符号理論の視点による 擬似ランダム構造の統一的理解

A Unified Framework for Understanding Pseudorandom Constructions
from a Coding Theoretic Perspective

安永憲司 Kenji YASUNAGA

アブストラクト ランダムに構成すると望ましい性質を持つオブジェクトは、多くの場合ランダムなままでは扱えないため、「擬似ランダム」に構成する必要がある。擬似ランダムオブジェクトは様々な存在し、それぞれの目的や性質は異なるが、幾つもの擬似ランダムオブジェクトは、統一的特徴付けが可能であることを Vadhan が示した。そのオブジェクトとは、誤り訂正符号、エクスパンドグラフ、標本器、乱数抽出器、困難性増幅器、擬似乱数生成である。本稿では、誤り訂正符号を基本的な視点として、その他のオブジェクトについて統一的理解することを目的とし、各オブジェクトの紹介と解説を行う。キーワード 擬似ランダム構造、誤り訂正符号、エクスパンドグラフ、標本器、乱数抽出器、困難性増幅器、擬似乱数生成器

1. 擬似ランダム構造

ランダムに行うことで望ましい結果が得られることはよくある。たくさんあるデータの平均値を知りたい場合、ランダムなサンプルに対して平均を取れば、実際の平均値に近い値が求まる。通信路で発生する誤りを訂正したい場合、送りたいメッセージに、ランダムな符号化で冗長性を加えれば、多くの誤りを訂正することができる。また、鍵を共有した状態で安全な秘匿通信をしたい場合、共有している鍵をランダムに符号化して長くすれば、共有した鍵よりも長いメッセージを安全に送ることができる。

しかし、ランダムに行うことが実質的に不可能な状況がある。例えば、通信路での誤りを訂正する場合、送信者と受信者の間で、どのようにメッセージを符号化するかをあらかじめ決めておく必要がある。そうでないと、受信者はどのメッセージが送られたかを特定することができない。そのため、実際にはランダムな符号化はできない。また、あらかじめランダムに選んだものを送受信者間で共有しておくことも可能である。しかし、その場合、必要な計算コストが膨大になり、現実的には実行できないことが多い。同様の問題は、安全に秘匿通信をする場合に、ランダムな符号化で鍵長を伸ばすときにも当てはまる。

そこで実際は「ランダムに見える」擬似ランダムな構造をもたせることで目的を達成することになる。このような、擬似ランダムな構造を持つオブジェクトは様々な研究分野で登場する。多くの場合、ランダムに構成すると望ましい性質を持つため、存

在性は確認できる。しかし、それをどのように構成すればよいかが問題である。望ましい性質を持つオブジェクトを明示的に構成することが大きな目標である。

それぞれの擬似ランダムオブジェクトは、別々の目的を持って誕生するものであるが、幾つかのオブジェクトは、非常に似た構造を持つことが近年の研究成果から明らかになってきた。そして、各オブジェクトを統一的な枠組みで特徴付ける方法が Vadhan により示された⁽¹⁾⁽²⁾。それぞれの擬似ランダムオブジェクトが持つ望ましい性質を、同じ枠組みの中で捉えているのである。そこで特徴付けられた擬似ランダムオブジェクトは、以下に挙げる六つである。

- 誤り訂正符号
- エクスパンドグラフ
- 標本器
- 乱数抽出器
- 困難性増幅器
- 擬似乱数生成器

これらは、情報理論、グラフ理論、計算の複雑さ理論、暗号理論など、様々な分野に及ぶものであり、統一的に表現できるという事実は大変興味深い。

そこで本稿では、Vadhan の統一理論に基づいて、各オブジェクトの紹介と解説を行う。特に、誤り訂正符号の視点から他のオブジェクトを眺めることで、他のオブジェクトがどのような機能を持ったものなのかを統一的に理解することを目的とする。なお、各オブジェクトに関する研究の背景及び発展については、文献(3)~(11)などを参照して頂きたい。

各オブジェクトについて解説した後、統一理論を利用した研究成果として、Guruswami, Umans, Vadhan⁽¹²⁾の研究成果を紹介する。この研究では、リードソロモン符号の一般化である Parvaresh-Vardy 符号を基にして、ほぼ最適な乱数抽出器の構成を与えている。既存の抽出器の構成が非常に複雑であったの

安永憲司 正員 東京工業大学大学院情報理工学研究科 数理・計算科学専攻
E-mail yasunaga@is.titech.ac.jp
Kenji YASUNAGA, Member (Department of Mathematical and Computing Sciences, Tokyo Institute of Technology, Tokyo, 152-8552 Japan).

電子情報通信学会 基礎・境界ソサイエティ
Fundamentals Review Vol.5 No.1 pp.56-67 2011 年 7 月
©電子情報通信学会 2011

に対し、簡素な構成法を与えている点が特徴である。この結果は、リードソロモン符号を起源とする代数的構造に基づく符号が、擬似ランダムオブジェクトとして、非常に強力な道具であることを示している。

以降では、2. にて、誤り訂正符号及び、Vadhan の統一理論で登場する、リスト復号可能な符号について解説する。3. では、Vadhan によって示された統一理論を紹介し、リスト復号可能な符号がどのように特徴付けられているかを見ていく。その後の章では、エクスパングラフ、標本器、乱数抽出器、困難性増幅器、擬似乱数生成器の順に、各オブジェクトが統一理論においてどのように特徴付けられているか、またリスト復号可能な符号とどのように対応しているか解説する。9. では、統一理論を利用した Guruswami, Umans, Vadhan の研究⁽¹²⁾を紹介する。

2. 誤り訂正符号のリスト復号

通信における誤り訂正とは、通信路で誤りが発生したとしても受信側でその誤りを訂正できるようにすることである。送りたいメッセージをそのまま送ってしまうと誤りを訂正することはできないため、メッセージに冗長性を加えることが誤り訂正の基本的な方法である。

自然数 N に対し、 $[N] = \{1, \dots, N\}$ と定義する。送りたいメッセージが N 個あったとき、それぞれのメッセージを別の系列に対応させることになる。対応させた系列のことを符号語と呼ぶ。符号語がアルファベット $[q]$ 上の長さ D の系列で表現されるとき、符号化は関数 $\text{Enc} : [N] \rightarrow [q]^D$ で定義される。つまり、メッセージ $x \in [N]$ に対応する符号語は $\text{Enc}(x)$ である。「符号」といった場合、この符号化関数を指すこともあれば、この関数の値域 $C = \{\text{Enc}(i) : i \in N\}$ を指すこともある。

符号に含まれる各符号語間のハミング距離が大きければ、誤り訂正能力は高いといえる。誤りが発生したとき、各符号語間の距離が大きければ、別の符号語を送ったと間違ってしまう可能性が低いからである。つまり、訂正能力の高い符号を得るには、 N 個の符号語を、空間 $[q]^D$ の中にバランス良く配置すればよい。この配置をランダムに行えば、高い確率でバランスの良い配置になる。すなわち、訂正能力の高い誤り訂正符号は、擬似ランダムオブジェクトなのである。

通信路で発生した誤りを含んだ符号語（受信語と呼ぶ）から、元の符号語を見つけることを復号と呼ぶ。通常の復号では、受信語から最も距離の近い符号語を見つけることを問題とする。ただし、このような復号には限界があり、符号の各符号語間の距離の最小値（最小距離と呼ぶ）が d だとすると、最悪ケースを考えると、 $d/2$ 個の誤りが発生すると、誤りの訂正を確実に行うことはできなくなる。現実世界における応用を考えると、 $d/2$ 個程度の誤りを訂正すれば十分なことが多く、 $d/2$ 個未満の誤りをいかに高速に復号するかが重要である。しかし、 $d/2$ 個以上の誤りが発生する場合でも、復号をあきらめるのではなく、何らかの誤り訂正をしたいと考える。そこで出てきたのがリスト復号という概念である。

リスト復号では、入力として受信語と距離のパラメータが与えられ、受信語からその距離以下の全ての符号語を出力することを問題とする。出力する符号語が一つである必要はないため、 $d/2$ 個以上の誤りが発生した場合でも送信した符号語を（リストとして）復元することができる。出力されたリストをどのように扱うかは、応用次第といえる。受信語から最小の距離にある符号語を見つけたい場合は、リストの中からそのような符号語を見つければよい。理論的な応用の場合、送信符号語がリストに含まれていれば十分という状況もある。

関数 $\text{Enc} : [N] \rightarrow [q]^D$ で定義される符号のリスト復号可能性を、次のように定義する。

[定義 1] 符号 $\text{Enc} : [N] \rightarrow [q]^D$ が (ϵ, K) リスト復号可能であるとは、任意の受信語 $r \in [q]^D$ に対して、 r と $\text{Enc}(x)$ 間でシンボルが一致する割合が $1/q + \epsilon$ を超えるようなメッセージ $x \in [N]$ の数が K 以下のときである。

