)$. Without loss of generality, we may assume that $y(k) = 0$ and $y(l) = 1$. Thus $l = m+1$ and $k \in \{1,\ldots,m\}$, and so $k \in S_{i_t}$ for some $t \in \{1,\ldots,d\}$. This means that $x_{i_t}(k) = 1$ whereas $x_{i_t}(l) = x_{i_t}(m+1) = 0$. By Proposition 1, $x_{i_1},\ldots,x_{i_d}$ explain T.
"$\Leftarrow$": Let $k \in U = \{1,\ldots,m\}$. As $k \neq m+1$, we have $y(k) = 0 \neq 1 = y(m+1)$. Hence there exists $t \in \{1,\ldots,d\}$ such that $x_{i_t}(k) \neq x_{i_t}(m+1) = 0$, implying $x_{i_t} = 1$ and thus $k \in S_{i_t}$. Consequently, U is covered by $S_{i_1},\ldots,S_{i_d}$. □

Applying an inapproximability result of Feige [8], we obtain the following lower bound.

Theorem 2 *For any* $\varepsilon > 0$, INFRA *cannot be approximated within a factor of* $(1-\varepsilon)\ln m$ *unless* $\mathrm{NP} \subseteq \mathrm{DTIME}(n^{O(\log\log n)})$.

Therefore, when faced with the INFRA problem, the best one can hope for are efficient approximation algorithms with a nonconstant approximation ratio or fast algorithms providing correct results for "most" inputs. In the rest of this paper, we will investigate the latter challenge.

```
Algorithm Greedy

input (x_1(k), ..., x_n(k); y(k))_{k=1,...,m}
V := {x_1, x_2, ..., x_n}
S := {{k,l} | y(k) ≠ y(l)}
while S ≠ ∅ do
    for i = 1 to n do
        S_i := {{k,l} ∈ S | x_i(k) ≠ x_i(l)}
    find an x_i ∈ V with maximum |S_i|
    output x_i
    S := S \ S_i
    V := V \ {x_i}
```

Figure 1: Algorithm GREEDY

```
Procedure Rank Sets

for i = 1 to n do
    S_i := {{k,l} ∈ S | x_i(k) ≠ x_i(l)}
compute π : {1,...,n} → {1,...,n} such
that |S_π(1)| ≥ |S_π(2)| ≥ ... ≥ |S_π(n)|
```

Figure 2: Procedure `Rank Sets`

4 From Greedy to Ranking

Let us start with the algorithm GREEDY discussed in [4] (see Figure 1). It makes use of the reduction from INFRA to SET COVER given by Proposition 1 and applies a well-known greedy approach to the SET COVER instance obtained. Johnson [10] first analyzed this approach for SET COVER.

We apply some modifications of this algorithm and analyze their effects. The strategy is based on a ranking of the sets $S_1, \ldots, S_n$ by their cardinalities which is done by the procedure `Rank Sets`. The results may be worse in some cases since the new greedy approach is based on a single static ranking. However, we will show that the ranking still yields properties similar to GREEDY. In the next section, these new strategies will be analyzed in more detail and will be applied to a broader class of functions and distributions compared to [4]. Moreover, in Section 7 we will show that the ranking approach performs quite robustly when confronted with input disturbances.

GREEDY RANKING (see Figure 3) outputs the variables x_i with maximum $|S_i|$ until these S_i's cover the whole edge set S. The major difference between GREEDY and GREEDY RANKING is that the latter algorithm does not recompute the sets S_i in each step. Given a concept f, the GREEDY RANKING algorithm works correctly, if with high probability, the sets S_i for $x_i \in V^+$ are larger than the sets S_i for $x_i \in V^-$. On the other hand, if the converse is the case, i.e., if the sets S_i for relevant variables are likely to be *smaller* than

```
Algorithm Greedy Ranking

input (x_1(k), ..., x_n(k); y(k))_{k=1,...,m}
Rank Sets
S := {{k,l} | y(k) ≠ y(l)}
i := 1
while S ≠ ∅ do
    output x_{π(i)}
    S := S \ S_{π(i)}
    i := i + 1
```

Figure 3: Algorithm GREEDY RANKING

```
Algorithm Modest Ranking

input (x_1(k), ..., x_n(k); y(k))_{k=1,...,m}
Rank Sets
S := {{k,l} | y(k) ≠ y(l)}
i := n
while S ≠ ∅ do
    output x_{π(i)}
    S := S \ S_{π(i)}
    i := i - 1
```

Figure 4: Algorithm MODEST RANKING

the sets S_i for the irrelevant variables, we should make use of an algorithm that outputs the variables corresponding to the *smallest* sets S_i. Instead of being greedy, this algorithm rather behaves modestly, so we call it MODEST RANKING (see Figure 4).

Note that, given a sample, all three algorithms terminate after a finite number of steps since by property (1), each pair $\{k, l\} \in S$ belongs to some S_i. Clearly, all algorithms presented here compute a cover of S. Thus by the reduction given in Proposition 1, the algorithms work correctly for the optimization problem, i.e., they output sets of variables that explain the input sample.

In order to see that in general none of the algorithms is superior to the others in terms of finding small admissible sets, we present three samples (see Tables 1-3).

k	x_1	x_2	x_3	y
1	1	0	0	1
2	1	1	0	1
3	1	1	0	1
4	0	0	1	1
5	0	0	0	0

Table 1: A sample for which GREEDY performs best

k	x_1	x_2	x_3	x_4	y
1	0	0	1	0	1
2	1	0	0	0	1
3	1	1	0	0	1
4	1	1	0	0	1
5	1	1	0	0	1
6	1	0	1	0	1
7	1	0	1	0	1
8	1	0	0	0	1
9	0	1	0	0	1
10	0	1	0	1	1
11	0	1	0	1	1
12	0	0	1	1	1
13	0	0	1	1	1
14	0	0	0	0	0

Table 2: A sample for which GREEDY RANKING performs best

k	x_1	x_2	x_3	x_4	y
1	1	1	1	1	1
2	1	1	1	1	1
3	1	1	1	1	1
4	1	1	0	0	1
5	1	1	0	0	1
6	1	1	0	0	1
7	0	0	0	0	0

Table 3: A sample for which MODEST RANKING performs best

The corresponding functional relations graphs are drawn in Figures 5-7. Figure 8 shows the inclusion relations of the edge sets S_i in the functional relations graphs. The dots represent the lines (i.e., the single examples) of the given samples, ordered top-down and from left to right.

Note that the graphics in Figure 8 may also be interpreted as SET COVER instances. Indeed, using the reduction in the proof of Proposition 2, one recovers the INFRA samples from the SET COVER instances.

Table 4 shows which variables are selected by the algorithms in each case.

	Table 1	Table 2	Table 3
GREEDY	x_1, x_3	x_1, x_4, x_2, x_3	x_1, x_2, x_3, x_4
GREEDY RANKING	x_1, x_2, x_3	x_1, x_2, x_3	x_1, x_2, x_3, x_4
MODEST RANKING	x_3, x_2, x_1	x_4, x_3, x_2, x_1	x_4, x_3, x_2

Table 4: Output of algorithms for example instances

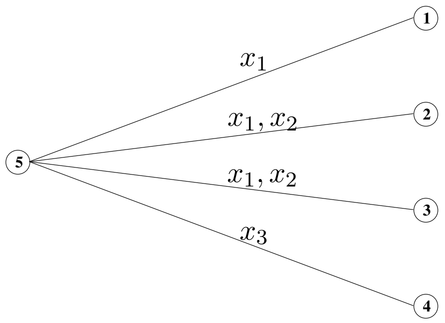

Figure 5: Functional relations graph corresponding to Table 1

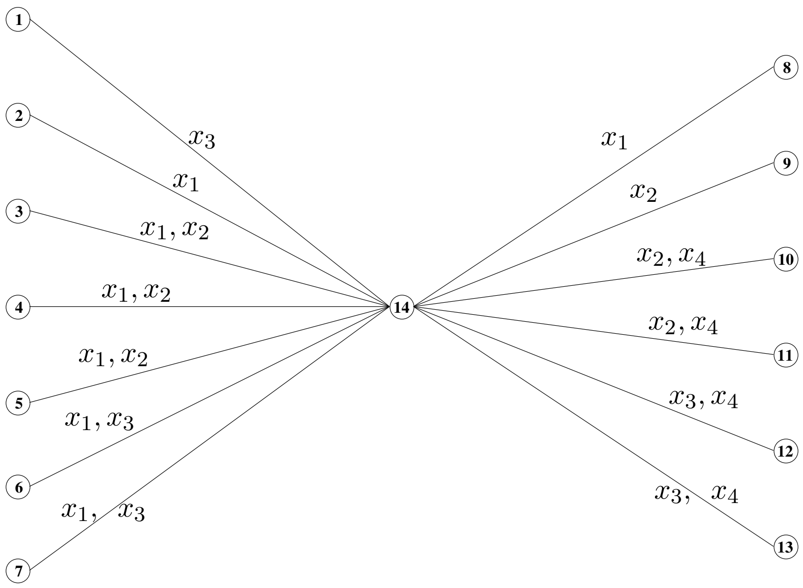

Figure 6: Functional relations graph corresponding to Table 2

Let us now turn to the case that the input samples arise from some concept f. In this case, it may happen that a sample can be explained by a proper subset of the relevant variables for f, because the number of examples does not suffice to recover f. In case the number d of relevant variables is a priori known, we can overcome this problem by giving d as additional input to the algorithms and output the d variables with the largest (resp., smallest) sets S_i. This is done by the TARGET RANKING and the MODEST TARGET RANKING algorithms which are presented in Figure 9 and Figure 10.

Moreover, if we only have an a priori upper bound d' on the number of relevant variables, then the target ranking algorithms will output a set of d' variables such that the d relevant ones are most likely among them.

Definition 2 Given a concept f with relevant variables $x_{i_1}, \ldots, x_{i_d}$ and a sample T for f, we say that an algorithm *succeeds in a step*, if the output generated in that step is a relevant variable. The algorithm is said to be *correct* (on input T, with respect to f), if it is successful in all steps it makes (i.e., it outputs a subset of all relevant variables). It is *complete* (on input T, with respect to f), if it finds *all* relevant variables (i.e., it outputs a superset of the

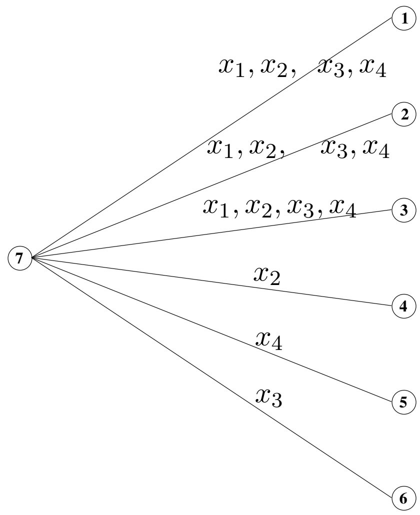

Figure 7: Functional relations graph corresponding to Table 3

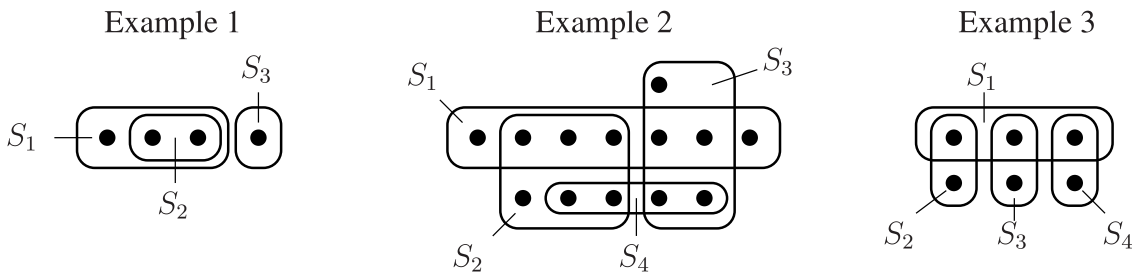

Figure 8: Examples of SET COVER instances

relevant variables). Finally, an algorithm is said to be *successful*, if it is both correct and complete (i.e., it outputs exactly all relevant variables).

