Algorithmic Learning for Steganography: Proper Learning of k-term DNF Formulas from Positive Samples

Matthias Ernst[1,2], Maciej Liśkiewicz[1], and Rüdiger Reischuk[1(✉)]

[1] Institut für Theoretische Informatik, Universität zu Lübeck, Lübeck, Germany
{ernst,liskiewi,reischuk}@tcs.uni-luebeck.de
[2] Graduate School for Computing in Medicine and Life Sciences, Universität zu Lübeck, Lübeck, Germany

Abstract. Proper learning from positive samples is a basic ingredient for designing secure steganographic systems for unknown covertext channels. In addition, security requirements imply that the hypothesis should not contain false positives. We present such a learner for k-term DNF formulas for the uniform distribution and a generalization to q-bounded distributions. We briefly also describe how these results can be used to design a secure stegosystem.

1 Introduction

Digital steganography is a fairly new field of modern computer science concerned with camouflaging the presence of secret data in legal communications. In the general setting, a sender, often called Alice or the *steganographer* wishes to send a hidden message to a recipient via a public channel, which is completely monitored by an adversary called Warden or *steganalyst*. Taking a "typical" document Alice tries to embed a secret message in it such that a steganalyst cannot determine whether the secret message is present or not. In particular, Warden should have little chances to distinguish original documents, called *coverdocuments*, from altered ones called *stegodocuments*. This implies in general that the distributions of coverdocuments and stegodocuments have to be fairly close.

A crucial component when modeling steganography and steganalysis is the *knowledge* of the parties involved about coverdocuments. Considering different levels of knowledge, various models have been defined and studied. For example, if both the steganographer and the steganalyst have perfect knowledge about the distribution of coverdocuments and these documents satisfy certain conditions, secure steganography can be modeled and investigated by means of information and coding theory, whereas steganalysis can be done by applying statistical detection theory. But, though well-understood, such models are quite artificial

M. Ernst—This work was supported by the Graduate School for Computing in Medicine and Life Sciences funded by Germany's Excellence Initiative [DFG GSC 235/2].

K. Elbassioni and K. Makino (Eds.): ISAAC 2015, LNCS 9472, pp. 151–162, 2015.
DOI: 10.1007/978-3-662-48971-0_14

and far away from reality (for more discussion, see [9]). The other extreme is to assume that the steganographer a priori has no knowledge whatsoever about typical documents and can only get information using a sampling oracle. Even if the steganalyst has full knowledge assuming the existence of secure cryptographic one-way functions, provably secure steganography is possible [7], but *any* secure steganographic system requires an exponential number of samples with respect to the message length [4]. Thus, steganography becomes highly inefficient.

To be closer to the real world, newer approaches to steganalysis and steganography assume some reasonable partial knowledge about the type of covertext channel. Then steganalysis can be formulated as a binary classification problem and examined using methods from machine learning. This line of research has currently received much attention (see e.g. [6,10,17]). However, learning approaches to steganography have not been studied systematically so far.

As in real applications of steganography we assume that Alice knows that the coverdocument distribution belongs to some class of distributions – she can choose the media where to embed into. Besides that, she can only use a sampling oracle to get information about the actual coverdocument distribution. Then the steganographic encoding can be stated as a two-stage problem (for a formal definition of steganography see Sect. 4):

(1) Algorithmic learning of the concrete distribution of coverdocuments and
(2) Generating a stegodocument that encodes a given piece of message.

Hence, the essential difficulties in constructing efficient algorithms arise because of two reasons. First, a standard PAC approach to model this situation typically fails because of a fundamental difference: only positive samples are available. Second, algorithms for random generation of combinatorial objects from a given (typically uniform) distribution, see e.g. [8], cannot be applied directly since the generated objects have to encode given messages.

Most recently Liśkiewicz et al. [12] have obtained several promising results in generating stegodocuments. They have considered three families of coverdocument channels described by monomials, by decision trees (DTs), and by DNF formulas, respectively, assuming uniform distribution of documents. The learning complexity of the corresponding concept classes in the general case ranges from low up to high (assuming $RP \neq NP$). For these families of channels efficient generic algorithms have been constructed that for a given description of the coverdocuments, suitably manipulate the documents to embed secret messages, even against a steganalyst with full knowledge. This solves Problem (2) above and allows secure steganography assuming the coverdocument distributions can be learned *properly*, i.e. such that the learning algorithm outputs a monomial, resp. a DT, or a DNF expression as its hypothesis, when learning from positive data only.