この定義を、誤り訂正という観点から説明すると、符号が (ϵ, K) リスト復号可能であるとは、符号語 $\text{Enc}(x)$ を送信したとき、 $1 - (1/q + \epsilon)$ の割合の誤りが発生したとしても、最大 K 個の送信符号語候補を見つけることができ、その中に $\text{Enc}(x)$ が含まれているという意味である。受信語と符号語の一致する割合が $1/q + \epsilon$ 以上の場合を考えているが、一致する割合がこれ以上小さい場合は意味のあるリスト復号ができないという点に注意したい。なぜならば、受信語空間 $[q]^D$ から一様ランダムに選んだ系列は、高い確率で、ほとんどの符号語と $1/q$ の割合で一致しているからである。つまり、一致する割合が $1/q$ 未満のとき、リストにほとんど全ての符号語を含める必要が出てきてしまうのである。

リスト復号という概念自体は、Elias⁽¹³⁾と Wozencraft⁽¹⁴⁾によって、1950 年代に導入されたが、効率的な復号アルゴリズムが登場するのは、そのおよそ 30 年後である。1989 年に、Goldreich と Levin⁽¹⁵⁾は、アダマール符号に対する効率的なリスト復号アルゴリズムを示した。しかし、彼らの研究は計算の複雑さ理論・暗号理論の文脈で登場しており、また、誤り訂正問題としては符号理論で通常考えるものとは異なっていたため^(注1)、情報理論・符号理論への影響は大きくならなかった。その後、1997 年に、Sudan⁽¹⁶⁾が、符号理論で通常考える誤り訂正問題として、リードソロモン符号に対して効率的なリスト復号アルゴリズムを提案し、符号理論においてもリスト復号の研究が活発に行われるようになった。

3. Vadhan による統一理論

Vadhan は、擬似ランダムオブジェクトを統一的に特徴付けるための枠組みを与えた。登場するのは関数 Γ とそれに対して定義される関数 LIST_Γ である。少し抽象的ではあるが、まず

(注1): Goldreich-Levin⁽¹⁵⁾では、アダマール符号に対して、メッセージ長の多項式時間で実行するリスト復号アルゴリズムを提案している。しかし、符号理論で通常考える誤り訂正問題では、符号長の多項式時間がどうかを気にする。アダマール符号は、メッセージ長 n に対して、符号長が 2^n であるため、符号長の多項式時間アルゴリズムで十分ならば、全ての符号語を列挙してリストに入れるかどうかを判断するという単純な復号法で十分なのである。

その定義を与える．

[定義 2] 関数 $\Gamma: [N] \times [D] \rightarrow [M]$ と集合 $T \subseteq [M]$ ，一致パラメータ $\epsilon \in [0, 1]$ に対して，

$$\text{LIST}_{\Gamma}(T, \epsilon) = \{x \in [N] : \Pr[\Gamma(x, U_{[D]}) \in T] > \epsilon\}$$

と定義する．ただし， U_X は集合 X 上の一様分布に従う確率変数である．また，

$$\text{LIST}_{\Gamma}(T, 1) = \{x \in [N] : \Pr[\Gamma(x, U_{[D]}) \in T] = 1\}$$

と定義する．

Vadhan の統一理論では，関数 Γ によって擬似ランダムオブジェクトを定義し，関数 LIST_{Γ} を用いてその性質を特徴付ける．その特徴付けの方法も，どのオブジェクトもほぼ同じような形で行うことができ，全ての $T \in \mathcal{C}$ に対して， $|\text{LIST}_{\Gamma}(T, \epsilon)| \leq K$ である」という形式をとる．

そこでまず，前節で定義したリスト復号可能な符号を，この枠組みによって特徴付ける．符号化関数 $\text{Enc}: [N] \rightarrow [q]^D$ から関数 $\Gamma: [N] \times [D] \rightarrow [D] \times [q]$ を以下のように定義する．

$$\Gamma(x, y) = (y, \text{Enc}(x)_y).$$

ここで， $\text{Enc}(x)_y$ は，符号語 $\text{Enc}(x)$ の y 番目のシンボルを表している．このとき，リスト復号可能な符号は次のように特徴付けられる．

[命題 3] 符号 Enc が (ϵ, K) リスト復号可能であるための必要十分条件は，

$$\forall r \in [q]^D, \quad |\text{LIST}_{\Gamma}(T_r, 1/q + \epsilon)| \leq K$$

である．ここで， $T_r = \{(y, r_y) : y \in [D]\}$ である．

この命題について簡単に説明しよう．任意の $r \in [q]^D$ というのは受信語を表しており， T_r は r をインデックス別に集合 $T_r = \{(1, r_1), (2, r_2), \dots, (D, r_D)\}$ と表したものである．そして， $\text{LIST}_{\Gamma}(T_r, 1/q + \epsilon)$ というのは，定義から

$$\begin{aligned} & \text{LIST}_{\Gamma}(T_r, 1/q + \epsilon) \\ &= \{x \in [N] : \Pr[\Gamma(x, U_{[D]}) \in T_r] > 1/q + \epsilon\} \\ &= \{x \in [N] : \Pr_{y \leftarrow \frac{R}{[D]}} [(y, \text{Enc}(x)_y) \in T_r] > 1/q + \epsilon\} \\ &= \{x \in [N] : \Pr_{y \leftarrow \frac{R}{[D]}} [\text{Enc}(x)_y = r_y] > 1/q + \epsilon\} \end{aligned}$$

と書き直すことができる．ここで， $s \xleftarrow{R} S$ とは，集合 S から一様ランダムに s という要素をサンプルすることを表す．最後の式は，受信語 r と一致する割合が $1/q + \epsilon$ を超えるメッセージ x の集合を表している．そのサイズが K 以下という条件は，まさに符号が (K, ϵ) リスト復号可能であることを言い換えているのである．

以降では，その他の擬似ランダムオブジェクトをこの枠組みを用いて特徴付ける．同じ枠組みを使うことで，擬似ランダムオブジェクトが非常に似た，場合によっては等価な関係にあることが分かる．一方で，この枠組みで登場するパラメータ N, M, D, q, ϵ, K については，各オブジェクト間で，興味のある

範囲が異なってくる．

そこで，リスト復号可能な符号では，どのようなパラメータに興味があるかを確認しておこう．符号にとって基本となるパラメータはメッセージ長 $n = \log N$ ，若しくは符号長 $D_q = D \log q$ である．様々な n, D_q に対して符号を定義することを考える．計算の効率を考える場合は， n や D_q をパラメータとしたときの効率を考える．一致パラメータ ϵ は，小さいほどリスト復号としての訂正能力が上がるため，小さいことが望ましい．また，符号化レートと呼ばれる，符号長に対するメッセージ長の割合 $\rho = n/D_q$ は，符号化によって生じる冗長性の少なさを表しており，大きいことが望ましい．アルファベットサイズ q は固定された定数であることが望ましいが， n に従って増加する場合も考えられる．つまり，メッセージ長 n を大きくし，アルファベットサイズ q を小さく抑えながら， ϵ や ρ として固定した定数を達成することを考えている．リストサイズ K は，効率的な復号を考えるならば， n や D_q の多項式で表現されるサイズである必要がある．一つの目標は， ϵ と ρ の最適なトレードオフを達成し，かつ効率的なリスト復号が可能な符号を見つけることである．

最後に，この統一理論の枠組みでは，符号を構成するためのアルゴリズムやリスト復号のためのアルゴリズム，及びそれらの効率については言及していないことに注意しよう．あくまでも，リスト復号の可能性を特徴付けているのである．その理由としては，アルゴリズムの計算量を含めた統一的な枠組みを提供することも可能かもしれないが，各オブジェクトごとに興味のあるパラメータ，または興味のあるアルゴリズムが異なるため，統一的に扱うと枠組みが複雑になってしまうからである．オブジェクトを構成するための計算量に関して，リスト復号可能な符号では，メッセージ長 $n = \log N$ や符号長 $D_q = D \log q$ の多項式時間で動作するかどうかで評価する．一方，エクスパンドグラフでは， N や M の多項式時間で動作するかどうかに興味がある．また，乱数抽出器では，関数 Γ を効率的に計算することに興味があるが，リスト復号に対応するようなアルゴリズムを考える必要はない．このように，アルゴリズムやその効率に関しては，各オブジェクトごとに興味のある部分が大きく異なっているのである．