The following properties are easy to show.

Lemma 1 *Let f be a concept depending on exactly d variables, and let T be a sample for f.*

(a) If TARGET RANKING *(resp.,* MODEST TARGET RANKING*) is correct on input (T, d), then it is successful, too.*

Algorithm Target Ranking

input $(x_1(k), \ldots, x_n(k);\ y(k))_{k=1,\ldots,m},\ d$
Rank Sets
$S := \{\{k, l\} \mid y(k) \neq y(l)\}$
for $i = 1$ to d do
    output $x_{\pi(i)}$

Figure 9: Algorithm TARGET RANKING

```
Algorithm Modest Target Ranking
input (x_1(k),...,x_n(k); y(k))_{k=1,...,m}, d
Rank Sets
S := {{k,l} | y(k) ≠ y(l)}
for i = 1 to d do
    output x_{π(n-i+1)}
```

Figure 10: Algorithm MODEST TARGET RANKING

(b) If TARGET RANKING *(resp.,* MODEST TARGET RANKING*) is complete on input* (T, d)*, then it is also successful.*

(c) If TARGET RANKING *(resp.,* MODEST TARGET RANKING*) is successful on input* (T, d)*, then* GREEDY RANKING *(resp.,* MODEST RANKING*) is correct on input* T*.*

Proof: (a) and (b) are obvious.

If TARGET RANKING is successful on input (T, d), then the largest d sets of the ranking exactly correspond to the relevant variables. Since S is completely covered by these sets in this case, GREEDY RANKING will terminate after at most d steps. Thus, all variables output by GREEDY RANKING are relevant, i.e., GREEDY RANKING is correct.

A similar reasoning shows that MODEST RANKING is correct on input T, provided that MODEST TARGET RANKING is successful on input (T, d). □

In order to uniquely recognize the relevance of some variable x_{i_r}, there has to be an edge in the functional relations graph whose only *relevant label* is x_{i_r}. For this reason we call $a = (a_1, \ldots, a_{d-1})$ a *witness* for the relevance of x_{i_r}, provided that $\tilde{f}(a_r^0) \neq \tilde{f}(a_r^1)$, where $\tilde{f}$ is the base function of f as defined in Section 2, and $a_r^b = (a_1, \ldots, a_{r-1}, b, a_r, \ldots, a_{d-1})$ for $b \in \{0, 1\}$. Thus, independent of the used learning algorithm, a necessary (but not sufficient) condition to infer the relevance of x_{i_r} is the occurrence of at least one pair a_r^0, a_r^1 in the input sample. By the birthday paradox, already for the completely uniform distribution (each $x \in \{0, 1\}$ has the same probability), roughly $\sqrt{2^{d-1}}$ examples are necessary to guarantee such an occurrence. This shows that in order for *any* algorithm to be correct and complete, $\Omega(2^{d/2})$ examples have to be provided due to information theoretic reasons.

Our main focus in the next two sections will be the investigation of concept classes $\mathcal{C}$ and probability distributions p such that the ranking algorithms succeed with high probability for input samples (for $f \in \mathcal{C}$) generated according to p. Note that we do not adjust the algorithms for specific conept classes, so we are interested in the performance of a fixed single algorithm for a variety of concepts and distributions.

5 Probabilistic Analysis of the Ranking Strategies

5.1 Chernoff Bounds

We write $X \sim B(n,p)$ in order to indicate that X is a random variable that is binomially distributed with parameters n and p. The following fact can, for example, be found in Alon, Spencer, and Erdős [1].

Fact 1 Let $X \sim B(n,p)$, and let $\mu = pn$ be the expectation of X. For all ε with $0 \le \varepsilon < 1$,

$$\Pr(X \le (1-\varepsilon)\mu) < e^{-\frac{1}{2}\varepsilon^2\mu} \quad \text{and} \quad \Pr(X \ge (1+\varepsilon)\mu) < e^{-\frac{1}{3}\varepsilon^2\mu}.$$

Fact 2 Let $X \sim B(n,p)$.

(a) For all $0 \le \varepsilon \le \sqrt{p}$,

$$\Pr(|X - pn| \ge \varepsilon\sqrt{p}n) < 2e^{-\frac{1}{3}\varepsilon^2 n}\ , \text{ and} \tag{2}$$

(b) for all $\varepsilon > \sqrt{p}$,

$$\Pr(|X - pn| \ge 2\varepsilon^2 n) < e^{-\frac{1}{3}\varepsilon^2 n}\ . \tag{3}$$

Proof: (a) is clear by substituting ε by $\frac{\varepsilon}{\sqrt{p}}$ in Fact 1. In order to show (b), let $X' \sim B(n,\varepsilon^2)$. Since $2\varepsilon^2 > \varepsilon^2 > p$, we have

$$\begin{aligned}\Pr(|X - pn| \ge 2\varepsilon^2 n) &= \Pr(X \ge (p+2\varepsilon^2)n) \quad \le \quad \Pr(X' \ge (p+2\varepsilon^2)n) \\ &\le \Pr(X' \ge 2\varepsilon^2 n) \quad \le \quad e^{-\frac{1}{3}\varepsilon^2 n}\ ,\end{aligned}$$

where the last inequality is again due to Fact 1. □

5.2 Ranking Strategies

Let $f : \{0,1\}^n \to \{0,1\}$ be a concept with relevant variables $x_{i_1}, \ldots, x_{i_d}$, and p a probability distribution. For $x, y \in \{0,1\}$, we denote by "$x_i = x$" the set of examples with $x_i = x$, and by "$f = y$" the set of examples with $f(x_1, \ldots, x_n) = y$.

For $i \in \{1, \ldots, n\}$ and $x, y \in \{0,1\}$ define the probability $\alpha_i^{(x,y)}$ that a randomly drawn example $(x_1, \ldots, x_n; f(x_{i_1}, \ldots, x_{i_d}))$ has $x_i = x$ and $f(x_{i_1}, \ldots, x_{i_d}) = y$, i.e.,

$$\alpha_i^{(x,y)} = \Pr(x_i = x \wedge f = y)\ .$$

Let $T = (X; y)$ be a sample of size m for f generated according to p. Define

$$K = \{1, \ldots, m\}$$

and

$$K_i^{(x,y)} = \{k \in K \mid x_i(k) = x \text{ and } y(k) = y\}\ .$$

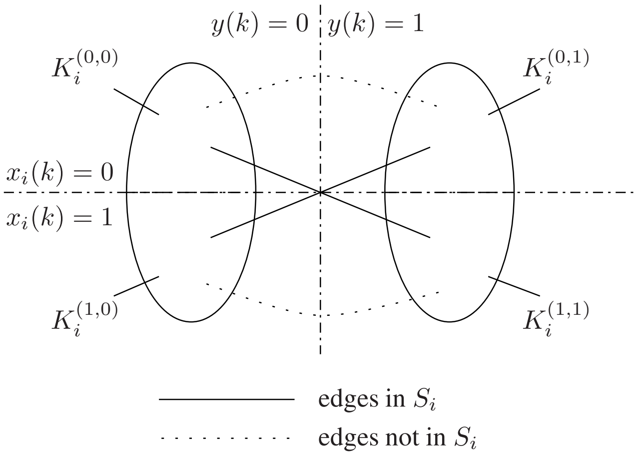

Figure 11: Partition of K with respect to variable x_i

The situation is depicted in Figure 11. Since each example is drawn according to the same probability distribution (i.e., the examples are identically distributed), $|K_i^{(x,y)}|$ can be considered as a binomially distributed random variable with parameters $\alpha_i^{(x,y)}$ and m. Analogously to the $\alpha_i^{(x,y)}$'s, we denote by $\beta_i^{(x,y)}$ the corresponding relative frequencies in the input sample, i.e.,

$$\beta_i^{(x,y)} = \frac{|K_i^{(x,y)}|}{|K|} .$$

Next, we define

$$\alpha_i = \alpha_i^{(0,0)}\alpha_i^{(1,1)} + \alpha_i^{(1,0)}\alpha_i^{(0,1)} \quad \text{and} \quad \beta_i = \beta_i^{(0,0)}\beta_i^{(1,1)} + \beta_i^{(1,0)}\beta_i^{(0,1)} .$$

Consider the set of edges labeled with x_i:

$$S_i = \Big\{\{k,l\} \subseteq K \mid (k,l) \in K_i^{(0,0)} \times K_i^{(1,1)} \cup K_i^{(1,0)} \times K_i^{(0,1)}\Big\} ,$$

It holds $|S_i| = \beta_i m^2$, and for large enough m, we get the approximation $|S_i| \approx \alpha_i m^2$.

Lemma 2 *For fixed $i \in \{1,\ldots,n\}$ and arbitrary $0 \le \delta \le \frac{1}{10}$, it holds that*

$$\Pr\big(\big||S_i| - \alpha_i m^2\big| \ge \delta m^2\big) \;\le\; 8e^{-\frac{1}{3}\delta^2 m} .$$

Proof: For each $i \in \{1,\ldots,n\}$ and $x,y \in \{0,1\}$, we obtain from Fact 2 the Chernoff bounds

$$\Pr\left(|\beta_i^{(x,y)} - \alpha_i^{(x,y)}| \ge \sqrt{\alpha_i^{(x,y)}}\delta\right) = \Pr\left(\Big||K_i^{(x,y)}| - \alpha_i^{(x,y)}m\Big| \ge \sqrt{\alpha_i^{(x,y)}}\delta m\right) \le 2e^{-\frac{1}{3}\delta^2 m} \tag{4}$$

provided that $0 \le \delta \le \sqrt{\alpha_i^{(x,y)}}$ and

$$\Pr\left(|\beta_i^{(x,y)} - \alpha_i^{(x,y)}| \ge 2\delta^2\right) = \Pr\left(\Big||K_i^{(x,y)}| - \alpha_i^{(x,y)}m\Big| \ge 2\delta^2 m\right) \le e^{-\frac{1}{3}\delta^2 m} \tag{5}$$

provided that $\delta > \sqrt{\alpha_i^{(x,y)}}$.

Let $\Delta_i^{(x,y)} = |\beta_i^{(x,y)} - \alpha_i^{(x,y)}|$ for each $x, y \in \{0,1\}$. Then

$$\begin{aligned}
|\beta_i - \alpha_i| &\le |\beta_i^{(0,0)}\beta_i^{(1,1)} - \alpha_i^{(0,0)}\alpha_i^{(1,1)}| + |\beta_i^{(1,0)}\beta_i^{(0,1)} - \alpha_i^{(1,0)}\alpha_i^{(0,1)}| \\
&\le |(\alpha_i^{(0,0)} + (\beta_i^{(0,0)} - \alpha_i^{(0,0)}))(\alpha_i^{(1,1)} + (\beta_i^{(1,1)} - \alpha_i^{(1,1)})) - \alpha_i^{(0,0)}\alpha_i^{(1,1)}| \\
&\quad + |(\alpha_i^{(1,0)} + (\beta_i^{(1,0)} - \alpha_i^{(1,0)}))(\alpha_i^{(0,1)} + (\beta_i^{(0,1)} - \alpha_i^{(0,1)})) - \alpha_i^{(1,0)}\alpha_i^{(0,1)}| \\
&\le \alpha_i^{(0,0)}\Delta_i^{(1,1)} + \alpha_i^{(1,1)}\Delta_i^{(0,0)} + \Delta_i^{(0,0)}\Delta_i^{(1,1)} \\
&\quad + \alpha_i^{(1,0)}\Delta_i^{(0,1)} + \alpha_i^{(0,1)}\Delta_i^{(1,0)} + \Delta_i^{(1,0)}\Delta_i^{(0,1)} .
\end{aligned}$$

Now suppose that for all $x, y \in \{0,1\}$, we have

$$\Delta_i^{(x,y)} \le \begin{cases} \sqrt{\alpha_i^{(x,y)}}\delta & \text{if } \delta \le \sqrt{\alpha_i^{(x,y)}} \\ 2\delta^2 & \text{if } \delta > \sqrt{\alpha_i^{(x,y)}} \end{cases} .$$

In order to find a bound on $|\beta_i - \alpha_i|$ in terms of δ, we have to consider the relationship between δ and the $\alpha_i^{(x,y)}$'s. Let $0 \le a_1 \le a_2 \le a_3 \le a_4 \le 1$ such that $\{a_1, a_2, a_3, a_4\} = \{\alpha_i^{(0,0)}, \alpha_i^{(1,0)}, \alpha_i^{(0,1)}, \alpha_i^{(1,1)}\}$. The following bounds on the underbraced parts can be verified by solving some easy maximization problems.