Notice the importance of the proper learning here. For example, it is well known that k-term DNF formulas can be learned efficiently from positive samples with respect to k-CNF formulas, i.e. such that the learning algorithm outputs a k-CNF formula for the concept represented by an unknown k-term DNF. However,

such a k-CNF representation of coverdocuments is useless for stegodocuments generation, because one would have to find satisfying assignments for k-CNF formulas which cannot be done efficiently in general. Unlike monomials and k-CNF formulas, the problem whether DTs and DNF-formulas can be learned properly from positive samples in an efficient way, remains open even for simple probability distributions like the uniform one. This paper gives an affirmative answer to this question for k-term DNFs.

Learnability of k-term DNF: Known Results. For the notion of learnability, we loosely follow the PAC model. In the standard setting (i.e. with positive and negative samples) it is not feasible to learn k-term DNF formulas properly in a *distribution-free* sense for fixed $k \geq 2$ unless $RP = NP$. Learning k-term DNF concepts for $k \geq 4$ remains infeasible even if allowing as hypothesis $f(k)$-term DNF, for $f(k) \leq (2k - 3)$ [14]. For unrestricted DNF formulas, it is infeasible to learn with respect to DNF hypothesis, even if the number of terms in the hypotheses is arbitrary large [1]. Assuming that samples are drawn from specific distributions over the learning domain but still allowing positive and negative samples, the situation changes drastically. Flammini et al. [5] have shown that k-term DNF formulas are learnable (properly) in polynomial time using positive and negative samples drawn from *q-bounded* distributions (the ratio of the probabilities $D(x)/D(y)$ for elements in the support does not exceed q for some number $q \geq 1$). This class is a natural generalization of the uniform distribution.

If the number of terms of the DNFs may grow, from [19] we know that n-term DNF formulas over the uniform distribution can be learned using a polynomial number of samples in quasi-polynomial time. However, the hypothesis space has to be extended to $(n \cdot t)$-term DNF with t depending on the sample complexity.

Concerning steganographic applications one has to learn DNF formulas properly *and* from positive samples only. The next serious complication is to exclude false positives in order to achieve steganographic security. In the distribution free setting, this learning task can efficiently be mastered for 1-term DNF (monomials) [18]. But it becomes infeasible for k-term DNF, with $k \geq 2$, and log-term as well as for unrestricted DNF formulas [13]. There is a positive result for monotone DNF (MDNF) formulas over the uniform distribution. It is possible to learn log-term MDNF formulas from positive samples only [15]. The class of k-term MDNFs can even be learned over q-bounded distributions from positive samples [11,16]. Also, a method for positively learning 2-term DNF over q-bounded distributions is known [5]. Most recently De et al. [3] have shown that DNF formulas have efficient learning algorithms from uniformly distributed positive samples, but instead of a k-term DNF hypothesis the learner outputs a *sampler*. This model seems to be unsuitable for embedding secret messages efficiently, because it is unknown how coverdocuments can be modified to securely embed a given message without knowing an adequate k-term DNF hypothesis.

Our Contribution. The main result of this paper is an efficient learner without false positives for k-term DNF formulas from positive samples with hypothesis space identical to the concept class for arbitrary fixed k over q-bounded distributions. The major challenge already occurs for the uniform distribution: false

positives cannot be tolerated at all. Our solution works in two phases. The learner switches from k-term DNF to k-CNF representation in phase 1 and then back in the second phase. In more details, in the first phase k-term DNF formulas are learned using k-CNF formulas with very high accuracy and without false positives using a first sequence of positive samples.

In phase 2, we construct a set of *maximal monomials* that should cover most of the k-CNF formula generated. The number of candidates for these monomials could be extremely large. Thus, we have to design a mechanism to select a suitable subset. This subset will still contain many more than k monomials. Finally, we apply tests with a second sequence of positive samples to select a subset of size at most k as final hypothesis.

As a negative result, we show that it is impossible to learn unrestricted DNF formulas without false positives. For q-bounded distributions learning n-term DNF formulas requires an exponential number of positive samples regardless of the hypothesis space. An overview of the current state of knowledge concerning DNF learning is given in Table 1.

Table 1. Positive and negative (unless $RP = NP$) results for learning DNF formulas from positive samples over several distributions in polynomial time.