4. エクスパンドグラフ

統一理論の枠組みを視覚的に理解するため，次にグラフに対する特徴付けであるエクスパンドグラフを紹介する．

エクスパンドグラフとは，各頂点の次数が低い（つまり，各頂点から出る辺の数が少ない）にも関わらず，頂点集合が高い連結性を持つグラフのことである．頂点集合の連結性が高いとは，どの頂点集合も，その他の多くの頂点と連結しているという意味である．これを言い換えると，どの頂点集合を考えても，その集合が隣接する頂点集合のサイズが大きくなっている．つまり，隣接する頂点集合のサイズが拡大（エクスパンド）しているのである．

ここでは，左頂点に関して正則な，2 部グラフを考える．2 部

図 1 エクスパンダグラフ

グラフの場合，エクスパンダである条件は，あるサイズ以下の左頂点集合は，多くの右頂点集合と隣接しなければならないというものである．

[定義 4] 左頂点集合が $[N]$ ，右頂点集合が $[M]$ ，左頂点次数が D である 2 部グラフ G を考える． G が $(=K, A)$ エクスパンダであるとは，サイズ K の全ての左頂点集合 $S \subseteq [N]$ が，サイズ $A \cdot K$ 以上の隣接頂点集合を持つときである．また， $K' \leq K$ である全ての K' に対して $(=K', A)$ エクスパンダであるとき， G を (K, A) エクスパンダという．

図 1 はエクスパンダグラフの例である．このグラフは，パラメータが $N = 8, M = 6, D = 3$ である．サイズ 2 の任意の左頂点集合が，サイズ 4 以上の隣接頂点集合を持つため， $(2, 2)$ エクスパンダである．

左頂点が正則な 2 部グラフ G に対して，関数 $\Gamma: [N] \times [D] \rightarrow [M]$ を次のように定義する．

$$\Gamma(x, y) = \text{「頂点 } x \text{ の } y \text{ 番目の隣接頂点」}$$

このとき，以下が成り立つ．

[命題 5] 左頂点が正則な 2 部グラフ G が $(=K, A)$ エクスパンダであるための必要十分条件は

$$\forall T \subseteq [M] \text{ s.t. } |T| < AK, \quad |\text{LIST}_\Gamma(T, 1)| < K \quad (1)$$

である．したがって， G が (K, A) エクスパンダであるための必要十分条件は，サイズ AK 未満の任意の $T \subseteq [M]$ に対して， $|\text{LIST}_\Gamma(T, 1)| \leq |T|/A$ を満たすことである．

この命題について説明する．エクスパンダであるための条件は，次のように言い換えができることに着目しよう．グラフ G が $(=K, A)$ エクスパンダであるとは，サイズ $A \cdot K$ 未満の任意の右頂点集合 $T \subseteq [M]$ に対して，左頂点のうち，全ての隣接頂点が T に含まれているものの数が K 未満のときである．そしてこれは，条件 (1) と等価であることが分かる．なぜならば，

$$\begin{aligned} & |\text{LIST}_\Gamma(T, 1)| \\ &= |\{x \in [N] : \Pr[\Gamma(x, U_{[D]}) \in T] = 1\}| \\ &= |\{x \in [N] : \\ &\quad \Pr[x \text{ の全ての隣接頂点が } T \text{ に含まれる}] = 1\}| \\ &= |\{x \in [N] : x \text{ の全ての隣接頂点が } T \text{ に含まれる}\}| \end{aligned}$$

と書き換えできるためである．

表 1 エクスパンダグラフとの対応関係

リスト復号可能符号	エクスパンダグラフ
メッセージ	左頂点
符号語	左頂点から出る辺
受信語	一定サイズ以下の右頂点集合 T
リストに含まれるメッセージ	T に全ての辺が含まれる左頂点
リストサイズ	$ T /(\text{拡大係数})$

リスト復号可能符号と比べて異なるのは，条件に出てくる集合 T のとり方である．符号の場合， $T_r = \{(y, r_y)\}$ という形の，インデックス付けされた，一定サイズの集合を考えていた．一方，エクスパンダグラフでは，一定サイズ以下の全ての T を考える必要がある．また，パラメータ ϵ が登場しないことも異なる．

計算の効率を考える場合，リスト復号可能符号では，メッセージ長 $n = \log N$ や符号長 $D_q = D \log q$ をパラメータとして考えていたが，エクスパンダグラフの場合， N や M をパラメータとして効率を考える．計算の尺度が大きく異なっている点に注意しよう．その N と M は，互いに多項式で表現できる程度のサイズを考える．古典的には $N = M$ を考える場合が多い．左頂点の次数 D は，定数若しくは $\text{poly}(\log N)$ 程度を考える．パラメータ ϵ は，エクスパンダグラフでは登場しない．その一方で，エクスパンダグラフでは，拡大係数と呼ばれるパラメータ A に興味がある．この A が大きいほど，エクスパンダとしての性質が優れているからである．そして， $A > 1$ を達成するには， $|T| > K$ を満たす必要がある．符号で考えると，符号長 D の方がリストサイズ K よりも大きい場合に相当する．拡大係数は，次数 D より大きくはできないため， A を D に近づけることが一つの目標である．また， $M = N$ の場合， A を定数にして， $K = \Omega(N)$ を達成することが目標となる．

表 1 では，リスト復号可能符号の概念が，エクスパンダグラフにおいてどのように対応しているかをまとめている．

5. 標 本 器

ブール関数 $T: [M] \rightarrow \{0, 1\}$ を考える．今，関数 T の平均値を見積りたいとする．もちろん M 個ある $T(z), z \in [M]$ の値を全て計算すれば正確に求まるが，もっと効率的に見積りたい．どのようにすればよいだろうか．一つの方法は， T にランダムな入力を幾つか与え，その結果に対して平均値をとることで， T の平均値を見積もるという方法である．この方法による見積りの精度は，チャーノフ限界で説明することができる．チャーノフ限界とは，互いに独立な $\{0, 1\}$ 上の確率変数 $X_1, \dots, X_n$ に対して， $X = \sum_{i \in [n]} X_i$ としたとき，任意の $t > 0$ について，

$$\Pr[X > E[X] + t] \leq e^{-2t^2/n}$$

が成り立つというものである．これは， n 回サンプルしたときの和が，実際の平均値から t よりも大きく離れてしまう確率を上から抑えている．関数 T の平均値を見積もるために， $[M]$ から D 個のランダムサンプル $z_1, \dots, z_D$ をとったとする．このとき，そこから計算した平均値が実際の値から ϵ よりも大きく

それしてしまう確率は、

$$\begin{aligned} & \Pr\left[\frac{1}{D} \sum_{i \in [D]} T(z_i) > E[T(U_{[M]})] + \epsilon\right] \\ &= \Pr\left[\sum_{i \in [D]} T(z_i) > E[T(U_{[M]})] \cdot D + \epsilon \cdot D\right] \\ &\leq e^{-2(\epsilon D)^2/D} = e^{-2\epsilon^2 D} \end{aligned}$$

と抑えられる． $D = O(\log(1/\delta)/\epsilon^2)$ とすれば、この確率は δ 以下に抑えられる．

しかし、平均値の見積りのためには、もっと効率的な方法が存在している．上記の方法は、全て独立にサンプルした入力に対して平均値を評価しているが、入力値の間にある程度の相関があったとしても、平均値の見積りはできることが知られている．相関のある入力を使うことで、サンプルに必要なランダムビットの数を削減することができる．

標本器とは、このような、平均を見積もるための標本を生成するアルゴリズムのことである．

[定義 6] 決定性アルゴリズム Smp は、ランダム系列 $x \in [N]$ を受け取り、標本 $z_1, \dots, z_D \in [M]$ を出力する．Smp が (δ, ϵ) 平均化標本器であるとは、任意の関数 $T : [M] \rightarrow \{0, 1\}$ に対して、

$$\Pr\left[\frac{1}{D} \sum_{i \in [D]} T(z_i) \leq E[T(U_{[M]})] + \epsilon\right] \geq 1 - \delta,$$

を満たすときである．

つまり、生成された標本で平均値を見積もったとき、実際の平均値との誤差が ϵ 以内である確率が $1 - \delta$ 以上であることを保証している．

標本器 Smp に対して、関数 $\Gamma : [N] \times [D] \rightarrow [M]$ を次のように定義する．

$$\Gamma(x, y) = \text{「ランダム系列 } x \text{ における Smp の } y \text{ 番目の標本」}.$$

このとき、以下が成り立つ．

[命題 7] Smp が (δ, ϵ) 平均化標本器であるための必要十分条件は、

$$\forall T \subseteq [M], \quad |\text{LIST}_\Gamma(T, |T|/M + \epsilon)| \leq K \quad (2)$$