Case 1: $0 \le \delta^2 \le a_1$. Since $a_1 + a_2 + a_3 + a_4 = 1$, we have $a_1 \le \frac{1}{4}$, and thus $\delta \le \sqrt{\frac{1}{4}} = \frac{1}{2}$.

$$\begin{aligned}
|\beta_i - \alpha_i| &\le \alpha_i^{(0,0)}\Delta_i^{(1,1)} + \alpha_i^{(1,1)}\Delta_i^{(0,0)} + \Delta_i^{(0,0)}\Delta_i^{(1,1)} \\
&\quad + \alpha_i^{(1,0)}\Delta_i^{(0,1)} + \alpha_i^{(0,1)}\Delta_i^{(1,0)} + \Delta_i^{(1,0)}\Delta_i^{(0,1)} \\
&\le \alpha_i^{(0,0)}\sqrt{\alpha_i^{(1,1)}}\delta + \alpha_i^{(1,1)}\sqrt{\alpha_i^{(0,0)}}\delta + \sqrt{\alpha_i^{(0,0)}\alpha_i^{(1,1)}}\delta^2 \\
&\quad + \alpha_i^{(1,0)}\sqrt{\alpha_i^{(0,1)}}\delta + \alpha_i^{(0,1)}\sqrt{\alpha_i^{(1,0)}}\delta + \sqrt{\alpha_i^{(1,0)}\alpha_i^{(0,1)}}\delta^2 \\
&\le \underbrace{\left(\alpha_i^{(0,0)}\sqrt{\alpha_i^{(1,1)}} + \alpha_i^{(1,1)}\sqrt{\alpha_i^{(0,0)}} + \alpha_i^{(1,0)}\sqrt{\alpha_i^{(0,1)}} + \alpha_i^{(0,1)}\sqrt{\alpha_i^{(1,0)}}\right)}_{\le \frac{1}{2}\sqrt{2}}\delta \\
&\quad + \underbrace{\left(\sqrt{\alpha_i^{(0,0)}\alpha_i^{(1,1)}} + \sqrt{\alpha_i^{(1,0)}\alpha_i^{(0,1)}}\right)}_{\le \frac{1}{2}}\delta^2 \\
&\le (\frac{1}{2}\sqrt{2} + \frac{1}{2}\delta)\,\delta \\
&\le (\frac{1}{2}\sqrt{2} + \frac{1}{4})\,\delta \quad \approx \quad 0.96\,\delta < \delta .
\end{aligned}$$

Case 2: $a_1 \le \delta^2 \le a_2$. W.l.o.g. we may assume that $a_1 = \alpha_i^{(0,0)}$.

$$\begin{aligned}
|\beta_i - \alpha_i| &\le \alpha_i^{(0,0)}\Delta_i^{(1,1)} + \alpha_i^{(1,1)}\Delta_i^{(0,0)} + \Delta_i^{(0,0)}\Delta_i^{(1,1)} \\
&\quad + \alpha_i^{(1,0)}\Delta_i^{(0,1)} + \alpha_i^{(0,1)}\Delta_i^{(1,0)} + \Delta_i^{(1,0)}\Delta_i^{(0,1)} \\
&\le \underbrace{\alpha_i^{(0,0)}}_{\le\sqrt{\alpha_i^{(0,0)}}\delta} \sqrt{\alpha_i^{(1,1)}}\delta + \alpha_i^{(1,1)} \cdot 2\delta^2 + 2\delta^2 \cdot \sqrt{\alpha_i^{(1,1)}}\delta \\
&\quad + \alpha_i^{(1,0)}\sqrt{\alpha_i^{(0,1)}}\delta + \alpha_i^{(0,1)}\sqrt{\alpha_i^{(1,0)}}\delta + \underbrace{\sqrt{\alpha_i^{(1,0)}\alpha_i^{(0,1)}}}_{\le\frac{1}{2}}\delta^2 \\
&\le \underbrace{\left(\alpha_i^{(1,0)}\sqrt{\alpha_i^{(0,1)}} + \alpha_i^{(0,1)}\sqrt{\alpha_i^{(1,0)}}\right)}_{\le\frac{1}{2}\sqrt{2}}\delta + \underbrace{(\sqrt{\alpha_i^{(0,0)}\alpha_i^{(1,1)}} + \alpha_i^{(1,1)} + \sqrt{\alpha_i^{(1,0)}\alpha_i^{(0,1)}})}_{\le\frac{17}{8}}\delta^2 + 2\delta^3 \\
&\le (\frac{1}{2}\sqrt{2} + \frac{17}{8}\delta + 2\delta^2)\ \delta\ .
\end{aligned}$$

In particular, for $\delta \le \frac{1}{10}$, we have $|\beta_i - \alpha_i| \le 0.94\ \delta < \delta$.

Case 3: $a_2 \le \delta^2 \le a_3$. W.l.o.g. it is enough to consider two subcases.
Case 3a: $a_1 = \alpha_i^{(0,0)}$, $a_2 = \alpha_i^{(1,1)}$.

$$\begin{aligned}
|\beta_i - \alpha_i| &\le \alpha_i^{(0,0)}\Delta_i^{(1,1)} + \alpha_i^{(1,1)}\Delta_i^{(0,0)} + \Delta_i^{(0,0)}\Delta_i^{(1,1)} \\
&\quad + \alpha_i^{(1,0)}\Delta_i^{(0,1)} + \alpha_i^{(0,1)}\Delta_i^{(1,0)} + \Delta_i^{(1,0)}\Delta_i^{(0,1)} \\
&\le \underbrace{\alpha_i^{(0,0)}}_{\le\delta^2} \cdot 2\delta^2 + \underbrace{\alpha_i^{(1,1)}}_{\le\delta^2} \cdot 2\delta^2 + 2\delta^2 \cdot 2\delta^2 \\
&\quad + \alpha_i^{(1,0)}\sqrt{\alpha_i^{(0,1)}}\delta + \alpha_i^{(0,1)}\sqrt{\alpha_i^{(1,0)}}\delta + \underbrace{\sqrt{\alpha_i^{(1,0)}\alpha_i^{(0,1)}}}_{\le\frac{1}{2}}\delta^2 \\
&\le \underbrace{\left(\alpha_i^{(1,0)}\sqrt{\alpha_i^{(0,1)}} + \alpha_i^{(0,1)}\sqrt{\alpha_i^{(1,0)}}\right)}_{\le\frac{1}{2}\sqrt{2}}\delta + \frac{1}{2}\delta^2 + 8\delta^4 \\
&\le (\frac{1}{2}\sqrt{2} + \frac{1}{2}\delta + 8\delta^3)\ \delta\ .
\end{aligned}$$

In particular, for $\delta \le \frac{1}{4}$, we have $|\beta_i - \alpha_i| \le 0.96\ \delta < \delta$.

Case 3b: $a_1 = \alpha_i^{(0,0)}$, $a_2 = \alpha_i^{(1,0)}$.

$$\begin{aligned}
|\beta_i - \alpha_i| &\le \alpha_i^{(0,0)}\Delta_i^{(1,1)} + \alpha_i^{(1,1)}\Delta_i^{(0,0)} + \Delta_i^{(0,0)}\Delta_i^{(1,1)} \\
&\quad + \alpha_i^{(1,0)}\Delta_i^{(0,1)} + \alpha_i^{(0,1)}\Delta_i^{(1,0)} + \Delta_i^{(1,0)}\Delta_i^{(0,1)} \\
&\le \alpha_i^{(0,0)}\sqrt{\alpha_i^{(1,1)}}\delta + \alpha_i^{(1,1)} \cdot 2\delta^2 + 2\delta^2 \cdot \sqrt{\alpha_i^{(1,1)}}\delta \\
&\quad + \alpha_i^{(1,0)}\sqrt{\alpha_i^{(0,1)}}\delta + \alpha_i^{(0,1)} \cdot 2\delta^2 + 2\delta^2 \cdot \sqrt{\alpha_i^{(0,1)}}\delta \\
&\le \underbrace{\left(\alpha_i^{(0,0)}\sqrt{\alpha_i^{(1,1)}} + \alpha_i^{(1,0)}\sqrt{\alpha_i^{(0,1)}}\right)}_{\le \frac{2}{9}\sqrt{3}} \delta \\
&\quad + 2\underbrace{(\alpha_i^{(1,1)} = \alpha_i^{(0,1)})}_{\le 1}\delta^2 + 2\underbrace{(\sqrt{\alpha_i^{(1,1)}} + \sqrt{\alpha_i^{(0,1)}})}_{\le\sqrt{2}}\delta^3 \\
&\le (\frac{2}{9}\sqrt{3} + 2\delta + 2\sqrt{2}\delta^2)\ \delta\ .
\end{aligned}$$

In particular, for $\delta \le \frac{1}{5}$, we have $|\beta_i - \alpha_i| \le 0.9\ \delta < \delta$.

Case 4: $a_3 \le \delta^2 \le a_4$. W.l.o.g. we may assume that $a_4 = \alpha_i^{(0,1)}$.

$$\begin{aligned}
|\beta_i - \alpha_i| &\le \alpha_i^{(0,0)}\Delta_i^{(1,1)} + \alpha_i^{(1,1)}\Delta_i^{(0,0)} + \Delta_i^{(0,0)}\Delta_i^{(1,1)} \\
&\quad + \alpha_i^{(1,0)}\Delta_i^{(0,1)} + \alpha_i^{(0,1)}\Delta_i^{(1,0)} + \Delta_i^{(1,0)}\Delta_i^{(0,1)} \\
&\le \alpha_i^{(0,0)} \cdot 2\delta^2 + \alpha_i^{(1,1)} \cdot 2\delta^2 + 2\delta^2 \cdot 2\delta^2 \\
&\quad + \alpha_i^{(1,0)}\sqrt{\alpha_i^{(0,1)}}\delta + \alpha_i^{(0,1)} \cdot 2\delta^2 + 2\delta^2 \cdot \sqrt{\alpha_i^{(0,1)}}\delta \\
&\le \underbrace{\left(\alpha_i^{(1,0)}\sqrt{\alpha_i^{(0,1)}}\right)}_{\le\frac{2}{9}\sqrt{3}}\delta + 2\underbrace{(\alpha_i^{(0,0)} + \alpha_i^{(1,1)} + \alpha_i^{(1,1)} + \alpha_i^{(0,1)})}_{\le 1}\delta^2 + 2\underbrace{\sqrt{\alpha_i^{(0,1)}}}_{\le 1}\delta^3 + 4\delta^4 \\
&\le (\frac{2}{9}\sqrt{3} + 2\delta + 2\delta^2 + 4\delta^3)\ \delta\ .
\end{aligned}$$

In particular, for $\delta \le \frac{1}{5}$, we have $|\beta_i - \alpha_i| \le 0.9\ \delta < \delta$.