Concept class	Distribution-free	Uniform/q-bounded
1-term DNF (monomials)	yes [18]	yes [18]
2-term DNF	no [13]	yes [5]
k-term DNF	no [13]	yes (Theorem 1)
log-term DNF	no [14]	open
unrestricted DNF	no [14]	no (Theorem 2)

2 Preliminaries

Let us start with some basic definitions. In the following, n will always denote the number of variables and $\mathcal{X} = \{0,1\}^n$ the set of binary strings of length n. For a distribution D over $\mathcal{X}$ let $\mathtt{sp}(D) := \{x \in \mathcal{X} \mid D(x) > 0\}$ denote the support of D. For $q \geq 1$ such a distribution is called *q-bounded* if $\max\{D(x) \mid x \in \mathtt{sp}(D)\} \leq q \cdot \min\{D(x) \mid x \in \mathtt{sp}(D)\}$.

For a Boolean formula φ let $\mathtt{sat}(\varphi) := \{x \in \mathcal{X} \mid \varphi(x)\}$ denote the set of assignments that satisfy φ; $\mathtt{sat}(\varphi)$ will also be called the *support* of φ. A k-CNF formula ψ is given by a conjunction of clauses each containing at most k literals. We may assume that ψ does not contain tautological clauses (having a variable and its negation simultaneously). A k-term DNF formula φ is a disjunction of at most k monomials. φ is called *non-redundant* if it does not contain monomials M such that removing M from φ does not change $\mathtt{sat}(\varphi)$, in particular there are no

identical monomials (that means having the same set of literals) or *trivial* monomials with empty support (containing a variable and its negation). A monomial M will be called *shorter* than a monomial M' if it consists of less literals than M'; we call M *larger* than M' if $|\mathtt{sat}(M)| > |\mathtt{sat}(M')|$. In this paper we consider the family of concept classes $\{\mathtt{sat}(\varphi) \subseteq \mathcal{X} \mid \varphi \text{ is a } k\text{-term DNF formula}\}$ and proper learning of the classes from positive examples, i.e. we require that a learner seeing only satisfying assignments outputs a k-term DNF formula.

The reader is assumed to be familiar with the standard concepts of PAC theory (see e.g. [18]). Below we present only the definition of learnability of a concept C from positive examples. This can be modeled by the condition that the underlying distribution D on $\mathcal{X}$ fulfills $\mathtt{sp}(D) = C$. Allowing false positives makes the problem trivial because the hypothesis $H = \mathcal{X}$ would make errors $D(C \triangle H)$ with weight 0. We therefore define: *$\mathcal{A}$ learns $\mathcal{C}$ from positive samples without false positives* if for every pair (C, D) of a concept $C \in \mathcal{C}$ and distribution $D \in \mathcal{D}$ that fulfills $\mathtt{sp}(D) = C$ its hypothesis satisfies: $H \subseteq C$ and $\Pr[D(C \setminus H) \geq \varepsilon] \leq \delta$. A concept class $\mathcal{C}$ with a set $\mathcal{D}$ of q-bounded distributions can be learned efficiently if a learner exists with running time bounded by a polynomial in $(1/\varepsilon, 1/\delta, n, q)$.

3 Learning k-term DNF from Positive Samples

Flammini et al. [5] have presented a method for learning a k-term DNF formula φ for q-bounded distributions. In a first phase candidate monomials are generated from positive samples in such a way that all monomials of φ having enough assignments actually occur. But there are generally more, and some of these monomials may have assignments that do not belong to $\mathtt{sat}(\varphi)$. Therefore, in the second phase, combinations of at most k candidate monomials are tested against a set of positive and negative samples. If such a combination fulfills a specific error bound then it becomes the output. It has been shown that with high probability this yields an approximate hypothesis.

In the following we will develop a generalization of this method that is capable of positively learning k-term DNF formulas. The learner gets only positive samples and is not allowed to generate false positives.

Computing Maximal Monomials from CNF-Formulas. It is known how to learn a k-term DNF formula φ without false positives by using as hypothesis space k-CNF formulas. In this case $((2n)^{k+1} - \ln \delta)/\varepsilon$ positive samples are needed [2,14,18]. The learner starts with the conjunction of all possible non-tautological clauses of length at most k, of which there are at most $(2n)^{k+1}$. Then clauses not satisfied by positive samples are deleted.

Our first innovation will construct candidate monomials for φ by learning a k-CNF representation ψ for φ and extracting monomials from ψ afterwards. We choose monomials M with $\mathtt{sat}(M) \subseteq \mathtt{sat}(\psi)$ as large as possible. Generally, for $k \geq 3$ it is NP-hard to find a single satisfying assignment for a k-CNF formula. But here we already know a number of satisfying assignments, namely the positive samples used to create ψ. For this purpose, we define a criterion for potential candidate monomials generated from ψ and a sample $x \in \mathtt{sat}(\psi)$.