である．ここで、 $K = \delta N$ である．

それでは、この命題について説明する．関数 $T : [M] \rightarrow \{0, 1\}$ は、以下の対応関係によって集合 $T \subseteq [M]$ として表現されると考える．

$$T(z) = 1 \iff z \in T.$$

以降、 T は関数としても集合としても扱うことにする．求めたい平均値は $E[T(U_{[M]})] = |T|/M$ と表すことができる．このとき、定義より、

$$\begin{aligned} & \text{LIST}_\Gamma(T, |T|/M + \epsilon) \\ &= \{x \in [N] : \Pr[\Gamma(x, U_{[D]}) \in T] > |T|/M + \epsilon\} \\ &= \{x \in [N] : \end{aligned}$$

表 2 平均化標本器との対応関係

リスト復号可能符号	平均化標本器
メッセージ	入力ランダム系列
符号語	出力標本集合
受信語	平均値を見積もるブール関数
リストに含まれるメッセージ	見積りに失敗する入力系列
リストサイズ	見積りに失敗する入力系列の数

$$\begin{aligned} & \Pr[\text{Smp}(x) \text{ の標本が } T \text{ に含まれる}] > |T|/M + \epsilon \\ &= \{x \in [N] : \end{aligned}$$

$$\begin{aligned} & (\text{Smp}(x) \text{ の標本による } T \text{ の平均値}) > |T|/M + \epsilon \\ &= \left\{x \in [N] : \frac{1}{D} \sum_{i \in [D]} T(z_i) > E[T(U_{[M]})] + \epsilon\right\} \end{aligned}$$

と変形できる．これは、実際の平均値から ϵ 以内の誤差で見積もることに失敗するような標本を出力する、入力ランダム系列 x の集合である．したがって、条件 (2) は、どのような T に対しても、見積りに失敗する入力系列の数が $K = \delta N$ 以下であるということを述べており、Smp が (δ, ϵ) 平均化標本器であるための条件に一致していることが分かる．

リスト復号可能符号と比べると、エクスパンドグラフ同様、集合 T のとり方が異なる．標本器では、任意の $T \subseteq [M]$ を考える必要がある．そして、標本器の場合、基本となるパラメータは $m = \log M$ であり、様々な m に対しての標本器を考える．サンプルを生成するのに必要なランダムビットの数を削減することが重要であるため、 $n = \log N$ は m の線形若しくは多項式サイズであることが望まれる．符号の場合、 $M = Dq$ が n の多項式程度でも許されているのとは対照的である．そのほかにも、標本器の場合、失敗確率 δ は定数でも許されるのに対して、符号で $\delta = K/N$ を定数とすると、リストサイズ K がメッセージ空間サイズ N の線形になり、リストサイズとしては大きすぎる．

リスト復号可能符号と平均化標本器の対応は、表 2 のとおりである．

6. 乱数抽出器

乱数抽出器とは、相関や偏りのある系列を入力とし、一様乱数に近い系列を出力する関数のことである．元々は、物理的なランダム情報源を使って確率的アルゴリズムをシミュレートすることを目的にして提案された概念であるが、暗号理論などの理論計算機科学において幅広く応用されている．

入力される情報源がある程度のランダムネスを持っているようなときに、そこから乱数を取り出したいと考える．そこで、入力情報源がある条件を満たしているならば、乱数抽出ができるという形式にしたい．入力情報源の条件については様々なものが提案されているが、ここでは最小エントロピーという概念で保証される情報源に対する乱数抽出器を考える．

確率変数 X の最小エントロピーとは、次のように定義される．

$$H_\infty(X) = \min_{x \in \text{Supp}(X)} \log \frac{1}{\Pr[X = x]}.$$

ここで、 $\text{Supp}(X) = \{x : \Pr[X = x] > 0\}$ である． X の

最小エントロピーが k 以上であるとは、全ての x に対して、 $\Pr[X = x] \leq 2^{-k}$ のときである。つまり、情報源のどの要素もある一定確率以下でしか出現しないとき、最小エントロピーが保証される。最小エントロピーが大きいほど、一様乱数に近く、 $k = \log |X|$ のとき、 X は一様分布である。情報源に対する条件としては、理解しやすい基準である。

乱数抽出器の出力は一様乱数に近いことが求められるが、この「近さ」は、統計的距離と呼ばれる距離で測る。標本空間 $[M]$ の確率変数 X, Y の間の統計的距離とは、次のように定義される。

$$\Delta(X, Y) = \max_{T \subseteq [M]} |\Pr[X \in T] - \Pr[Y \in T]|.$$

ここで、 T は、 X と Y を識別するための一つの検査を表していると考えられる。その検査とは、標本空間 $[M]$ の部分集合 T に含まれる要素が出現する確率を調べることで、入力した分布が X と Y のどちらであるかを識別しようとするものである。その差は、 $|\Pr[X \in T] - \Pr[Y \in T]|$ であり、これが大きいとき、 X と Y は識別しやすい、つまり、距離が大きいと考える。そして、全ての検査 T を考えたときの差の最大値が統計的距離である。もし統計的距離が δ であるならば、 $|\Pr[X \in T'] - \Pr[Y \in T']| = \delta$ となる検査 T' が存在することを保証しているともいえる。

乱数抽出器としては、最小エントロピーがある値以上のどのような情報源からでも抽出できるようにしたい。しかし、最小エントロピーだけが保証されている情報源からは 1 ビットの乱数でさえ抽出できないことが分かる。 $\{0, 1\}^n$ 上の情報源と任意の関数 $\text{Ext} : \{0, 1\}^n \rightarrow \{0, 1\}$ を考える。すると、 $\text{Ext}^{-1}(0)$ 上の一様分布若しくは $\text{Ext}^{-1}(1)$ 上の一様分布のいずれか一方は、最小エントロピーが $n-1$ であり、非常に大きい ($\{0, 1\}^n$ 上の一様分布は最小エントロピーが n)。しかし、この分布に対して関数 Ext で抽出を行うと、出力は定数 (0 若しくは 1) である。つまり、 n ビットの入力から 1 ビットの乱数を抽出するための関数 Ext をどのように選んだとしても、その関数に対して最小エントロピーは高いが、出力が定数となってしまう情報源が存在するのである。

このような不可能性を避けるため、入力として短い長さの一様乱数 (種と呼ばれる) を許すような抽出器が考えられるようになった。このようにすると、最小エントロピーによって保証された任意の情報源から乱数を抽出することができる。

[定義 8] 関数 $\text{Ext} : [N] \times [D] \rightarrow [M]$ が (k, ϵ) 乱数抽出器であるとは、標本空間 $[N]$ 、最小エントロピー k 以上の、任意の情報源 X に対して、 $\text{Ext}(X, U_{[D]})$ と $U_{[M]}$ との統計的距離が ϵ 以下のときである。

抽出器に入力される $U_{[D]}$ が種を表しており、種の長さ $d = \log D$ は、出力系列の長さ $m = \log M$ に比べ、短いことが望まれる。実際、 d は m の対数程度に抑えることが可能である。

それでは、乱数抽出器を統一理論の枠組みで特徴付ける。乱数抽出器の場合、関数 Γ は Ext がそのまま対応する。そして、以下の性質が成り立つ。

[命題 9] $\Gamma = \text{Ext} : [N] \times [D] \rightarrow [M]$, $K = 2^k$ とする。この

とき、

(1) Ext が (k, ϵ) 乱数抽出器であるとき、

$$\forall T \subseteq [M], |\text{LIST}_\Gamma(T, |T|/M + \epsilon)| < K. \quad (3)$$

(2) 逆に、条件 (3) が成り立つとき、 Ext は $(k + \log(1/\epsilon), 2\epsilon)$ 乱数抽出器である。

この命題について説明する。まず、条件 (3) について、

$$\begin{aligned} \text{LIST}_\Gamma(T, |T|/M + \epsilon) \\ &= \{x \in [N] : \Pr[\Gamma(x, U_{[D]}) \in T] > |T|/M + \epsilon\} \\ &= \{x \in [N] : \Pr[\text{Ext}(x, U_{[D]}) \in T] > |T|/M + \epsilon\} \end{aligned}$$

と変形できる。 T は検査を表しており、確率 $|T|/M$ は、標本空間 $[M]$ 上の一様分布を考えたとときに、 T の要素が出現する確率を表している。つまり、 $|\text{LIST}_\Gamma(T, |T|/M + \epsilon)|$ は、 T によって一様分布よりも ϵ だけ高い確率で出現するような、入力情報源の要素 x の数を表している。このような x は、検査 T をパスできない情報源の要素といえる。なぜなら、乱数抽出器の出力は、どのような検査に対しても一様分布との確率の差が ϵ 以下である必要があるためである。条件 (3) は、どのような検査 T に対しても、それをパスできない入力 x の数が K 未満という条件である。

命題の一つ目について説明する。条件 (3) を満たさないと仮定する。このとき、ある検査 T に対して、それをパスできない入力 x の数が K 以上であることを意味する。そのような入力 x の集合上の一様分布 X を考えると、 X の最小エントロピーは $\log K = k$ 以上である。しかし、入力情報源が X のとき、検査 T を使うと、一様分布と確率 ϵ より大きな差が出る。これは Ext が (k, ϵ) 抽出器でないことを意味する。