Case 5: $a_4 \le \delta^2 \le 1$. Since $a_4 \ge \frac{1}{4}$, we have $\delta \ge \sqrt{\frac{1}{4}} = \frac{1}{2}$. This case will not be interesting for us, since we are going to assume $\delta \le \frac{1}{10}$.

Cases 1 to 5 show that for $\delta \le \frac{1}{10}$, we have $|\beta_i - \alpha_i| < \delta$.

Define $\delta^{(x,y)} = \begin{cases} \sqrt{\alpha_i^{(x,y)}}\delta & \text{if } \delta \le \sqrt{\alpha_i^{(x,y)}} \\ 2\delta^2 & \text{if } \delta > \sqrt{\alpha_i^{(x,y)}} \end{cases}$ for each $x, y \in \{0,1\}$. Now $|\beta_i - \alpha_i| \le \delta$ implies $||S_i| - \alpha_i m^2| = |\beta_i - \alpha_i| m^2 \le \delta m^2$. Furthermore we have

$$\begin{aligned}
&\Pr(\forall x, y \in \{0,1\} : |\beta_i^{(x,y)} - \alpha_i^{(x,y)}| \le \delta^{(x,y)}) \\
&\quad = 1 - \Pr(\exists x, y \in \{0,1\} : |\beta_i^{(x,y)} - \alpha_i^{(x,y)}| > \delta^{(x,y)}) \\
&\quad \ge 1 - \sum_{x,y\in\{0,1\}} \Pr(|\beta_i^{(x,y)} - \alpha_i^{(x,y)}| > \delta^{(x,y)}) \\
&\quad \ge 1 - 8e^{-\frac{1}{3}\delta^2 m}\ ,
\end{aligned}$$

where the last inequality follows from inequalities (4) and (5). We conclude that

$$\begin{aligned}\Pr(\big||S_i| - \alpha_i m^2\big| \leq \delta m^2) &= \Pr(|\beta_i - \alpha_i| \leq \delta) \\ &\geq \Pr(\forall x, y \in \{0,1\} : |\beta_i^{(x,y)} - \alpha_i^{(x,y)}| \leq \delta^{(x,y)}) \\ &\geq 1 - 8e^{-\frac{1}{3}\delta^2 m} .\end{aligned}$$

□

The following theorem provides very general conditions that guarantee the correctness of the ranking algorithms with respect to a concept f.

Theorem 3 *Let $f : \{0,1\}^n \to \{0,1\}$ depend on $x_{i_1}, \ldots, x_{i_d}$, let T be a sample for f generated according to a probability distribution $p : \{0,1\}^n \to [0,1]$, and let $c > 0$ be an arbitrary constant.*

(a) If $\min\{\alpha_i \mid x_i \in V^+\} > \max\{\alpha_j \mid x_j \in V^-\}$, then with probability $1 - n^{-c}$, TARGET RANKING *is successful on input (T, d), provided that*

$$m \geq 12\varepsilon^{-2}((c+1)\ln n + \ln 8) ,$$

where

$$\varepsilon = \min\left\{ \min\{\alpha_i \mid x_i \in V^+\} - \max\{\alpha_j \mid x_j \in V^-\} ,\ 1/5\right\} .$$

(b) If $\max\{\alpha_i \mid i \in V^+\} < \min\{\alpha_j \mid j \in V^-\}$, then with probability $1 - n^{-c}$, MODEST TARGET RANKING *is successful on input (T, d), provided that*

$$m \geq 12\varepsilon^{-2}((c+1)\ln n + \ln 8) ,$$

where

$$\varepsilon = \min\left\{ \min\{\alpha_j \mid x_j \in V^-\} - \max\{\alpha_i \mid x_i \in V^+\} ,\ 1/5\right\} .$$

Proof: We first prove part (a). Let

$$t = \frac{1}{2}\left(\min_{i \in V^+} \alpha_i + \max_{j \in V^-} \alpha_j\right) m^2 .$$

Then for $x_i \in V^+$ it holds:

$$\begin{aligned}\Pr(|S_i| \leq t) &\leq \Pr\left(|S_i| \leq \left(\alpha_i - \frac{\varepsilon}{2}\right) m^2\right) \leq \Pr\left(\big||S_i| - \alpha_i m^2\big| \geq \frac{\varepsilon}{2} m^2\right) \\ &\leq 8e^{-\frac{1}{3}\left(\frac{\varepsilon}{2}\right)^2 m} = 8e^{-\frac{1}{12}\varepsilon^2 m} ,\end{aligned}$$

where the last inequality is due to Lemma 2. Similarly, for $x_j \in V^-$,

$$\Pr(|S_j| \geq t) \leq 8e^{-\frac{1}{12}\varepsilon^2 m} .$$

TARGET RANKING is successful on input (T, d) provided that it is correct in all of its d steps. This is exactly the case, if the largest d sets S_i correspond to the relevant variables, i.e., if $\min_{x_i \in V^+} |S_i| > \max_{x_j \in V^-} |S_j|$. We have

$$\begin{aligned}
\Pr\left(\min_{x_i \in V^+} |S_i| > \max_{x_j \in V^-} |S_j|\right) &\geq \Pr\left(\min_{x_i \in V^+} |S_i| > t \text{ and } \max_{x_j \in V^-} |S_j| < t\right) \\
&= 1 - \Pr\left(\exists x_i \in V^+ : |S_i| \leq t \text{ or } \exists x_j \in V^- : |S_j| \geq t\right) \\
&\geq 1 - \left(\sum_{x_i \in V^+} \Pr(|S_i| \leq t) + \sum_{x_j \in V^-} \Pr(|S_j| \geq t)\right) \\
&\geq 1 - 8ne^{-\frac{1}{12}\varepsilon^2 m} .
\end{aligned}$$

Now if $m \geq \frac{12}{\varepsilon^2} m((c+1)\ln n + \ln 8)$, then $8ne^{-\frac{1}{12}\varepsilon^2 m} \leq n^{-c}$, and thus part (a) follows.

The proof of part (b) works similar: Let

$$t = \frac{1}{2}\left(\min_{j \in V^-} \alpha_j + \max_{i \in V^+} \alpha_i\right) m^2 .$$

For the same reasons as in the proof of part (a), we have $\Pr(|S_i| \geq t) \leq 8e^{-\frac{1}{12}\varepsilon^2 m}$ for $x_i \in V^+$ and $\Pr(|S_j| \geq t) \leq 8e^{-\frac{1}{12}\varepsilon^2 m}$ for $x_j \in V^-$. Now MODEST TARGET RANKING is successful on input (T, d), if the smallest d sets S_i correspond to the relevant variables, i.e., if $\min_{x_j \in V^-} |S_j| > \max_{x_i \in V^+} |S_i|$. Again similar to part (a), it holds that

$$\begin{aligned}
\Pr\left(\min_{x_j \in V^-} |S_j| > \max_{x_i \in V^+} |S_i|\right) &\geq \Pr\left(\min_{x_j \in V^-} |S_j| > t \text{ and } \max_{x_i \in V^+} |S_i| < t\right) \\
&= 1 - \Pr\left(\exists x_j \in V^- : |S_j| \leq t \text{ or } \exists x_i \in V^+ : |S_i| \geq t\right) \\
&\geq 1 - \left(\sum_{x_j \in V^-} \Pr(|S_j| \leq t) + \sum_{x_i \in V^+} \Pr(|S_i| \geq t)\right) \\
&\geq 1 - 8ne^{-\frac{1}{12}\varepsilon^2 m} .
\end{aligned}$$

Finally, choosing $m \geq \frac{12}{\varepsilon^2} m((c+1)\ln n + \ln 8)$ we obtain $8ne^{-\frac{1}{12}\varepsilon^2 m} \leq n^{-c}$, and the claim in (b) follows. □

As Theorem 3 is stated for a general setting, let us now consider some typical input distributions and simplify its conditions in these cases.

5.3 Typical Input Distributions

• Independent Attributes (IA)

Suppose that the values for the x_i's ($i = 1, \ldots, n$) are generated independently of each other, say with $\Pr(x_i = 1) = p_i \in [0, 1]$ (thus $\Pr(x_i = 0) = 1 - p_i$). Then we say that the sample is $\mathrm{IA}(p_1, \ldots, p_n)$*-generated*. In this case we have

$$p(x_1, \ldots, x_n) = \prod_{i=1}^{n} p_i^{x_i}(1 - p_i)^{1 - x_i} .$$

Lemma 3 *Let T be an* $\mathrm{IA}(p_1, \dots, p_n)$*-generated sample for f. Then, for $x_j \in V^-$, we have*

$$\alpha_j = 2p_j(1-p_j)\Pr(f=0)\Pr(f=1)\ . \tag{6}$$

Proof: Since in particular the values for relevant and irrelevant variables are generated independently of each other, the outcomes for the irrelevant attributes and the lables are also independent. Thus we have

$$\begin{aligned}
\alpha_j &= \alpha_j^{(0,0)}\alpha_j^{(1,1)} + \alpha_j^{(1,0)}\alpha_j^{(0,1)} \\
&= \Pr(x_j = 0 \wedge f = 0)\Pr(x_j = 1 \wedge f = 1) \\
&\quad + \Pr(x_j = 1 \wedge f = 0)\Pr(x_j = 0 \wedge f = 1) \\
&= (1-p_j)\Pr(f=0)\cdot p_j \Pr(f=1) \ + \ p_j \Pr(f=0)\cdot(1-p_j)\Pr(f=1) \\
&= 2p_j(1-p_j)\Pr(f=0)\Pr(f=1)\ .
\end{aligned}$$

□

• Independent Equiprobable Attributes (IEA)
If T is $\mathrm{IA}(p_1, \dots, p_n)$-generated with $p_1 = \ldots = p_n = q$, then we say that T is $\mathrm{IEA}(q)$-*generated.*

Lemma 4 *Let T be an* $\mathrm{IEA}(q)$*-generated sample for f.*

(a) For each $x_j \in V^-$, it holds $\alpha_j = 2q(1-q)\Pr(f=0)\Pr(f=1)$. In particular, α_j is independent of $x_j \in V^-$. We denote the common value of these α_j's by α^- in this case.

(b) If f is symmetric, then the α_i's with $x_i \in V^+$ are also independent of i. We denote the common value of these α_i's by α^+ in this case.

Proof: Part (a) is just a direct plug-in of Lemma 3.

Since f is symmetric, $\Pr(f = y \mid x_i = x)$ is independent of $x_i \in V^+$, for each pair $x, y \in \{0,1\}$. Consequently,

$$\begin{aligned}
\alpha_i^{(x,y)} &= \Pr(x_i = x \wedge f_i = y) \\
&= \Pr(x_i = x)\cdot\Pr(f = y \mid x_i = x) \\
&= q\cdot\Pr(f = y \mid x_i = x)
\end{aligned}$$

is also independent of x_i. □

From the previous lemma and Theorem 3 we immediately obtain the following result on the successfulness of the target ranking algorithms when applied to symmetric Boolean functions.

Corollary 1 *For f with a symmetric base function $\tilde{f}$, three cases can occur:*

- *$\alpha^+ > \alpha^-$: $O(\log n)$ input examples suffice such that* TARGET RANKING *is successful with high probability.*

- $\alpha^+ < \alpha^-$: *$O(\log n)$ input examples suffice such that* MODEST TARGET RANKING *is successful with high probability.*
- $\alpha^+ = \alpha^-$: *No success ratios can be guaranteed for the ranking algorithms, regardless of how many input examples are provided.*

• Uniformly Distributed Attributes (UDA)
If the examples are uniformly distributed, i.e., if a sample T is IEA($\frac{1}{2}$)-generated, then we say that T is UDA-generated.