Definition 1. *Let ψ be a Boolean formula and $x \in \mathtt{sat}(\psi)$. A monomial M is* (ψ, x)-maximal *if $x \in \mathtt{sat}(M) \subseteq \mathtt{sat}(\psi)$ and there is no submonomial of M with this property (a submonomial is obtained by removing some literals from M).*

Algorithm 1 given below computes such maximal monomials. It starts with the monomial $M = 1$ and adds literals until $\mathtt{sat}(M) \subseteq \mathtt{sat}(\psi)$ is satisfied. We may assume that every clause of ψ does not contain any variable more than once.

Lemma 1. *For a k-CNF formula ψ and $x \in \mathtt{sat}(\psi)$ Algorithm 1 computes a (ψ, x)-maximal monomial. Its runtime is bounded by a polynomial $p_k(n)$. For every (ψ, x)-maximal monomial M there exists a sequence of literals selected in line 10 such that the algorithm outputs M.*

Input: k-CNF formula ψ without tautological clauses; assignment $x \in \mathtt{sat}(\psi)$
Output: some (ψ, x)-maximal monomial M

```
1  M ⟵ 1; remove every literal from ψ that is not satisfied by x;
2  while true do
3  |  foreach clause K in ψ do
4  |  |  if there is exactly one literal ℓ in K then
5  |  |  |  M ⟵ (M ∧ ℓ);
6  |  |  |  remove all clauses that contain ℓ from ψ;
7  |  |  end
8  |  end
9  |  if ψ is empty then return M;
10 |  select an arbitrary literal ℓ′ from ψ;
11 |  remove ℓ′ from every clause in ψ;
12 end
```

Algorithm 1. `MaxMonomial`(ψ, x)

The learner to be defined below needs several (ψ, x)-maximal monomials, but at most $2^k - 1$ many. To get them one could perform a depth-first search over those literals that are selected and then deleted from ψ until enough maximal monomials have been found. However, different choices may lead to the same monomial eventually. In order to be efficient we need a suitable mechanism to prune the search tree. Our strategy and its analysis are quite involved; therefore, the details will be presented in a full version of this paper.

Learning Candidate Monomials. Considering every maximal monomial for each positive sample used to learn the k-CNF formula ψ, one might get a very large set of monomials. Thus, a new idea is needed to handle such a situation. To obtain a bounded number of candidates to continue with we try to prune the set of maximal monomials without losing too many satisfying assignments. To this aim every monomial of the unknown k-term DNF formula φ that has a large support should become a candidate monomial. On the other hand, monomials with a small support might be removed without losing much accuracy.

Let us start by considering the number of maximal monomials in case the k-CNF formula ψ is equivalent to the unknown k-term DNF formula φ. In general

$\mathtt{sat}(\psi)$ may cover only parts of the satisfying region of a monomial in a scattered way. Hence, there could exist many (ψ, x)-maximal monomials.

Definition 2. *Let $\varphi = M_1 \vee \cdots \vee M_k$ be a non-redundant k-term DNF formula, $x \in \mathtt{sat}(\varphi)$, and $I = \{i_1, \ldots, i_p\} \subseteq \{1, \ldots, k\}$ be a non-empty set of indices. A monomial $M_{I,x}$ is called (φ, I, x)-maximal if it is (φ, x)-maximal and $\mathtt{sat}(M_{I,x}) \subseteq \mathtt{sat}(M_{i_1} \vee \cdots \vee M_{i_p})$ and after removing any M_{i_j} from the right side this inclusion fails.*

Lemma 2. *For fixed φ, I, and x, a (φ, I, x)-maximal monomial $M_{I,x}$ is unique. If $y \in \mathtt{sat}(M_{i_1} \vee \cdots \vee M_{i_p})$ has a maximal monomial $M_{I,y}$ then $M_{I,y} = M_{I,x}$.*

This implies that the number of different (φ, I, x)-maximal monomials over all $x \in \mathtt{sat}(\varphi)$ and nonempty $I \subseteq \{1, \ldots, k\}$ is bounded by $2^k - 1$. Next we will derive a bound on the number of satisfying assignments for those maximal monomials that intersect potentially scattered regions of φ.