次に命題の二つ目について説明する。最小エントロピー $(k + \log(1/\epsilon))$ の情報源 X を考える。このとき、任意の検査 T に対して、 $\Pr[\text{Ext}(X, U_{[D]}) \in T] \leq |T|/M + 2\epsilon$ を示す必要がある。実際に、

$$\begin{aligned} \Pr[\text{Ext}(X, U_{[D]}) \in T] \\ &\leq \Pr[X \in \text{LIST}_\Gamma(T, |T|/M + \epsilon)] \\ &\quad + \Pr[\text{Ext}(X, U_{[D]}) \in T \mid X \notin \text{LIST}_\Gamma(T, |T|/M + \epsilon)] \\ &\leq |\text{LIST}_\Gamma(T, |T|/M + \epsilon)| \cdot 2^{-(k + \log(1/\epsilon))} + |T|/M + \epsilon \\ &\leq K \cdot 2^{-(k + \log(1/\epsilon))} + |T|/M + \epsilon \\ &= |T|/M + 2\epsilon \end{aligned}$$

である。

乱数抽出器では、入力情報源からなるべく多くの乱数を抽出することが望ましいため、出力系列の長さ $m = \log M$ を k に近づけることが一つの目標となる。情報源の最小エントロピー k は、定数 $\alpha \in (0, 1)$ に対して、 $k = \alpha n, n^\alpha$ などを考える。種の長さ $d = \log D$ は小さいことが望ましく、 $d = \text{poly}(\log n)$ 程度を考える。リスト復号可能符号と比べると、 $m = n^{\Omega(1)}$ など考える点が大きく異なる。符号の場合、アルファベットサイズ q が定数のとき、 $m = \log(D)$ である。そして、符号化レート $n/(D \log q)$ を定数にすることを目指すとして、 $m = O(\log n)$ を目

表 3 乱数抽出器との対応関係

リスト復号可能符号	乱数抽出器
メッセージ	情報源からの入力系列
符号語	出力系列
受信語	検査 T
リストに含まれるメッセージ	T をパスできない入力系列
リストサイズ	T をパスできない入力系列の数

指すことになる。

リスト復号可能符号と乱数抽出器の対応関係は、表 3 のようにまとめることができる。

7. 困難性増幅器

困難性増幅器とは、一部の入力に対して計算することが困難な関数を、多くの入力に対して計算することが困難な関数に変換することである。計算できない入力の割合を増やしているため、困難性増幅器と呼ばれる。困難性増幅器は、計算の複雑さ理論や暗号理論などでよく登場する。具体的にどのように実現するかというと、例えば、関数 $f: \{0, 1\}^n \rightarrow \{0, 1\}$ に対して、関数 $f': \{0, 1\}^{nk} \rightarrow \{0, 1\}$ を次のように定義すると困難性が増幅する。

$$f'(x_1, x_2, \dots, x_k) = f(x_1) \oplus f(x_2) \oplus \dots \oplus f(x_k).$$

ここで、 $x_i \in \{0, 1\}^n, i \in [k]$ である。関数 f' の出力は k 個の $f(x_i)$ の結果の排他的論理和であるため、直観的には各 $f(x_i)$ 全てについて正しく計算できなければ f' は正しく計算できない。したがって、 f' を計算することは f を計算することよりも困難だと考えられる。

形式的に議論するために、まず、関数の困難性とは何かを考えよう。関数とは、入力が与えられ、それを基にしてある結果を出力するものである。その関数と全く同等の振舞いをするのであれば、その関数を計算できたと考えられる。例えば、マークシート式の試験問題では、問題が入力として与えられ、それを基にして回答を選択することになる。全問正解することができれば、試験問題を全て解いた（計算した）と考えられる。ある程度難しい試験問題では、問題の全てを正解することができず、例えば 6 割だけ正解できたということになる。この考えを基にして、関数の困難性を定義できそうであるが、このままではうまくいかない。なぜならば、試験問題の困難性は、それを回答する人に依存するからである。同じ試験問題を与えても、ある人は満点かもしれないし、別のある人は 0 点かもしれない。これでは試験問題の困難性を議論できない。そこで、問題の回答をする人を階層的にクラス分けすることを考える。例えば、中学生以下のクラス、高校生以下のクラス、大学生以下のクラスといった形に階層を分ける。そして、中学生以下では正答率が 4 割以下の試験問題である、といった形にすれば関数の困難性が定義できそうである。

実際は、実行時間で階層分けしたアルゴリズム、サイズで階層分けした回路などを用いて関数の困難性を定義する。本稿で議論する範囲内において、どちらの定義を用いても大きな違い

はないため、ここでは実行時間で階層分けしたアルゴリズムを用いて関数の困難性を定義する。

[定義 10] 関数 $f: [n] \rightarrow [q]$ が (t, δ) 困難であるとは、実行時間 t の任意のアルゴリズム A に対して、

$$\Pr[A(U_{[n]}) \neq f(U_{[n]})] > \delta$$

を満たすときである。

つまり、実行時間 t のアルゴリズムはいずれも、 δ より大きな割合の入力を正しく答えることができないことを表している。

困難性増幅とは、パラメータ δ を小さい値から大きい値に変換することである。そして、なるべく小さい値から始めて、なるべく大きい値に変換できることが望ましい。理想的なのは、 $\delta = 0$ から始めて、 $\delta = 1 - 1/q - \epsilon$ にすることである。 $\delta = 0$ は、最悪ケースの困難性に対応しており、先ほどの例を用いれば、解けない問題が 1 個しかない場合に対応している。 $\delta = 1 - 1/q - \epsilon$ は達成可能な困難性の最大値である。なぜならば、 q 通りある答えから常に同じ答えをする人を見ると、その中には、全体の $1/q$ 以上の割合を正解し、間違いが $1 - 1/q$ 以下の割合の人が存在する。つまり、簡単な答え方しかしない人で、パラメータ $\delta = 1 - 1/q$ を達成する人がいるからである。

困難性増幅は、一般的に、ただで手に入れることはできず、別の部分で代償を払うことになる。先ほどの例を用いると、大学生以下のクラスで正答率が 9 割以下の試験問題に対して困難性増幅を用いると、高校生以下のクラスで正答率が 1 割以下の試験問題へと変換される。つまり、対象となる階層クラスが下がってしまうのである。また、多くの場合、変換して得られた試験問題は、回答しなければならぬ問題の数が多くなっている。

次に、困難性増幅を証明する方法について説明する。ここでは特に、ブラックボックス的に困難性増幅する方法について説明する。関数を変換するためのアルゴリズム Amp と帰着のためのアルゴリズム Red が登場する。アルゴリズム Amp は、どのような (t, δ) 困難な f が与えられたとしても、それを (t', δ') 困難な f' に変換するアルゴリズムである。実際は、関数 f にオラクルアクセスするので、変換した関数 f' は Amp^f と表すことができる。オラクルアクセスとは、この場合、アルゴリズム Amp が実行中に、オラクル関数 f に入力 x を問い合わせることができ、問い合わせると 1 ステップで $f(x)$ を得ることができるというものである。 f にオラクルアクセスしていることを明示して、 Amp^f と書く。 f の計算時間を Amp^f の実行時間に含めなくてよいという点に注意しよう。次に、困難性が増幅したことを証明するために帰着アルゴリズム Red が登場する。アルゴリズム Red は、 f' の入力の $1 - \delta'$ 以上の割合を正しく計算できるアルゴリズム r が存在したとき、その r にオラクルアクセスすることで、 f の入力の $1 - \delta$ 以上の割合を正しく計算することができる。このような Amp と Red が存在すれば、 (t, δ) 困難な関数を (t', δ') 困難な関数へブラックボックス的に困難性が増幅できたことになる。まず、関数 f' への変換は、 Amp^f で実現できるため、 f の具体的な計算方法を知らずとも、ブラックボックス的に f' を構成できる。また、困難性が増幅したことは、 f' が (t', δ') 困難でないと仮定して f が (t, δ) 困難でないことを示せばよい。関数 f' が (t', δ') 困難でないときに