Lemma 5 *Let $f : \{0,1\}^n \to \{0,1\}$ depend on $x_{i_1}, \ldots, x_{i_d}$, and let T be a* UDA*-generated sample for f. Define $L_r^{(x,y)} = \left\{(x_1,\ldots,x_d) \in \{0,1\}^d \mid x_r = x \ \wedge \ \tilde{f}(x_1,\ldots,x_d) = y\right\}$ for $x, y \in \{0,1\}$ and $r \in \{1,\ldots,d\}$.*

(a) For $x_i \in V^+$ with $i = i_r$, we have

$$\alpha_i^{(x,y)} = \frac{1}{2^d}|L_r^{(x,y)}| \quad \text{and thus} \quad \alpha_i = \frac{1}{2^{2d}}\left(|L_r^{(0,0)}| \cdot |L_r^{(1,1)}| + |L_r^{(1,0)}| \cdot |L_r^{(0,1)}|\right) \ .$$

(b) For $x_j \in V^-$, we have $\alpha_j^{(x,y)} = \frac{1}{2^{d+1}}\left|\tilde{f}^{-1}(y)\right|$ and thus $\alpha_j = \frac{1}{2^{2d+1}}|\tilde{f}^{-1}(0)| \cdot |\tilde{f}^{-1}(1)|$.

Proof: The first equation in (a) is immediate, since $\Pr(x_i = x \wedge f = y) = \frac{1}{2^d}\left|L_r^{(x,y)}\right|$, and the second equation follows from the definition of α_i.

Next, for $x_j \in V^-$, we have $\alpha_j^{(x,y)} = \Pr(x_j = x) \cdot \Pr(f = y) = \frac{1}{2} \cdot \frac{1}{2^d}\left|\tilde{f}^{-1}(y)\right|$ which yields the equations in (b). □

5.4 The First Step of Greedy

Note that the success probability for the first step of the GREEDY algorithm is higher than the probability that GREEDY RANKING succeeds in all steps. More precisely, it holds that

$$\begin{aligned} &\Pr(\text{GREEDY succeeds in the first step}) \\ &= \Pr\left(\max\{|S_i| \mid i \in V^+\} > \max\{|S_j| \mid j \in V^-\}\right) \\ &\geq \Pr\left(\min\{|S_i| \mid i \in V^+\} > \max\{|S_j| \mid j \in V^-\}\right) \\ &= \Pr(\text{GREEDY RANKING succeeds in all steps}) \ . \end{aligned}$$

However, it is much more complicated to do a careful analysis of the GREEDY algorithm for the remaining steps in this general setting, as the cardinalities for the remaining sets S_i are effected by the removal of other sets S_i. The only way out for this algorithm seems to be to assume more restrictive conditions as for example uniform input distribution and to consider only some special functions as it is done in [4].

5.5 An improved algorithm for monotone functions

Assuming that the underlying function $f : \{0,1\}^d \to \{0,1\}$ of a given IFR instance is *monotone*, we can slightly change the GREEDY RANKING algorithm as follows. Instead of labeling an edge $\{k,l\}$ with $y(k) = 0$ and $y(l) = 1$ of the functional relations graph with all variable indices i such that $x_i(k) \neq x_i(l)$, we only take the indices with $x_i(k) = 0$ and $x_i(l) = 1$. In other words, we work with the edge covering sets

$$S_i^{\mathrm{mon}} = \{\{k,l\} \mid x_i(k) = y(k) = 0 \ \wedge \ x_i(l) = y(l) = 1\} \ .$$

It is easy to see that a set cover $S = S_{i_1}^{\mathrm{mon}} \cup \ldots \cup S_{i_d}^{\mathrm{mon}}$ corresponds to a set of variables $\{x_{i_1}, \ldots, x_{i_d}\}$ on which a monotone function explaining the input table depends and vice versa. For the values $\alpha_i^{(x,y)}$ (now adjusted to the modified algorithm), this means that $\alpha_i^{(0,1)} = \alpha_i^{(1,0)} = 0$, so $\alpha_i = \alpha_i^{(0,0)} \alpha_i^{(1,1)}$. In particular, for the irrelevant variables, the α_j's are exactly half as large as in the general setting, whereas for the for the relevant variables, both types of α_i's coincide. Thus, if (in the general setting) the α_j's with $x_j \in V^-$ are smaller than the α_i's with $x_i \in V^+$ but for only a few $j \in V^-$, then it may be the case that halving the α_j's makes them all smaller than the α_i's. Moreover, the gap $\alpha^+ - \alpha^-$ may be considerably enlarged by using this modified algorithm for monotone functions, resulting in a much lower necessary sample size.

6 Inferring Specific Concepts

Sometimes it is more convenient to express the α_i's in terms of conditional probabilities. Equation (6) can be rewritten as follows:

$$\alpha_j = 2 \Pr(x_j = 0) \Pr(x_j = 1) \Pr(f = 0) \Pr(f = 1) \quad \text{for } x_j \in V^- \ . \tag{7}$$

As we are interested in the difference between α_i and α_j for $x_i \in V^+$ and $x_j \in V^-$ (in particular, if we are faced with IA-, IEA(q), or UDA-distributed samples, we are going to factor out $\Pr(x = 0) \Pr(x = 1)$ or $\Pr(f = 0) \Pr(f = 1)$ in the definition of α_i ($x_i \in V^+$), and then hope to be able to calculate the difference more easily.

Depending on what part we would like to factor out, we can use

$$\Pr(x_i = x \wedge f = y) = \Pr(x_i = x) \Pr(f = y \mid x_i = x)$$

or

$$\Pr(x_i = x \wedge f = y) = \Pr(f = y) \Pr(x = x_i \mid f = y) \ .$$

Using the first equation, we obtain

$$\begin{aligned}
\alpha_i \ &= \ \Pr(x_i = 0 \wedge f = 0) \Pr(x_i = 1 \wedge f = 1) \\
&\qquad + \Pr(x_i = 1 \wedge f = 0) \Pr(x_i = 0 \wedge f = 1) \\
&= \ \Pr(x_i = 0) \Pr(x_i = 1) \\
&\qquad \cdot \ (\Pr(f = 0 \mid x_i = 0) \Pr(f = 1 \mid x_i = 1) \\
&\qquad\quad + \Pr(f = 0 \mid x_i = 1) \Pr(f = 1 \mid x_i = 0)) \ .
\end{aligned}$$

Similarly, using the second equation, we get

$$\begin{aligned}\alpha_i &= \Pr(f=0)\Pr(f=1)\\ &\quad\cdot\left(\Pr(x_i=0 \mid f=0)\Pr(x_i=1 \mid f=1)\right.\\ &\qquad \left.+\Pr(x_i=0 \mid f=1)\Pr(x_i=1 \mid f=0)\right)\ .\end{aligned}$$

These observations will prove very useful when investigating the values for the $\alpha_i's$ for threshold functions in Theorem 7.

We will now consider several basic Boolean functions, starting with the AND-function.

Lemma 6 *Let $\{i_1,\dots,i_d\} \subseteq \{1,\dots,n\}$, let $f:\{0,1\}^n \to \{0,1\}$ be defined by $f(x_1,\dots,x_n) = x_{i_1} \wedge \dots \wedge x_{i_d}$, and let T be a sample for f.*

(a) If T is $\mathrm{IA}(p_1,\dots,p_n)$-generated, then

- *(i) for $x_i \in V^+$, $\alpha_i = (1-p_i)\prod_{x_{i'}\in V^+} p_{i'}$ and*
- *(ii) for $x_j \in V^-$, $\alpha_j = 2p_j(1-p_j)\left(\prod_{x_i\in V^+} p_i\right)\left(1-\prod_{x_i\in V^+} p_i\right)$.*

(b) If T is $\mathrm{IEA}(q)$-generated, then

- *(i) $\alpha^+ = q^d(1-q)$ and*
- *(ii) $\alpha^- = 2q^{d+1}(1-q)(1-q^d)$.*

Proof: By definition, it holds that $y(k)=1$ if and only if $x_{i_1}(k) = x_{i_2}(k) = \dots = x_{i_d}(k) = 1$. Assume that T is $\mathrm{IA}(p_1,\dots,p_n)$-generated. For each $x_i \in V^+$, we have

$$\begin{aligned}\alpha_i^{(0,0)} &= \Pr(x_i=0 \wedge f=0)\\ &= \Pr(x_i=0) \;=\; 1-p_i\ ,\\ \alpha_i^{(1,0)} &= \Pr(x_i=1 \wedge f=0)\\ &= \Pr(x_i=1 \wedge \exists r\in\{1,\dots,d\}: x_{i_r}=0)\\ &= p_i\cdot\left(1-\prod_{x_{i'}\in V^+\setminus\{x_i\}} p_{i'}\right)\ ,\\ \alpha_i^{(0,1)} &= \Pr(x_i=0 \wedge f=1) \;=\; 0\ , \text{and}\\ \alpha_i^{(1,1)} &= \Pr(x_i=1 \wedge f=1)\\ &= \Pr(x_{i_1}=\dots=x_{i_d}=1)\\ &= \prod_{x_{i'}\in V^+} p_{i'}\ .\end{aligned}$$

This leads to

$$\alpha_i = (1-p_i)\prod_{x_{i'}\in V^+} p_{i'}\ .$$

Since

$$\begin{aligned}
\Pr(f=1) &= \Pr(x_{i_1}=\ldots=x_{i_d}=1) &= \prod_{x_i\in V^+} p_i \quad \text{and} \\
\Pr(f=0) &= 1-\Pr(f=1) &= 1-\prod_{x_i\in V^+} p_i\ ,
\end{aligned}$$

we obtain for $x_j \in V^-$ that

$$\alpha_j = 2p_j(1-p_j)\left(\prod_{x_i\in V^+} p_i\right)\left(1-\prod_{x_i\in V^+} p_i\right)$$

from equality (6).

Now part (b) follows from (a) by plugging in $p_1 = \ldots = p_n = q$. □

Theorem 4 (AND-function) *Let $\{i_1,\ldots,i_d\} \subseteq \{1,\ldots,n\}$, let $f : \{0,1\}^n \to \{0,1\}$ be defined by $f(x_1,\ldots,x_n) = x_{i_1} \wedge \ldots \wedge x_{i_d}$, and let T be an $\mathrm{IEA}(q)$-generated sample for f.*

(a) If $q \le \frac{1}{2}$, then it holds that $\alpha^+ > \alpha^-$. Thus the success ratio for Target Ranking *may be raised arbitrarily close to 1 by choosing a large enough sample size $m \in O(\log n)$.*

(b) If $q = \frac{1}{2}$, then it holds that $\alpha^+ - \alpha^- = 2^{-2d-1} > 0$. Thus the success ratio for Target Ranking *may be raised arbitrarily close to 1 by choosing a large enough sample size $m \in O(\log n)$ with the constant being of order 2^{4d}.*

(c) If $q > \frac{1}{2}$, then for sufficiently large d, we have $\alpha^+ < \alpha^-$. Thus the success ratio for Modest Target Ranking *may be raised arbitrarily close to 1 by choosing a large enough sample size $m \in O(\log n)$.*

The same ideas apply to the OR-function, except that we have to substitute q by $1-q$.

Proof: By Part (b) of Lemma 6,

$$\alpha^+ > \alpha^- \iff q(1-q^d) < \frac{1}{2}\ .$$

In particular, $\alpha^+ > \alpha^-$, for $q \le \frac{1}{2}$ which proves (a) (along with Theorem 3).

Part (b) is obtained from Lemma 6 (b) for $q = \frac{1}{2}$.