Lemma 3. *Let $\varphi = M_1 \vee \cdots \vee M_k$ be a non-redundant k-term DNF formula with monomials M_i ordered by increasing length. For $d \in \mathbb{N}$ let $\varphi_d = M_1 \vee \cdots \vee M_u$ be composed of all M_i with $|\mathtt{sat}(M_i)| \geq 2^d$. For a Boolean formula χ_d with $\mathtt{sat}(\chi_d) \subseteq \mathtt{sat}(M_{u+1} \vee M_{u+2} \vee \cdots \vee M_k)$ define $\psi_d := \varphi_d \vee \chi_d$, $\mathcal{M}^{[d]} := \{M \mid M is a (\psi_d, x) - max.\ monom.\ for some x \in (\mathtt{sat}(\chi_d) \setminus \mathtt{sat}(\varphi_d))\}$, and $\xi_d := \bigvee_{M \in \mathcal{M}^{[d]}} M$. Then it holds $|\mathtt{sat}(\xi_d)| \leq 2^{d+k-1}$.*

These notions provide the foundation for the learner specified in Algorithm 2 giving the following result.

Theorem 1. *For constant k, Algorithm 2 learns k-term DNF formulas without false positives over q-bounded distributions in polynomial time with respect to $(1/\varepsilon, 1/\delta, n, q)$ by drawing no more positive samples than*

$$\sigma(\varepsilon, \delta, n, k, q) := \varepsilon^{-1}\, q\, k\, 2^{3k+1} \left((2n)^{k+1} + \ln(2/\delta)\right) + 48\varepsilon^{-2} \ln\left(2^{k^2+2}/\delta\right).$$

Correctness Proof. We first show a bound on how much monomials may overlap (their sat-regions have a nonempty intersection).

Lemma 4. *Let $\varphi = M_1 \vee \cdots \vee M_k$ be a non-redundant k-term DNF formula and φ_i equal φ without M_i. Then $|\mathtt{sat}(M_i) \setminus \mathtt{sat}(\varphi_i)| \geq |\mathtt{sat}(M_i)| \cdot 2^{-k+1}$.*

Next, let us estimate how well a k-CNF formula ψ can reconstruct the original monomials of the unknown k-term DNF φ.

Definition 3. *Let $g(\varphi, q, k) := q\, 2^k |\mathtt{sat}(\varphi)|$. For $\gamma > 0$ call a monomial M_i of φ γ-*large *if $|\mathtt{sat}(M_i)| \geq \gamma\, g(\varphi, q, k)$.*

Lemma 5. *Let $\varphi = M_1 \vee \cdots \vee M_k$ be a k-term DNF formula with monomials M_i and $\psi = K_1 \wedge \cdots \wedge K_p$ be a k-CNF formula with clauses K_j and $\mathtt{sat}(\psi) \subseteq \mathtt{sat}(\varphi)$. Let D be a q-bounded distribution with $\mathtt{sp}(D) = \mathtt{sat}(\varphi)$ and let $\gamma > 0$. If $D(\mathtt{sat}(\varphi) \setminus \mathtt{sat}(\psi)) < \gamma$ then for every γ-large M_i it holds $\mathtt{sat}(M_i) \subseteq \mathtt{sat}(\psi)$.*

Input: ε, δ, k, q, sampling oracle EX
Output: hypothesis φ'
$\varepsilon_1 \longleftarrow \varepsilon\, q^{-1}\, k^{-1}\, 2^{-(3k+1)}$;
$N_1 \longleftarrow \varepsilon_1^{-1}\, ((2n)^{k+1} + \ln(2/\delta))$;
draw N_1 samples $E = (e_1, \ldots, e_{N_1})$ using EX;
learn k-CNF formula ψ using samples in E;
$\mathcal{M} \longleftarrow \emptyset$;
for $j \longleftarrow 1$ **to** N_1 **do**
  let $\mathcal{M}_j$ denote all (ψ, e_j)-maximal monomials and $m_j := \min\{|\mathcal{M}_j|, 2^k - 1\}$;
  generate an arbitrary subset $\mathcal{M}'_j$ of $\mathcal{M}_j$ of size m_j;
  $\mathcal{M} \longleftarrow \mathcal{M} \cup \mathcal{M}'_j$;
end
reduce $\mathcal{M}$ to the $(2^k - 1)$-shortest monomials;
$N_2 \longleftarrow 48\, \varepsilon^{-2}\, \ln(2^{k^2+2}/\delta)$;
draw N_2 samples $S = (s_1, \ldots, s_{N_2})$ using EX;
foreach *subset* W *of* $\mathcal{M}$ *of size at most* k **do**
  $\varphi_W := \bigvee_{M \in W} M$;
  if φ_W *misclassifies less than* $3\varepsilon N_2/4$ *samples of* S **then return** $\varphi' := \varphi_W$;
end

Algorithm 2. Learn-k-Term-DNF$(\varepsilon, \delta, k, q, EX)$

Thus, if a CNF-formula ψ approximates a k-term DNF-formula φ quite well then every monomial of φ with large support is completely covered by ψ. Only monomials with small support may give rise to errors in the approximation.