図 2 困難性増幅器

存在する計算アルゴリズムを r とする．このとき，Red の性質から， r にオラクルアクセスする Red^r は f の困難性を破ることができる．

以上を踏まえて困難性増幅器を定義する．ここでは，最悪ケースからの困難性増幅を考える．

[定義 11] 関数 $f: [n] \rightarrow \{0, 1\}$ をオラクルとして持つアルゴリズム $\text{Amp}^f: [D] \rightarrow [q]$ を考える． Amp^f が（最悪ケースから平均ケースへの） (t, k, ϵ) ブラックボックス困難性増幅器であるとは，実行時間 t の帰着アルゴリズム Red が存在し，

$$\Pr[r(U_{[D]}) = \text{Amp}^f(U_{[D]})] > 1/q + \epsilon$$

を満たす任意の $r: [D] \rightarrow [q]$ に対し，アドバイス $z \in [K]$, $K = 2^k$ が存在し，

$$\forall i \in [n], \quad \text{Red}^r(i, z) = f(i)$$

を満たすときである．

図 2 において，アルゴリズム Amp と Red の入出力およびオラクルアクセスの関係を示している．

帰着アルゴリズムは，関数 f を計算するためにアドバイスを受け取っている．アドバイス z は， r に依存して存在してよいが，全ての入力 $i \in [n]$ に対しては同じ z でなければならない．アドバイスを受け取ることができるようなアルゴリズムを考えているため，通常よりも少し弱い意味でのアルゴリズムの存在性を議論しているといえる．もちろん，アドバイスなしで計算できることが望ましいが，困難性増幅器が存在するにはアドバイスを入力として受け取る必要があることが知られている．

ここで定義した困難性増幅器は，最悪ケースから平均ケースへの困難性増幅が可能であることが，上で行った議論から分かる．

[命題 12] Amp^f が (t_r, t_r, ϵ) ブラックボックス困難性増幅器であり， f が $(t, 0)$ 困難であるとき， Amp^f は $(t/O(t_r), 1 - 1/q + \epsilon)$ 困難である．

困難性が増幅した関数 Amp^f が対象とするアルゴリズムの実行時間は，帰着アルゴリズム Red の実行時間である t_r の割合だけ小さくなっていることに注意しよう．

それでは，困難性増幅器を統一理論の枠組みで説明しよう．関数 $\Gamma: [N] \times [D] \rightarrow [D] \times [q]$ を次のように定義する．

$$\Gamma(f, y) = (y, \text{Amp}^f(y)).$$

ここで， $N = 2^n$ である．関数 $f: [n] \rightarrow \{0, 1\}$ が $f \in N$ と表現されていることに注意しよう．これは， f が真理値表で表

現されていると考えればよい．

このとき，以下が成り立つ．

[命題 13] Amp^f が (∞, k, ϵ) ブラックボックス困難性増幅器であるための必要十分条件は，

$$\forall r \in [q]^D, \quad |\text{LIST}_\Gamma(T_r, 1/q + \epsilon)| \leq K \quad (4)$$

である．ここで， $T_r = \{(y, r_y) : y \in [D]\}$, $K = 2^k$ である．

この命題で示されている条件は，リスト復号可能符号と全く同じである点に注意しよう．つまり，ブラックボックス困難性増幅器は，リスト復号可能符号と本質的に同じである．

また，困難性増幅器の帰着アルゴリズムの実行時間が無限大になっているが，これは，実行時間を考慮しない場合の困難性増幅器が特徴付けられているとことを意味している．通常の困難性増幅器は，帰着アルゴリズムの実行時間を考慮する必要があるが，それは，ここで示された条件 (4) よりも強い条件を満たしているのである．リスト復号可能符号の復号アルゴリズムの実行時間を考えていないことに対応しているともいえる．

ここで， $r \in [q]^D$ を関数 $r: [D] \rightarrow [q]$ とみなす．定義より，

$$\begin{aligned} \text{LIST}_\Gamma(T_r, 1/q + \epsilon) &= \{x \in [N] : \Pr[\Gamma(x, U_{[D]}) \in T_r] > 1/q + \epsilon\} \\ &= \{f \in [N] : \Pr_{y \leftarrow \frac{R}{[D]}}[(y, \text{Amp}^f(y)) \in T_r] > 1/q + \epsilon\} \\ &= \{f \in [N] : \Pr_{y \leftarrow \frac{R}{[D]}}[\text{Amp}^f(y) = r(y)] > 1/q + \epsilon\}. \end{aligned}$$

つまり，条件 (4) は，どのような関数 r に対しても，関数 Amp^f の $1/q + \epsilon$ 割合以上を正しく計算できている，そのような f の数は K 以下であるというものである．

まず，条件 (4) を満たすときに， Amp^f が (∞, k, ϵ) 困難性増幅器であることを説明する．上記の議論より，関数 r が関数 Amp^f の $1/q + \epsilon$ 割合以上を正しく計算できているとき，これを満たす関数 f の数は K 以下である．これらの関数を $f_{r,1}, f_{r,2}, \dots, f_{r,K}$ とする．このとき，帰着アルゴリズム Red を， $\text{Red}^r(i, z) = f_{r,z}(i)$ と定義する．すると，アドバイス $z \in [K]$ があれば， $\text{Red}^r(\cdot, z) = f(\cdot)$ とすることで f を正しく計算することができる．これは Amp と Red によって，最悪ケースから平均ケースへ困難性増幅できることを意味する．

次に， Amp^f が (∞, k, ϵ) 困難性増幅器であるときに，条件 (4) を満たすことを示す．帰着アルゴリズム Red と任意の関数 $r: [D] \rightarrow [q]$ を考える．困難性増幅器の性質から， $\Pr_{y \leftarrow \frac{R}{[D]}}[\text{Amp}^f(y) = r(y)] > 1/q + \epsilon$ であれば，あるアドバイス $z \in K$ が存在して， $\text{Red}^r(\cdot, z)$ によって f を計算することができる．アドバイス z は K 通りしかないため， $\Pr_{y \leftarrow \frac{R}{[D]}}[\text{Amp}^f(y) = r(y)] > 1/q + \epsilon$ を満たす f は K 以下である．

困難性増幅器の条件は，リスト復号可能符号の条件と同じであるため，両者は本質的に等価である．しかし，両者は興味のあるパラメータに違いがある．まず，パラメータ ϵ に関して，符号の場合，定数であれば十分なことが多い．一方，困難性増幅器の場合，無視できるほど小さい値として $\epsilon = 1/n^{\Omega(n)}$ が必要となることがある．ほかに，符号のリストサイズ K は，復号アル

表 4 困難性増幅器との対応関係

リスト復号可能符号	困難性増幅器
メッセージ	増幅したい関数 f
符号語	増幅後の関数 Amp^f
受信語	Amp^f を近似する関数 r
リストに含まれるメッセージ	Red^r で計算できる f
リストサイズ	アドバイスサイズ

ゴリズムの実行時間程度で抑える必要があるが、困難性増幅器の場合、アドバイスの長さ $k = \log K$ が、帰着アルゴリズムの実行時間程度であればよい。

リスト復号可能符号と困難性増幅器の概念を対応付けると、表 4 のようになる。

8. 擬似乱数生成器

擬似乱数生成器とは、ランダムに見える系列を出力する関数のことである。ランダムに見える系列というのは直観的には理解しやすいが、形式的にはどのように定義するのだろうか。そもそも、完全にランダムな系列とは何なのだろうか。様々な定式化が存在するが、ここでは、計算の複雑さ理論や暗号理論でよく使われる定式化を考える。

まず、完全にランダムな系列とは、アルファベット $\{0, 1\}$ 上の系列を考える場合、各ビットが独立に、一様ランダムにサンプルされた系列を指す。つまり、各ビットごとにコイン投げを行って得られた系列のことである。チューリング機械などの計算モデルにおいて確率的な振舞いを許すモデルでは、このようなコイン投げ装置を仮定することが多い。また、実際に使用するランダム系列の長さがあらかじめ決まっている場合、例えば n ビットまでしか使わない場合、 $\{0, 1\}^n$ 上の一様分布からサンプルされた系列だと考えることもできる。今回の議論では、実行時間に制限のあるアルゴリズムを想定するため、使うことができるランダム系列の長さは制限されており、一様分布からサンプルされたものが完全にランダムな系列であると考えてよい。

それでは、ランダムに見える系列について考える。まず、ある系列がランダムに見えるかどうかは、前節の関数の困難性と同様、それを見る人に依存すると考えられる。つまり、階層化されたクラスごとに、ランダムに見えるという概念を定義する。そして、ある系列が「ランダムに見える」とは、その系列を見ているのかそれとも完全にランダムな系列を見ているのか、どちらであるか区別できない状況であると考えられる。

以上を踏まえ、形式的な定義を与える。確率変数 X, Y が (t, ϵ) 識別不能であるとは、実行時間 t の任意のアルゴリズム A に対して、

$$|\Pr[A(X) = 1] - \Pr[A(Y) = 1]| \leq \epsilon$$

を満たすときである。この定式化の意味を考えるならば、実行時間 t のアルゴリズムでは、入力が X であった場合と Y であった場合では、確率として ϵ 以下の差しか出ないことを表している。例えば A は入力が X であると判断した場合は 1 を出力し、 Y と判断した場合はそれ以外の値を出力するアルゴリズム