Furthermore, for sufficiently large d, we have $q(1-q^d) > \frac{1}{2}$, so $\alpha^+ < \alpha^-$. Hence (c) follows for $q > \frac{1}{2}$. □

Theorem 5 (Monomials) *Let $\{i_1,\ldots,i_d\} \subseteq \{1,\ldots,n\}$, let $l_r \in \{x_{i_r}, \neg x_{i_r}\}$ for each $r \in \{1,\ldots,d\}$, and let $f : \{0,1\}^n \to \{0,1\}$ be defined by $f(x_1,\ldots,x_n) = l_1 \wedge \ldots \wedge l_d$. Let T be a* UDA*-generated sample for f. Then* Target Ranking *is successful with high probability provided that a sample of size $m \in \Omega(2^{4d} \cdot \log n)$ is given.*

Proof: For $l_r = x_{i_r}$, we have $\alpha_i^{(0,0)} = \frac{1}{2}$, $\alpha_i^{(1,0)} = \frac{2^{d-1}-1}{2^d}$, $\alpha_i^{(0,1)} = 0$, and $\alpha_i^{(1,1)} = \frac{1}{2^d}$. For $l_r = \neg x_{i_r}$, we have $\alpha_i^{(0,0)} = \frac{2^{d-1}-1}{2^d}$, $\alpha_i^{(1,0)} = \frac{1}{2}$, $\alpha_i^{(0,1)} = \frac{1}{2^d}$, and $\alpha_i^{(1,1)} = 0$. In both cases it follows that $\alpha_i = \frac{1}{2^{d+1}}$. For $x_j \in V^-$ the analysis is the same as for the AND-function in part (b) of Theorem 4, so $\alpha_j = \frac{2^d-1}{2^{2d+1}}$. In particular, it holds that $\alpha^+ - \alpha^- = 2^{-2d-1} > 0$. Hence the claim follows from Theorem 3. □

Akutsu, Miyano, and Kuhara [4] showed a similar result for the GREEDY algorithm. Note that for monomials under uniform input distribution, 2^d rows are necessary in order to obtain (in expectation) at least one example with label 1. (If there is no such example, then the sample can be explained by the constant zero function.)

Theorem 6 (Negation of functions) *Let $f, g : \{0,1\}^n \to \{0,1\}$ with $f(x_1, \ldots, x_n) = \overline{g(x_1, \ldots, x_n)}$ for all $x_1, \ldots, x_n \in \{0,1\}$. If we denote the corresponding α_i's for f and g by $\alpha_i(f)$ and $\alpha_i(g)$, respectively, then it holds that $\alpha_i(f) = \alpha_i(g)$ for all $i \in \{1, \ldots, n\}$. Thus the success ratios of the algorithms remain the same in case the underlying concept is negated.*

Proof: We have $\alpha_i^{(x,y)}(f) = \Pr(x_i = x \wedge f = y) = \Pr(x_i = x \wedge g = 1 - y) = \alpha_i^{(x,1-y)}(g)$, so

$$\begin{aligned}
\alpha_i(f) &= \alpha_i^{(0,0)}(f)\alpha_i^{(1,1)}(f) + \alpha_i^{(1,0)}(f)\alpha_i^{(0,1)}(f) \\
&= \alpha_i^{(0,1)}(g)\alpha_i^{(1,0)}(g) + \alpha_i^{(1,1)}(g)\alpha_i^{(0,0)}(g) \\
&= \alpha_i(g) \;.
\end{aligned}$$

□

Corollary 2 (Clauses) *Theorem 5 also holds, if f is a clause, i.e., if $f : \{0,1\}^n \to \{0,1\}$ is defined by $f(x_1, \ldots, x_n) = l_1 \vee \ldots \vee l_d$, where $l_r \in \{x_{i_r}, \neg x_{i_r}\}$ for each $r \in \{1, \ldots, d\}$ and some subset $\{i_1, \ldots, i_d\}$ of $\{1, \ldots, n\}$.*

Theorem 7 (Threshold functions) *Let $\{i_1, \ldots, i_d\} \subseteq \{1, \ldots, n\}$, $t \in \{1, \ldots, d\}$, and $f : \{0,1\}^n \to \{0,1\}$ be defined by $f(x_1, \ldots, x_n) = 1$ iff $\sum_{r=1}^{d} x_{i_r} \geq t$. Let T be a* UDA*-generated sample for f. Then* TARGET RANKING *is successful with high probability, provided that $m \in \Omega\left(\binom{d}{t}^{-4} 2^{4d} \log n\right)$.*

Proof: We calculate some conditional probabilities. Let $x_i \in V^+$ and $x_j \in V^-$. Define the random variables $X = \sum_{x_k \in V^+} x_k$ and $X' = \sum_{x_k \in V^+ \setminus \{x_i\}} x_k = X - x_i$. In order to compute α_i, we calculate

$$\begin{aligned}
\Pr(f = 0 \mid x_i = 0) &= \Pr(X' \leq t - 1) \;, \\
\Pr(f = 0 \mid x_i = 1) &= \Pr(X' \leq t - 2) \;, \\
\Pr(f = 1 \mid x_i = 0) &= \Pr(X' \geq t) \;, \text{ and} \\
\Pr(f = 1 \mid x_i = 1) &= \Pr(X' \geq t - 1) \;.
\end{aligned}$$

This results in

$$\begin{aligned}\alpha_i &= \Pr(x_i = 0)\Pr(x_i = 1) \\ &\quad \cdot (\Pr(X' \le t-1)\Pr(X' \ge t-1) \; + \; \Pr(X' \le t-2)\Pr(X' \ge t))\end{aligned}$$

Let $a = \Pr(X' \le t-2)$, $b = \Pr(X' = t-1)$, and $c = \Pr(X' \ge t)$. Then we have

$$\alpha_i = \Pr(x_i = 0)\Pr(x_i = 1)\left((a+b)(c+b) + ac\right) \; .$$

Now let us turn to the α_j's. We express some probabilities for X in terms of probabilities of X' as follows:

$$\Pr(X \ge k) = \Pr(x_i = 1)\Pr(X' \ge k-1) + \Pr(x_i = 0)\Pr(X' \ge k)$$

and similarly

$$\Pr(X \le k) = \Pr(x_i = 1)\Pr(X' \le k-1) + \Pr(x_i = 0)\Pr(X' \le k) \; .$$

This implies

$$\begin{aligned}\Pr(f=0) &= \Pr(X \le t-1) \\ &= \Pr(x_i = 1)\Pr(X' \le t-2) + \Pr(x_i = 0)\Pr(X' \le t-1) \quad \text{and} \\ \Pr(f=1) &= \Pr(X \ge t) \\ &= \Pr(x_i = 1)\Pr(X' \ge t-1) + \Pr(x_i = 0)\Pr(X' \ge t) \; .\end{aligned}$$

Plugging the results into equation (7), we obtain for $x_j \in V^-$

$$\begin{aligned}\alpha_j &= 2\Pr(x_i = 0)\Pr(x_i = 1) \\ &\quad \cdot (\Pr(x_i = 1)\Pr(X' \le t-2) + \Pr(x_i = 0)\Pr(X' \le t-1)) \\ &\quad \cdot (\Pr(x_i = 1)\Pr(X' \ge t-1) + \Pr(x_i = 0)\Pr(X' \ge t)) \; .\end{aligned}$$

Now if we assume that T is IEA(q)-generated, then $X \sim B(d,q)$ and $X' \sim B(d-1,q)$. Continuing our analysis under this assumption, we can further write

$$\begin{aligned}\alpha^+ &= \alpha_i = q^2(2ac + ab + bc + b^2) \quad \text{and} \\ \alpha^- &= \alpha_j = 2q^2\left(qa + (1-q)(a+b)\right)\left(q(b+c) + (1-q)c\right) \\ &= 2q^2(a + (1-q)b)(c + qb) \\ &= 2q^2(ac + qab + (1-q)bc + q(1-q)b^2)\end{aligned}$$

and thus

$$\alpha^+ - \alpha^- = q^2\left((1-2q)b(a+b-c) + 2q^2b^2\right) \; .$$

In particular, if $q \le \frac{1}{2}$ and $\Pr(X' \le t-1) \ge \Pr(X' \ge t)$, then $\alpha^+ > \alpha^-$.

For the special case $q = \frac{1}{2}$, we thus have $\alpha^+ - \alpha^- = 2\left(\frac{1}{2}\right)^4 b^2$. Now $b = \binom{d-1}{t-1}\left(\frac{1}{2}\right)^{d-1}$, so

$$\alpha^+ - \alpha^- = \binom{d-1}{t-1}^2 2^{-2d-1} \; . \tag{8}$$

□

If $t = d$ in the previous theorem, then $f = \mathrm{AND}$, and we recover our result from Theorem 4, part (a). Moreover, under uniformly distributed inputs, the gap between α^+ and α^- for threshold functions is smallest for $t \in \{1, d\}$. The largest such gap is reached for $t = \lceil \frac{d}{2} \rceil$, the majority function. Since $\binom{d-1}{t/2-1} \in \Theta(\frac{1}{\sqrt{d}} 2^d)$, we have $\alpha^+ - \alpha^- \in \Theta(d^{-1})$. Applying Theorem 3, this proves the following corollary.

Corollary 3 (Majority function) *Let* $f : \{0,1\}^n \to \{0,1\}$ *such that its base function* $\tilde{f} : \{0,1\}^d \to \{0,1\}$ *is the majority function. Then* TARGET RANKING *is succeessful with high probability, provided that* $m \in \Omega(d^2 \cdot \log n)$.

For symmetric Boolean functions, one cannot always guarantee $\alpha^+ \neq \alpha^-$, even for UDA-generated samples. We will show that the parity function

$$f(x_1, \ldots, x_n) = (x_{i_1} + \ldots + x_{i_d}) \bmod 2$$

(with $d \geq 2$) is a simple counter-example: It holds that $\alpha_i^{(x,y)} = \Pr(x_i = x \wedge f = y) = \frac{1}{2}$ for all $i \in \{1, \ldots, n\}$ and $x, y \in \{0,1\}$, so $\alpha_i = \frac{1}{8}$ for all $i \in \{1, \ldots, n\}$, no matter whether $x_i \in V^+$ or $x_i \in V^-$. Thus the ranking strategies do not work for the parity function. We provide an alternative solution for such concepts in Section 8.

7 Robust Inference

As real data usually contain noise, our main goal in this paper is to handle cases in which the attribute values underly certain disturbances. More precisely, we assume that in each input example, attribute x_i is flipped with probability δ_i, i.e., an algorithm obtains $\overline{x_i(k)}$ instead of the correct value $x_i(k)$ with probability δ_i. We call the resulting set of disturbed examples a δ-*disturbed sample*, where $\delta = (\delta_1, \ldots, \delta_n)$. Note that this assumption introduces a linear number of faults (with respect to the number of attributes).

Fortunately, it can be shown that the ranking algorithms still perform well, if they are given such a disturbed sample. The key idea in this case is to examine how much the sets S_i computed by the ranking algorithms deviate from the S_i's intended by the real data. We will denote the sets derived from the disturbed data with hats (e.g., $\hat{\beta}_i^{(x,y)}$ and so forth). Furthermore we need some more notation: Let

$$K^{(y)} = \{k \in \{1, \ldots, m\} \mid y(k) = y\} ,$$

$$F_i = \{k \in \{1, \ldots, m\} \mid \text{ the input table contains } \overline{x_i(k)} \text{ instead of } x_i(k)\} ,$$

and

$$F_i^{(y)} = F_i \cap K^{(y)}$$

for $y \in \{0,1\}$ and $i \in \{1, \ldots, n\}$. The following lemma is analogous to Lemma 2:

Lemma 7 *Let $i \in \{1, \ldots, n\}$ with $\delta_i \leq \frac{1}{30}$. Then, for ε such that $6\delta_i \leq \varepsilon \leq \frac{1}{5}$, it holds that*

$$\Pr\left(\left||\hat{S}_i| - \alpha_i m^2\right| \geq \varepsilon m^2\right) \;\leq\; 9e^{-\frac{1}{12}\varepsilon^2 m} \;.$$

Proof: We use the inequality

$$\left||\hat{S}_i| - \alpha_i m^2\right| \leq \left||\hat{S}_i| - |S_i|\right| + \left||S_i| - \alpha_i m^2\right|$$

and compute the probability that each of the summands on the right hand side is bounded by $\frac{\varepsilon}{2}m^2$.