Now we show that the set of candidate monomials $\mathcal{M}$ constructed by Algorithm 2 contains all large monomials.

Lemma 6. *Let* $\varphi = M_1 \vee \cdots \vee M_k$ *be a non-redundant* k*-term DNF formula. With probability at least* $1 - \delta/2$*, Algorithm 2 adds a monomial* M'_i*, with* $\mathtt{sat}(M'_i) \supseteq \mathtt{sat}(M_i)$*, to* $\mathcal{M}$ *for every* $(\varepsilon_1 2^{2k})$*-large* M_i*, where* $\varepsilon_1 = \varepsilon\, q^{-1}\, k^{-1}\, 2^{-(3k+1)}$.

Proof sketch. Let M_i be an $(\varepsilon_1 2^{2k})$-large monomial. Assume that the algorithm has learned a k-CNF formula ψ with $D(\mathtt{sat}(\varphi) \setminus \mathtt{sat}(\psi)) \leq \varepsilon_1$, which happens with probability at least $1 - \delta/2$. Then, using Lemmas 3, 4, and 5 one can show that the sample sequence E contains at least one element $e_j \in \mathtt{sat}(M_i)$, such that no (φ, e_j)-maximal monomial intersects with potential scattered regions of φ. Hence the number of (ψ, e_j)-maximal monomials can be bounded by Lemma 2 and some M'_i with $\mathtt{sat}(M'_i) \supseteq \mathtt{sat}(M_i)$ will be added to $\mathcal{M}$. All maximal monomials that intersect with scattered regions have less assignments than M_i by Lemmas 3 and 5. Thus M'_i is among the $2^k - 1$ shortest monomials in $\mathcal{M}$ by Lemma 2. □

From Lemma 6 one can conclude the correctness of Algorithm 2. The learning algorithm can be made applicable even if q is unknown (see [5]).

A Negative Result. Verbeurgt [19] has developed a method for learning poly(n)-term DNF over the uniform distribution from a polynomial number of

positive and negative samples with a quasi-polynomial running time. In contrast, we can show (proof omitted):

Theorem 2. *For every q-bounded distribution D and every hypothesis space $\mathcal{H}$, learning n-term DNF formulas without false positives requires an exponential number of positive samples drawn according to D for $\varepsilon < 1/q$.*

4 Learning Documents for Steganography

We start this section with a short review of basic definitions similar to [7]. Let $\mathcal{X}$ denote the set of cover- or stegodocuments. A channel $\mathcal{C}$ is a mapping with domain $\mathcal{X}^*$ that for every sequence h of documents, called a *history*, defines a probability distribution $\mathcal{C}_h$ on $\mathcal{X}$.

A *sampling oracle* for $\mathcal{C}$ takes a history h as input and returns a random element according to $\mathcal{C}_h$. In order to generate a typical sequence of coverdocuments $c_1, c_2, \ldots$ of $\mathcal{C}$ one starts with the empty history and asks the sampling oracle for a first element c_1, then with history $h_1 = c_1$ a second element c_2 is requested, and so on. $\mathcal{C}$ is called *supuniform* if for every h, $\mathcal{C}_h$ is the uniform distribution on $\mathtt{sp}(\mathcal{C}_h)$.

A *stegosystem* for $\mathcal{X}$ is a pair of polynomial-time bounded probabilistic algorithms $\mathcal{S} = [SE, SD]$ such that, for a security parameter κ,

(1) the encoder SE having access to a sampling oracle for a channel $\mathcal{C}$ gets as input a history h (elements that have already been generated by $\mathcal{C}$), a secret key $K \in \{0,1\}^\kappa$, and a message $\mu \in \{0,1\}^m$ and returns a sequence of stegodocuments $s_1, s_2, \ldots$ that should look like typical elements of $\mathcal{C}$ starting with history h (the length of this sequence may depend on κ and m).
(2) The decoder SD takes as input a secret key K and a sequence of documents S and returns a string $\mu \in \{0,1\}^m$.