だと考える。そのような動作をする実行時間 t の任意のアルゴリズムに対して、入力が X だと判断する確率は、実際の入力が X の場合と Y の場合とでは ϵ 以下の差しかないという意味である。擬似乱数生成器とは、一様分布と識別不能な分布を生成する関数のことである。

[定義 14] 関数 $G : [D] \rightarrow [M]$ が (t, ϵ) 擬似乱数生成器であるとは、 $G(U_{[D]})$ と $U_{[M]}$ が (t, ϵ) 識別不能のときである。擬似乱数生成器は、入力として一様ランダムな系列を受け取っている点に注意しよう。つまり、擬似ランダム系列を生成するために、一様ランダムな系列を使うことができるのである。ただし、受け取る一様ランダムな系列は、出力する系列よりも短くないと意味のある議論ができないため、入力長 $d = \log D$ と出力長 $m = \log M$ に対して、 $d < m$ である必要がある。

擬似乱数生成器によって生成される擬似ランダムな分布は、一様分布の代わりとして使うことができる。確率 ϵ の差を除けば、一様分布であるとだまされてしまうからである。そのため、計算の複雑さ理論では、確率的アルゴリズムが必要とするコイン投げの回数を減らすため（場合によっては完全になくすため）に使われたりする。暗号理論では、Shannon によって不可能性が示された、メッセージよりも短い鍵を使った秘密鍵暗号を、計算量の制限された敵に対して実現するために使うことができる。

暗号理論で用いられる擬似乱数生成器と、そうでないものには違いがある。それは、暗号理論の場合、だます側とだまされる側が同じ計算の階層クラスに入っているという点である。つまり、擬似乱数生成器は多項式時間で計算可能であり、かつ、出力分布は多項式時間では一様分布と見分けがつかないことが必要である。一方で、計算の複雑さ理論では、だますためには、だまされる側よりも多くの計算コストをかけてもよい場合がある。この違いは、もちろん目的の違いに由来するが、非暗号理論の擬似乱数生成器は、暗号理論にとって必要不可欠な仮定「一方向性関数の存在」なしで議論することが可能である。

そして、この章で特徴付けを行うのは、非暗号理論的な擬似乱数生成器を、最悪ケースの計算困難関数から構成する場合である。困難性増幅器と同様に、ブラックボックス的に構成することを考える。

[定義 15] 関数 $f : [n] \rightarrow \{0, 1\}$ をオラクルとして持つアルゴリズム $G^f : [D] \rightarrow [M]$ を考える。 G が (t, k, ϵ) ブラックボックス擬似乱数生成器構成であるとは、実行時間 t の帰着アルゴリズム Red が存在し、

$$\Pr[T(G^f(U_{[D]})) = 1] - \Pr[T(U_{[M]}) = 1] > \epsilon$$

を満たす任意の $T : [M] \rightarrow \{0, 1\}$ に対し、アドバイス $z \in [K]$ が存在し、

$$\forall i \in [n], \quad \text{Red}^T(i, z) = f(i)$$

を満たすときである。ここで、 $K = 2^k$ である。

困難性増幅器の定義と比較すると、帰着アルゴリズムを使って f が計算できるためのアルゴリズム T （困難性増幅器の場合は r ）の条件が、困難性を破ることであるか、擬似ランダム性を破ることであるかが違うだけである。

表 5 擬似乱数生成器構成との対応関係

リスト復号可能符号	擬似乱数生成器構成
メッセージ	最悪ケース困難関数 f
符号語	擬似乱数生成器 G^f
受信語	検査 T
リストに含まれるメッセージ	Red^T で計算できる f
リストサイズ	アドバイスサイズ

上記で定義した擬似乱数生成器構成を用いると、最悪ケースで計算困難な関数から擬似乱数生成器を構成することができる．
 [命題 16] G が (t_r, t_r, ϵ) ブラックボックス擬似乱数生成器構成であり、 f が $(t, 0)$ 困難であるとき、 G^f は $(t/O(t_r), \epsilon)$ 困難である．

統一理論の枠組みで特徴付けるために、関数 $\Gamma: [N] \times [D] \rightarrow [M]$ を以下のように定義する．

$$\Gamma(f, y) = G^f(y).$$

困難性増幅器の場合と同様、関数 f は $[N]$ の要素として表現されている点に注意しよう．

[命題 17] G が (∞, k, ϵ) ブラックボックス擬似乱数生成器構成であるための必要十分条件は、

$$\forall T \subseteq [M], \quad |\text{LIST}_\Gamma(T, |T|/M + \epsilon)| \leq K$$

を満たすときである．ただし、 $K = 2^k$ である．

ここで示された条件は、乱数抽出器と全く同じである点に注意しよう．つまり、擬似乱数生成器のブラックボックス構成は、乱数抽出器と本質的に同じである．また、帰着アルゴリズムの実行時間が無限大になっているのは、実行時間を考慮しない場合が特徴付けられていることを意味する．

パラメータに関して、擬似乱数生成器構成が乱数抽出器と異なる点は、出力長 m である．擬似乱数生成器構成の場合、 m は、実行時間 t やアドバイスの長さ k の多項式で関連付けられていけばよいが、乱数抽出器の場合、なるべく多くの乱数を抽出したいため、 m を k に近づけることに興味がある．

リスト復号可能符号の概念と、擬似乱数生成器構成を対応づけると、表 5 のようになる．

9. Guruswami, Umans, Vadhan の研究成果

本章では、統一理論の枠組みを利用して得られた成果として、Guruswami, Umans, Vadhan⁽¹²⁾の研究成果を紹介する．彼らは、リスト復号可能符号として提案された Parvaresh-Vardy 符号⁽¹⁷⁾を基にすると、拡大係数が次数に任意に近いエクスパンドグラフが構成できることを示した．そして、そのエクスパンドグラフを利用することで、ほぼ最適な乱数抽出器を構成できることを示した．乱数抽出器の既存の構成法が非常に複雑であったのに対し、代数的性質を利用することで、簡素な構成と性能分析が可能になっている．

4. において、エクスパンドグラフとリスト復号可能符号を比較しているが、一般的には、リスト復号可能符号の構成はそのままエクスパンドグラフにはならない．エクスパンドグラフである

ための条件 (1) における集合 T のとり方が異なるためである．リスト復号可能符号では、ある一定サイズの集合 $T_r = \{(y, r_y)\}$ に対して条件を満たせばよかったが、エクスパンドグラフでは、ある一定サイズ以下の任意の集合 T に対して条件を満たす必要がある．誤り訂正符号の言葉で言うと、ある長さ以下の任意の受信語に対してリストサイズが抑えられることを保証する必要がある．通常、受信語の長さは符号長に等しいため、受信語が短い場合にリストサイズが抑えられるかどうかについては、更なる分析が必要である．Guruswami らは、Parvaresh-Vardy 符号という具体的な構成法が、リスト復号可能符号だけでなく、エクスパンドグラフの性質も満たしていることを示したのである．ここでは、Parvaresh-Vardy 符号の構成、及びそれがエクスパンドグラフであることについて説明を行う．

誤り訂正符号のメッセージを $n-1$ 次多項式 $f \in \mathbb{F}_q[Y]$ で表現するとき、リードソロモン符号では、 f を $\mathbb{F}_q$ の全ての点で評価した系列が符号語である．Parvaresh-Vardy 符号では、メッセージ f に対し、 f を基に構成される $n-1$ 次多項式 $f_0, f_1, \dots, f_{m-1}$ を $\mathbb{F}_q$ の全ての点で評価し、 m 個の評価値をまとめて $\mathbb{F}_q^m$ の要素だとみなし、符号語としている．符号長はリードソロモン符号と同じく q であるが、符号語のアルファベットが $\mathbb{F}_q^m$ であるため、シンボル当りの情報量が多く、リスト復号性能が優れている．統一理論の枠組みを用いると、Parvaresh-Vardy 符号は次の関数 $\Gamma: \mathbb{F}_q^n \times \mathbb{F}_q \rightarrow \mathbb{F}_q \times \mathbb{F}_q^m$ で定義される．

$$\Gamma(f, y) = [y, f_0(y), f_1(y), \dots, f_{m-1}(y)]. \quad (5)$$

ただし、 $f_0 = f$ 、 $i = 1, \dots, m-1$ に対して $f_i(Y) = f(Y)^{h^i} \bmod E(Y)$ である．ここで、 h はある整数、 $E(Y)$ は $\mathbb{F}_q$ 上の n 次既約多項式である．この関数 Γ によって定義されるグラフは、エクスパンドグラフである．

[定理 18] 条件 (5) で定義される 2 部グラフは、 $(h^m, q - (n-1)(h-1)m)$ エクスパンドである．

Parvaresh-Vardy 符号のパラメータ n, m, h, q をうまく設定することで、拡大係数が次数に任意に近いエクスパンドグラフを与えることができる．