Let $\kappa^{(x,y)} = \left||\hat{K}_i^{(x,y)}| - |K_i^{(x,y)}|\right|$ for $x, y \in \{0,1\}$. It is easy to see that

$$\hat{K}_i^{(x,y)} \triangle K_i^{(x,y)} = (K_i^{(1-x,y)} \cap F_i) \cup (K_i^{(x,y)} \cap F_i) = K^{(y)} \cap F_i \;,$$

so that $\kappa^{(x,y)}| \leq \left|\hat{K}_i^{(x,y)} \triangle K_i^{(x,y)}\right| = |K^{(y)} \cap F_i|$. Now similar to the bounding of $|\beta_i - \alpha_i|$ in Lemma 2, we obtain

$$\begin{aligned}
\left||\hat{S}_i| - |S_i|\right| \;&\leq\; |K_i^{(0,0)}| \cdot \kappa^{(1,1)} + |K_i^{(1,1)}| \cdot \kappa^{(0,0)} + \kappa^{(0,0)} \cdot \kappa^{(1,1)} \\
&\qquad + |K_i^{(0,1)}| \cdot \kappa^{(1,0)} + |K_i^{(1,0)}| \cdot \kappa^{(0,1)} + \kappa^{(1,0)} \cdot \kappa^{(0,1)} \\
&\leq\; |K_i^{(0,0)}| \cdot |K^{(1)} \cap F_i| + |K_i^{(1,1)}| \cdot |K^{(0)} \cap F_i| \\
&\qquad + |K_i^{(1,0)}| \cdot |K^{(1)} \cap F_i| + |K_i^{(0,1)}| \cdot |K^{(0)} \cap F_i| \\
&\qquad\quad + 2 \cdot |K^{(0)} \cap F_i| \cdot |K^{(1)} \cap F_i| \\
&=\; \left(|K_i^{(0,0)}| + |K_i^{(1,0)}|\right) \cdot |K^{(1)} \cap F_i| + \left(|K_i^{(1,1)}| + |K_i^{(0,1)}|\right) \cdot |K^{(0)} \cap F_i| \\
&\qquad\quad + 2 \cdot |K^{(0)} \cap F_i| \cdot |K^{(1)} \cap F_i| \\
&=\; \underbrace{|K_i^{(0)}| \cdot |K^{(1)} \cap F_i| + |K_i^{(1)}| \cdot |K^{(0)} \cap F_i|}_{\leq m|F_i|} + 2 \cdot \underbrace{|K^{(0)} \cap F_i| \cdot |K^{(1)} \cap F_i|}_{\leq \frac{1}{4}|F_i|^2} \\
&\leq\; m|F_i| + \frac{1}{2}|F_i|^2 \;.
\end{aligned}$$

In particular, if $|F_i| \leq \frac{1}{3}\varepsilon m$, then $\left||\hat{S}_i| - |S_i|\right| \leq \frac{1}{2}\varepsilon m^2$. From part (b) in Fact 2, we have

$$\Pr\left(|F_i| \geq \frac{\varepsilon}{3}m\right) \leq e^{-\frac{1}{3}\cdot\frac{\varepsilon}{6}\cdot m} = e^{-\frac{1}{18}\varepsilon m} \;,$$

since $\delta_i < \frac{\varepsilon}{6}$ (note that $|F_i|$ can be considered as a binomially distributed random variable with parameters δ_i and m). Now

$$\begin{aligned}
\Pr\left(\left||\hat{S}_i| - \alpha_i m^2\right| \geq \varepsilon m^2\right) \;&\leq\; \Pr\left(\left||\hat{S}_i| - |S_i|\right| \geq \frac{\varepsilon}{2}m^2 \vee \left||S_i| - \alpha_i m^2\right| \geq \frac{\varepsilon}{2}m^2\right) \\
&\leq\; \Pr\left(\left||\hat{S}_i| - |S_i|\right| \geq \frac{\varepsilon}{2}m^2\right) + \Pr\left(\left||S_i| - \alpha_i m^2\right| \geq \frac{\varepsilon}{2}m^2\right) \\
&\leq\; \Pr\left(|F_i| \geq \frac{\varepsilon}{3}\right) + 8e^{-\frac{1}{3}\left(\frac{\varepsilon}{2}\right)^2 m} \\
&\leq\; e^{-\frac{1}{18}\varepsilon m} + 8e^{-\frac{1}{12}\varepsilon^2 m} \\
&\leq\; 9e^{-\frac{1}{12}\varepsilon^2 m}
\end{aligned}$$

where we make use of Lemma 2 and the fact that $\frac{1}{18}\varepsilon m > \frac{1}{12}\varepsilon^2 m$ for $\varepsilon \leq \frac{1}{5}$. □

Besides the general information theoretical problem that a sample may already be explained by a proper subset of the relevant variables, just the opposite phenomenon can occur due to disturbances: $\hat{S}$ may not be covered by $\hat{S}_1, \ldots, \hat{S}_n$, so GREEDY, GREEDY RANKING, and MODEST RANKING — as introduced in Section 4 — do not terminateon the corresponding input samples. Therefore, when faced with the disturbed situation, we modify our algorithms a bit: We ignore all edges that do not belong to any of the computed $\hat{S}_i$'s, i.e., we compute a new

$$\hat{S}_{\text{new}} := \hat{S} \setminus \{\{k,l\} \in \hat{S} \mid \forall i \in \{1,\ldots,n\} : \{k,l\} \notin \hat{S}_i\} \ .$$

Note that the ignored edges are exactly those connecting two example nodes with identical attribute values but different labels. All algorithms are supposed to run with this set $\hat{S}_{\text{new}}$ instead of $\hat{S}$. However, this modification does not effect our analysis, so we continue by writing $\hat{S}$.

In the noisy scenario, Lemma 1 has to be modified as follows:

Lemma 8 *Let f be a concept depending on d variables, and T a δ-disturbed sample for f such that* TARGET RANKING *(resp.,* MODEST TARGET RANKING*) is successful on* (T, d).

(a) If GREEDY RANKING *(resp.,* MODEST RANKING*) outputs at most d variables on input T, then it is correct.*

(b) Otherwise, the first d variables output by GREEDY RANKING *(resp.,* MODEST RANKING*) are the relevant ones.*

Proof: For notational convenience, we omit the "(resp.)" parts. If TARGET RANKING is successful on input (T, d), then the largest d sets S_i exactly correspond to the relevant variables. Thus, if less then d sets suffice to cover S, then GREEDY RANKING is correct. On the other hand, if d sets do not suffice, then at least the first d variables output by GREEDY are the relevant ones. □

We now state our main theorem for the case of disturbed samples:

Theorem 8 *Let $f : \{0,1\}^n \to \{0,1\}$ with relevant variables $x_{i_1}, \ldots, x_{i_d}$, and let $\delta = (\delta_1, \ldots, \delta_n) \in [0,1]^n$. Let T be a δ-disturbed sample for f generated according to a probability distribution $p : \{0,1\}^n \to [0,1]$, and let $c > 0$ be an arbitrary constant.*

(a) If $\min\{\alpha_i \mid x_i \in V^+\} > \max\{\alpha_j \mid x_j \in V^-\}$ and $\delta_k \leq \frac{1}{12}\varepsilon$ for all $k \in \{1,\ldots,n\}$, then with probability $1 - n^{-c}$, TARGET RANKING *is successful on input (T, d), provided that*

$$m \geq 48\, \varepsilon^{-2}((c+1)\ln n + \ln 9) \ ,$$

where

$$\varepsilon = \min\{\ \min\{\alpha_i \mid i \in V^+\} - \max\{\alpha_j \mid j \in V^-\}\ ,\ 2/5\} \ .$$

(b) If $\max\{\alpha_i \mid i \in V^+\} < \min\{\alpha_j \mid j \in V^-\}$ *and* $\delta_k \leq \frac{1}{12}\varepsilon$ *for all* $k \in \{1, \ldots, n\}$*, then* MODEST TARGET RANKING *is successful on input* (T, d)*, provided that*

$$m \geq 48\,\varepsilon^{-2}((c+1)\ln n + \ln 9)\ ,$$

where

$$\varepsilon = \min\{\ \min\{\alpha_j \mid j \in V^-\} - \max\{\alpha_i \mid i \in V^+\}\ ,\ 2/5\}\ .$$

Proof: The proof is similar to the proof of Theorem 3. We only prove (a), since (b) follows the same line. Let

$$t = \frac{1}{2}\left(\min_{x_i \in V^+} \alpha_i + \max_{x_j \in V^-} \alpha_j\right) m^2\ .$$

Then it holds for $x_i \in V^+$ that

$$\begin{aligned} \Pr(|\hat{S}_i| \leq t) \ &\leq\ \Pr\left(|\hat{S}_i| \leq (\alpha_i - \frac{\varepsilon}{2})m^2\right)\ \leq\ \Pr\left(\left||\hat{S}_i| - \alpha_i m^2\right| \geq \frac{\varepsilon}{2}m^2\right) \\ &\leq\ 9e^{-\frac{1}{12}\left(\frac{\varepsilon}{2}\right)^2 m}\ \leq\ 9e^{-\frac{1}{48}\varepsilon^2 m}\ , \end{aligned}$$

where the last inequality is due to Lemma 7. Similarly, for $x_j \in V^-$, $\Pr(|S_j| \geq t) \leq 9e^{-\frac{1}{48}\varepsilon^2 m}$. For the same reasons given in the proof of Theorem 3, we calculate the following bound:

$$\begin{aligned} \Pr(\min_{x_i \in V^+} |\hat{S}_i| > \max_{x_j \in V^-} |\hat{S}_j|) \ &\geq\ \Pr(\min_{x_i \in V^+} |\hat{S}_i| > t \text{ and } \max_{x_j \in V^-} |\hat{S}_j| < t) \\ &=\ 1 - \Pr(\exists x_i \in V^+ : |\hat{S}_i| \leq t \text{ or } \exists x_j \in V^- : |\hat{S}_j| \geq t) \\ &\geq\ 1 - \left(\sum_{x_i \in V^+} \Pr\left(|\hat{S}_i| \leq t\right) + \sum_{x_j \in V^-} \Pr\left(|\hat{S}_j| \geq t\right)\right) \\ &\geq\ 1 - 9ne^{-\frac{1}{48}\varepsilon^2 m}\ . \end{aligned}$$

Now if $m \geq \frac{48}{\varepsilon^2}((c+1)\ln n + \ln 9)$, then $9ne^{-\frac{1}{48}\varepsilon^2 m} \leq n^{-c}$ and thus the claim follows. □

We would like to stress that the algorithms have not been modified in any way in order to overcome the disturbances. In particular, we do not have to assume that the algorithms have any knowledge about the error probabilities $\delta_1, \ldots, \delta_n$. Even more, the sample size required for TARGET RANKING only has to be enlarged by the factor 4 in order to obtain the same success probability in case of a (small) constant percentage of errors in the input sample.

8 Inferring Relevant Attributes of the Parity Function

Throughout this section we identify $\{0,1\}$ with the two-element field $\mathrm{GF}(2)$ and denote by $\oplus$ the sum operation in this field. Furthermore, we define $|\xi| = \sum_{i=1}^{n} \xi_i$ for $\xi \in \{0,1\}^n$ (here the sum is taken in $\mathbb{Z}$).