The *unreliability* of $\mathcal{S} = [SE, SD]$ with respect to a channel $\mathcal{C}$ is given by

$$\mathrm{UnRel}_{\mathcal{S},\mathcal{C}} \;:=\; \max_{h,\mu\in\{0,1\}^m} \left\{\Pr_{K\in\{0,1\}^\kappa}[SD(K, SE(h,K,\mu)) \neq \mu]\right\}.$$

For security analysis we take as adversary a probabilistic machine W called a (t,ζ)-warden that can perform a chosen hiddentext attack:

- W can access a sampling oracle for the channel $\mathcal{C}$ that in the following will be called his *reference oracle*;
- W selects a history h and a message μ and queries a *challenge oracle* CH which is either $SE(h,K,\mu)$ or $\mathcal{C}(h,\mu)$, where $\mathcal{C}(h,\mu)$ returns a sequence of random elements of $\mathcal{C}$ with history h of the same length as $SE(h,\cdot,\mu)$;
- W runs in time t and can make up to ζ queries;
- with the help of the reference oracle $\mathcal{C}$ and the challenge oracle CH the warden $W^{\mathcal{C},CH}$ tries to distinguish stego- from coverdocuments.

His *advantage* over random guessing is defined as the difference

$$\mathrm{Adv}_{\mathcal{S},\mathcal{C}}(W) \;:=\; \left|\Pr_{K\in\{0,1\}^\kappa}\left[W^{\mathcal{C},SE(\cdot,K,\cdot)}=1\right] - \Pr\left[W^{\mathcal{C},\mathcal{C}(\cdot,\cdot)}=1\right]\right|.$$

For a given family $\mathcal{F}$ of channels $\mathcal{C}$ the strongest notion of security for a stegosystem $\mathcal{S}$ is defined as $\mathrm{InSec}_{\mathcal{S},\mathcal{F}}(t,\zeta) := \sup_{\mathcal{C}\in\mathcal{F}} \sup_W \mathrm{Adv}_{\mathcal{S},\mathcal{C}}(W)$, where W runs over all (t,ζ)-wardens. Thus, if $\mathrm{InSec}_{\mathcal{S},\mathcal{F}}$ is small then for every channel $\mathcal{C}$ of $\mathcal{F}$ no W – even those having perfect knowledge about $\mathcal{C}$ – can detect the usage of $\mathcal{S}$ with significant advantage.

Now let us consider channels $\mathcal{C}$ over the document space $\mathcal{X} = \{0,1\}^n$ such that for every history h the support of $\mathcal{C}_h$ can be described by a k-term DNF formula. These will be called *k-term DNF channels*. In [12] a polynomial-time bounded embedding algorithm has been constructed that for a given string $\omega \in \{0,1\}^b$, an arbitrary key K, and a k-term DNF formula φ with sufficiently large support (depending on b) generates a document $s \in \mathtt{sat}(\varphi)$ that encodes ω. The distribution of these stegodocuments is uniform over $\mathtt{sat}(\varphi)$ where the probability is taken over random choices of K and the internal randomization of the algorithm. Assuming that the underlying k-term DNF channel $\mathcal{C}$ is known exactly – this means for every h a k-term DNF formula for $\mathtt{sp}(\mathcal{C}_h)$ – one can use this embedding procedure to construct an efficient stegosystem $\hat{\mathcal{S}}$ for the family $\mathcal{F}$ of all supuniform k-term DNF channels $\mathcal{C}$. It has both small unreliability and small insecurity.

Definition 4. *For $\eta \geq 1$ and an integer $k \geq 1$ let $\mathcal{F}_{k,\eta}$ be the set of all supuniform k-term DNF channels $\mathcal{C}$ such that for every history h it holds $|\mathtt{sp}(\mathcal{C}_h)| \geq 2^\eta$.*

Let b denote the number of bits encoded per document and $m = \ell \cdot b$ the length of the secret message μ to be embedded. Combining the embedding technique of [12] with the results of the previous section we can show:

Theorem 3. *For the channel family $\mathcal{F}_{k,\eta}$ and given reliability parameters $\varepsilon, \delta > 0$ there exists a stegosystem $\mathcal{S}_k$ that for every $\mathcal{C} \in \mathcal{F}_{k,\eta}$ achieves the insecurity bound of $\hat{\mathcal{S}}$ and the unreliability bound* $\mathrm{UnRel}_{\mathcal{S}_k,\mathcal{C}} \leq 2\ell(\varepsilon+\delta) + 2em\left(k \cdot 2^{-\eta}/(1-\varepsilon)\right)^{(\log \mathrm{e})/b}$.

Trying to extend this result to q-bounded channels one faces the problem that the corresponding distributions are not efficiently learnable – their support can be learned, but not the individual probabilities which cannot even be specified in polynomial length in general. Thus, the stegoencoder cannot get complete knowledge about the channel and the same should hold for the steganalyst – otherwise he can easily detect any deviation from the channel distribution implying that secure and efficient steganography would be impossible. The analysis for this situation is given in a full version of this paper.

5 Conclusions

We have provided a polynomial-time algorithm for properly learning k-term DNF formulas from positive samples only. Further, we have shown that unrestricted DNF formulas cannot be learned from positive samples without false positives due to information theoretical reasons. Although the analogous learnability problem for log-term DNF formulas remains still open, the negative result

for unrestricted DNF formulas shows that this new method for learning k-term DNF formulas is quite powerful.

Combining our learning algorithm with the embedding procedure of [12] we are able to construct an efficient and provably secure stegosystem for a family of channels that can be defined by k-term DNF formulas. This illustrates that methods of algorithmic learning are important for steganography. Here, however, both learning *and* embedding components are crucial. As an example, the embedding problem for supports represented by efficiently learnable k-CNF formulas seems to be infeasible.

References

1. Alekhnovich, M., Braverman, M., Feldman, V., Klivans, A.R., Pitassi, T.: The complexity of properly learning simple concept classes. J. Comput. Syst. Sci. **74**(1), 16–34 (2008)
2. Blumer, A., Ehrenfeucht, A., Haussler, D., Warmuth, M.K.: Learnability and the Vapnik-Chervonenkis dimension. J. ACM **36**(4), 929–965 (1989)
3. De, A., Diakonikolas, I., Servedio, R.A.: Learning from satisfying assignments. In: Indyk, P. (ed.) Proc. SODA, pp. 478–497. SIAM, Philadelphia (2015)
4. Dedić, N., Itkis, G., Reyzin, L., Russell, S.: Upper and lower bounds on black-box steganography. J. Cryptology **22**(3), 365–394 (2009)
5. Flammini, M., Marchetti-Spaccamela, A., Kučera, L.: Learning DNF formulae under classes of probability distributions. In: Proc. COLT, pp. 85–92. ACM, New York (1992)
6. Fridrich, J.: Steganography in digital media: principles, algorithms, and applications. Cambridge University Press, New York (2009)
7. Hopper, N., von Ahn, L., Langford, J.: Provably secure steganography. IEEE T. Comput. **58**(5), 662–676 (2009)
8. Jerrum, M.R., Valiant, L.G., Vazirani, V.V.: Random generation of combinatorial structures from a uniform distribution. Theor. Comput. Sc. **43**, 169–188 (1986)
9. Ker, A.D., Bas, P., Böhme, R., Cogranne, R., Craver, S., Filler, T., Fridrich, J., Pevný, T.: Moving steganography and steganalysis from the laboratory into the real world. In: Proc. IH&MMSec, pp. 45–58. ACM, New York (2013)
10. Kodovsky, J., Fridrich, J., Holub, V.: Ensemble classifiers for steganalysis of digital media. IEEE T. Inform. Forensics and Sec. **7**(2), 432–444 (2012)
11. Kucera, L., Marchetti-Spaccamela, A., Protasi, M.: On learning monotone DNF formulae under uniform distributions. Inform. Comput. **110**(1), 84–95 (1994)
12. Liśkiewicz, M., Reischuk, R., Wölfel, U.: Grey-box steganography. Theor. Comput. Sc. **505**, 27–41 (2013)
13. Natarajan, B.K.: Probably approximate learning of sets and functions. SIAM J. Comput. **20**(2), 328–351 (1991)
14. Pitt, L., Valiant, L.G.: Computational limitations on learning from examples. J. ACM **35**(4), 965–984 (1988)
15. Sakai, Y., Maruoka, A.: Learning monotone log-term DNF formulas under the uniform distribution. Theory of Comput. Syst. **33**(1), 17–33 (2000)
16. Sakai, Y., Maruoka, A.: Learning k-term monotone boolean formulae. In: Doshita, S., Furukawa, K., Jantke, K.P., Nishida, T. (eds.) ALT 1992. LNCS, vol. 743, pp. 195–207. Springer, Heidelberg (1993)

17. Schaathun, H.G.: Machine Learning in Image Steganalysis. Wiley-IEEE Press, Chichester (2012)
18. Valiant, L.G.: A theory of the learnable. CACM **27**(11), 1134–1142 (1984)
19. Verbeurgt, K.: Learning DNF under the uniform distribution in quasi-polynomial time. In: Proc. COLT, pp. 314–326. Morgan Kaufmann Publishers Inc., San Francisco (1990)