Let $f : \{0,1\}^n \to \{0,1\}$ be defined by $f(x_1, \ldots, x_n) = x_{i_1} \oplus \ldots \oplus x_{i_d}$ for some set of variable indices $I = \{i_1, \ldots, i_d\} \subseteq \{1, \ldots, n\}$. Since ranking the variables according to their occurences in the functional relations graph does not work for the parity function, we present a different algorithm PARITY INFER to find the relevant variables. The idea is simply to compute a solution of a system of linear equations associated with the input sample and then to infer from this solution a set of variables that can explain the sample.

```
        Algorithm Parity Infer
input (X;y) ∈ {0,1}^{m×(n+1)}
solve Xξ = y
if there is no solution
   then output ``sample contains wrong data''
   else choose any solution ξ
        output all x_i's such that ξ_i = 1
```

Let us again differenciate between the two aspects, the optimization problem INFRA($\oplus$) obtained by restricting the instances and the solutions of INFRA to samples for concepts whose base functions are parity functions – such functions can be uniquely described by the set V^+ of relevant variables – and on the other hand finding exactly the relevant variables of a given but unknown parity function provided that the sample size is large enough.

Let $T = (X; y) = (x_i(k), y(k)) \in \{0,1\}^{m\times(n+1)}$ be a sample for f. Note that there is a one-to-one correspondence between solutions V^+ of the $\mathrm{INFRA}(\oplus)$-instance $T = (X; y)$ and the solutions $\xi \in \{0,1\}^n$ for the system of linear equations

$$X\xi = y \tag{9}$$

given by $\xi_i = 1$ iff $x_i \in V^+$. The task of finding an optimal solution for a given $\mathrm{INFRA}(\oplus)$-instance is equivalent to finding a solution ξ of equation (9) with minimum $|\xi|$. Since $\{x_i \mid i \in I\}$ is a solution for T, equation (9) has at least one solution. Moreover, if X has full rank (i.e., rank$(X) = n$), then equation (9) has a unique solution which is of course also an optimal solution in this case.

In order to establish a lower bound on the approximability of INFRA($\oplus$), we consider the well-studied NEAREST CODEWORD problem. A NEAREST CODEWORD instance consists of a matrix $A \in \{0,1\}^{n\times r}$ and a vector $b \in \{0,1\}^n$. A solution is a vector $x \in \{0,1\}^r$, and the goal is to minimize the Hamming distance of Ax and b (i.e., $|Ax \oplus b|$).

Proposition 3 NEAREST CODEWORD *is reducible to* $\mathrm{INFRA}(\oplus)$ *via a polynomial time computable approximation factor preserving reduction.*

Proof: Let $A \in \{0,1\}^{n\times r}$ and $b \in \{0,1\}^n$. We denote by A^T the transposed matrix. Let $x_1, \ldots, x_m \in \{0,1\}^n$ such that $\mathrm{Ke}(A^T) = \langle x_1, \ldots, x_m\rangle$, where Ke denotes the kernel of a matrix, and $\langle x_1, \ldots, x_m\rangle$ denotes the linear span of the vectors. Define

$$X = \begin{pmatrix} x_1^T \\ \vdots \\ x_m^T \end{pmatrix} \in \{0,1\}^{m\times n}$$

and $y = X \cdot b \in \{0,1\}^m$. Note that $\mathrm{Ke}(X) = \mathrm{Rg}(A)$ (Rg denotes the range of a matrix) and in particular $X \cdot A = 0$. We show that for every $\ell \in \mathbb{N}$, there exists $x \in \{0,1\}^r$ such that $|Ax \oplus b| = \ell$ iff the $\mathrm{INFRA}(\oplus)$ instance $(X; y)$ has a solution of measure ℓ.

The first direction is easy: If $|Ax \oplus b| = \ell$, then $\xi = Ax \oplus b$ corresponds to a solution of $(X; y)$ of measure $|\xi| = \ell$, since $X\xi = X(Ax \oplus b) = X \cdot b = y$.

On the other hand, if $X\xi = y$ for a given $\xi \in \{0,1\}^n$ with $|\xi| = \ell$, then we have to find $x \in \{0,1\}^r$ such that $|Ax \oplus b| = |\xi|$: Since $X\xi = Xb$, we have $X(\xi \oplus b) = 0$, so $\xi \oplus b \in \mathrm{Ke}(X) = \mathrm{Rg}(A)$. Thus there exists $x \in \{0,1\}^r$ such that $Ax = \xi \oplus b$, and so $|Ax \oplus b| = |\xi|$. □

Combining Theorem 3 with a result of Arora, Babai, and Stern [6], we obtain the following theorem.

Theorem 9 *For any $\varepsilon > 0$, INFRA($\oplus$) cannot be approximated within a factor of $2^{\log^{1-\varepsilon} m}$ unless NP $\subseteq$ DTIME($n^{\mathrm{polylog}(n)}$).*

Despite this negative result, INFRA($\oplus$) can be solved efficiently on the average. We show that under certain assumptions the variables detected by PARITY INFER are exactly the relevant ones with high probability.

Theorem 10 *Let $f : \{0,1\}^n \to \{0,1\}$ such that its base function $\tilde{f}$ is a parity function, and let $T = (X; y) \in \{0,1\}^{m\times n} \times \{0,1\}^n$ be a UDA-generated sample for f. If $m \geq n + k(2\ln k + 1)$ with $k = c\log n + 1$ for some $c > 0$, then with probability $1 - n^{-c}$, equation (9) has exactly one solution.*

Proof: Let $k \in \mathbb{N}$, and suppose that $m = n - k + c_1 + \ldots + c_k$ for some $c_1, \ldots, c_k \in \mathbb{N}$. We denote by X_ℓ the matrix consisting of the first ℓ rows of X (for $\ell \in \{1, \ldots, m\}$). Define a sequence of increasing matrix lengths as follows: $\ell_i = i$ for $1 \leq i \leq n-k$ and $\ell_{n-k+j} = n - k + c_1 + \ldots + c_j$ for $1 \leq j \leq k$. Now if $\mathrm{rank}(X) < n$, then there is some $i \in \{1, \ldots, n\}$ such that $\mathrm{rank}(X_{\ell_i}) < i$. Choosing i minimal, we have $\mathrm{rank}(X_{\ell_1}) = 1, \ldots, \mathrm{rank}(X_{\ell_{i-1}}) = i - 1 = \mathrm{rank}(X_{\ell_i})$. The probability for this event is bounded by the probability that all rows $\ell_{i-1} + 1, \ldots, \ell_i$ lie in the $(i-1)$-dimensional linear hull of rows $1, \ldots, \ell_{i-1}$ which is $(2^{i-1}/2^n)^{\ell_i - \ell_{i-1}}$.

Thus, the probability that X is nonregular is bounded by

$$2^{-n} + 2^{-n+1} + \ldots + 2^{-k-1} + 2^{-c_1 k} + 2^{-c_2(k-1)} + \ldots + 2^{-c_k \cdot 1} \leq 2^{-k} + \sum_{i=1}^{k} 2^{-c_i(k-i+1)} .$$

We choose $c_i = \lceil \frac{k+i}{k-i+1} \rceil$. Then we have $\Pr(X \text{ is nonregular}) \leq 2^{-k} + 2^{-k-1} + \ldots + 2^{-2k} \leq 2^{-k+1}$, since $c_i \cdot (k - i + 1) \geq k + i$. Moreover, since $\lceil \frac{k+i}{k-i+1} \rceil \leq \frac{2k}{k-i+1}$ for $1 \leq i \leq k$, we have

$$\begin{aligned} m &= n - k + \sum_{i=1}^{k} c_i \quad = \quad n - k + \left\lceil \frac{k+1}{k} \right\rceil + \ldots + \left\lceil \frac{2k}{1} \right\rceil \\ &\leq n - k + 2k\left(\frac{1}{k} + \ldots + \frac{1}{1}\right) \quad \leq \quad n + k(2\ln k + 1) . \end{aligned}$$

Finally, we choose $k = c\log n + 1$ and obtain $\Pr(X \text{ is nonregular}) \leq n^{-c}$ for $m \geq n + k(2\ln k + 1)$. □

Corollary 4 *Under the conditions described in Theorem 10, using sample size* $m = n + O(c \log n \log\log n)$ PARITY INFER *is successful with probability* $1 - n^{-c}$*, where* c *may be chosen arbitrarily large.*

9 Conclusions and Further Research

In this paper we have investigated the problem of inferring relevant attributes that are Boolean valued.

We have introduced some ranking algorithms which basically are simplifications of a well-known greedy algorithm for the inference of relevant attributes. We have extended a negative approximability result to the restriction of only Boolean values and have improved a lower bound by using Feige's result. We have introduced some statistical values (depending on the considered concept and probability distribution) in terms of which we have stated general criteria for the successfulness of our algorithms. We have applied these results to a series of typical input distributions and specific functions.

Next we have investigated the case of noisy attribute values. We have shown that our algorithms still succeed with high probability, even if their input contains a (small) constant fraction of wrong values.

The ranking strategies discussed here turn out to be quite efficient on the average as long as the assumption of independence between different attributes is fulfilled. More important, they are very robust against noisy input data, and do not require specific knowledge about the likelihood of errors. Of course, one would like to extend these results to more complex Boolean functions like DNF formulas with a constant number of monomials. Furthermore, the case of robustly inferring relevant attributes of parity functions remains open.

Given a correct input instance, GREEDY RANKING always outputs a proper solution that is capable of explaining the sample. However, if the input has some disturbances, a matching solution might be much more complex measured in the number of variables to select than the solution of the undisturbed instance. Given the bound d, this is no issue for TARGET RANKING while GREEDY RANKING might indeed stop after having chosen significantly more than d attributes. Also in other noisy situations one might be interested in algorithms that – rather than computing an exact solution for the given input data – output a *simple solution* fitting to an input instance that is in some sense 'near' to the given input instance. Following Occam's razor such a simple solution may be much more likely to explain the real phenomenon. Currently, we are working on a general framework for this setting.

References

[1] N. Alon, J. Spencer, and P. Erdős, *The Probabilistic Method*. Wiley Intersc. Series, John Wiley & Sons, New York, 1992.

[2] R. Agrawal, T. Imielinski, and A. Swami, Mining Association Rules between Sets of Items in Large Databases. In Proc. *1993 ACM SIGMOD Conference* (1993), 207–216.

[3] T. Akutsu, F. Bao, Approximating Minimum Keys and Optimal Substructure Screens. In Proc. *2. COCOON*, Springer LNCS 1090 (1996), 290–299.

[4] T. Akutsu, S. Miyano, and S. Kuhara, A Simple Greedy Algorithm for Finding Functional Relations: Efficient Implementation and Average Case Analysis. In Proc. *3. DS*, Springer LNAI 1967 (2000), 86–98.

[5] D. Angluin, L. Valiant, Fast Probabilistic Algorithms for Hamiltonian Circuits and Matchings, *J. CSS 19* (1979), 155–193.

[6] S. Arora, L. Babai, J. Stern, and Z. Sweedyk, The Hardness of Approximate Optima in Lattices, Codes, and Systems of Linear Equations, *J. CSS 54* (1997), 317–331.

[7] A. Blum, L. Hellerstein, and N. Littlestone, Learning in the Presence of Finitely or Infinitely Many Irrelevant Attributes. In Proc. *4. COLT* (1991), 157-166.

[8] U. Feige, A Threshold of $\ln n$ for Approximating Set Cover. *J. ACM 45* (1998), 634–652.

[9] S. Goldman, H. Sloan, Can PAC Learning Algorithms Tolerate Random Attribute Noise? *Algorithmica 14* (1995), 70–84.

[10] D. Johnson, Approximation Algorithms for Combinatorial Problems. *J. CSS 9* (1974), 256–278.

[11] H. Mannila, K. Räihä, On the Complexity of Inferring Functional Dependencies. *Discrete Applied Mathematics 40* (1992), 237–243.

[12] L. Valiant, Projection Learning. In Proc. *11. COLT* (1998), 287–293.