[定理 19] 任意の定数 $\alpha > 0$ 、任意の $N \in \mathbb{N}, K \leq N, \epsilon > 0$ に対して、明示的構成法を持つ $(K, (1-\epsilon)D)$ エクスパンドグラフが存在する．ただし、左頂点数は N 、左頂点次数は $D = O((\log N)(\log K)/\epsilon)^{1+1/\alpha}$ 、右頂点数は $M \leq D^2 \cdot K^{1+\alpha}$ である．更に、 D や M は 2 のべき乗である．

以降では、定理 18 の証明を解説する． $K \leq h^m$ を任意に固定し、 $A = q - (n-1)(h-1)m$ とする．命題 5 より、サイズ AK 未満の全ての $T \subseteq \mathbb{F}_q^{m+1}$ に対して、 $|\text{LIST}_\Gamma(T, 1)| < K$ を示せば十分である．

(1) T 中の全ての点において評価値が零である低次数多項式 $Q(Y, Y_1, \dots, Y_m)$ が存在することを示す． Q において非零の係数を持つ項は、 $Y^i M_j(Y_1, \dots, Y_m), 0 \leq i \leq A-1, 0 \leq j \leq K-1 \leq h^m$ の形の項だけであるとする．ここで、 $M_j(Y_1, \dots, Y_m) = Y_1^{j_1} Y_2^{j_2} \dots Y_m^{j_m}$ であり、 $j_1, \dots, j_m$ は j の h 進展開である (つまり、 $j = j_1 + j_2 h + \dots + j_m h^{m-1}$)．したがって、項の数

は合計で AK 個である．一方で，各 $z \in T$ に対して， $Q(z) = 0$ という条件は， Q の AK 個の係数に対して線形制約を課していることに等しい． Q は AK 個の項を持っており，それに対して $|T| \leq AK - 1$ 個の制約条件を課しているため，非零多項式 Q が存在する．更に， Q は条件を満たす多項式の中で， Y についての次数が最小のものだと仮定する．このとき， $Q(Y, Y_1, \dots, Y_m) = \sum_{j=0}^{K-1} p_j(Y) \cdot M_j(Y_1, \dots, Y_m)$ と表現した場合，多項式 $p_0, \dots, p_{K-1}$ のうち，少なくとも一つは $E(Y)$ で割り切れないことが分かる．(もし割り切れた場合， $Q/E(Y)$ という多項式は，条件を満たし，次数がより低くなる．)

(2) 任意の $f \in \text{LIST}_\Gamma(T, 1)$ に対して，

$$R_f(Y) = Q(Y, f_0(Y), f_1(Y), \dots, f_{m-1}(Y)) = 0$$

である．なぜならば， $R_f(Y)$ の次数は， $A-1+(n-1)(h-1)m = q-1$ 以下である．一方で， $f \in \text{LIST}_\Gamma(T, 1)$ であるため，任意の $y \in \mathbb{F}_q$ について， $R_f(y) = 0$ である．したがって， $R_f(Y)$ は，零点の数より次数が小さいため，恒等的に零である．

(3) 多項式 $f(Y)$ を拡大体 $\mathbb{G} = \mathbb{F}_q[Y]/E(Y)$ の要素だと考えると，全ての $f(Y) \in \text{LIST}_\Gamma(T, 1)$ は以下に定義される多項式 $Q^*(Z) \in \mathbb{G}[Z]$ の根であることが分かる．

$$Q^*(Z) = Q(Y, Z, Z^h, Z^{h^2}, \dots, Z^{h^{m-1}}) \bmod E(Y).$$

なぜならば， $f(Y) \in \text{LIST}_\Gamma(T, 1)$ に対して， $f_i(Y) \equiv f(Y)^{h^i} \pmod{E(Y)}$ であるので，

$$\begin{aligned} Q(Y, f(Y), f(Y)^h, \dots, f(Y)^{h^{m-1}}) \\ \equiv Q(Y, f_0(Y), \dots, f_{m-1}(Y)) \equiv 0 \pmod{E(Y)} \end{aligned}$$

が成り立つからである．また，このことから， $Q^*(Z)$ の根の数は $|\text{LIST}_\Gamma(T, 1)|$ 以上であることが分かる．次に， $Q^*(Z)$ が零多項式でないことを示す．定義より，

$$\begin{aligned} Q^*(Z) &= Q(Y, Z, Z^h, Z^{h^2}, \dots, Z^{h^{m-1}}) \bmod E(Y) \\ &= \sum_{j=0}^{K-1} (p_j(Y) \bmod E(Y)) \cdot M_j(Z, Z^h, \dots, Z^{h^{m-1}}) \\ &= \sum_{j=0}^{K-1} (p_j(Y) \bmod E(Y)) \cdot Z^j. \end{aligned}$$

仮定より， $p_j(Y)$ のうち少なくとも一つは $E(Y)$ で割り切れない．したがって $Q^*(Z)$ は非零多項式である． $Q^*(Z)$ の次数は $K-1$ 以下であるので， $|\text{LIST}_\Gamma(T, 1)| \leq K-1$.

10. おわりに

本稿では，Vadhan によって示された擬似ランダムオブジェクトの統一理論の枠組みを，リスト復号可能符号の視点を基に解説してきた．そして最後に，具体的構成法として，リードソ

ロモン符号を一般化した Parvaresh-Vardy 符号が，優れたエクスパンダグラフの構成になっていることを紹介した．

この統一理論の枠組みが，なぜ様々なオブジェクトを特徴付けできているか，その理由については説明できない．単なる偶然かもしれないし，本質的な何かがあるのかもしれない．ただ，具体的な例を通して，統一的な構造がどのように各オブジェクトに対して働いているかを見ることができた．研究をしていく中で，擬似ランダム構造を持つオブジェクトに遭遇したときは，この統一理論の枠組みに当てはめて考えてみるのもよいかもしれない．もし，この枠組みに当てはまれば，既存の様々な道具を使うことができる．完全に当てはまらない場合でも，本稿で解説したような形で，誤り訂正符号，グラフ，乱数抽出器など，様々な文脈のオブジェクトとして眺めることで，興味深い特徴付けができるかもしれない．

文 献

- (1) S. Vadhan, “The unified theory of pseudorandomness,” ACM SIGACT News, vol. 38, no. 3, pp. 39–54, 2007.
- (2) S. Vadhan, “The unified theory of pseudorandomness,” in Proc. ICM 2010.
- (3) M. Sudan, “List decoding: algorithms and applications,” ACM SIGACT News, vol. 31, no. 1, pp. 16–27, 2000.
- (4) V. Guruswami, “List decoding of binary codes – a brief survey of some recent results,” in Proc. IWCC 2009, LNCS, vol. 5557, pp. 97–106, 2009.
- (5) L. Trevisan, “Some applications of coding theory in computational complexity,” Quaderni di Matematica, vol. 13, pp. 347–424, 2004.
- (6) S. Hoory, N. Linial, and A. Wigderson, “Expander graphs and their applications,” Bull. Aust. Math. Soc., vol. 43, no. 4, pp. 439–561, 2006.
- (7) O. Goldreich, “A sample of samplers: A computational perspective on sampling,” available at <http://www.wisdom.weizmann.ac.il/~oded/sampler.html>, 2010. Preliminary version appeared as ECCC Technical Report, TR97-020, 1997.
- (8) N. Nisan and A. Ta-Shma, “Extracting randomness: A survey and new constructions,” J. Comput. Syst. Sci., vol. 58, no. 1, pp. 148–173, 1999.
- (9) R. Shaltiel, Recent developments in explicit constructions of extractors, in G. Păun, G. Rozenberg, and A. Salomaa ed., Current Trends in Theoretical Computer Science: The Challenge of the New Century, vol. 1, pp. 189–228, 2004.
- (10) V. Kabanets, “Derandomization: A brief overview,” Bull. Eur. Assoc. Theor. Comput. Sci., no. 76, pp. 88–103, 2002.
- (11) S. Vadhan, “Pseudorandomness,” Foundations and Trends in Theoretical Computer Science, 2011, to appear. A draft version is available at <http://people.seas.harvard.edu/~salil/pseudorandomness/>.
- (12) V. Guruswami, C. Umans, and S. Vadhan, “Unbalanced expanders and randomness extractors from Parvaresh-Vardy codes,” J. ACM, vol. 56, no. 4, 2009. Preliminary version appeared in Proc. CCC 2007.
- (13) P. Elias, “List decoding for noisy channels,” Technical Report 335, Research Laboratory of Electronics, MIT, 1957.
- (14) J.M. Wozencraft, “List decoding,” Quarterly Progress Report, Research Laboratory of Electronics, MIT, no. 48, pp. 90–95, 1958.
- (15) O. Goldreich and L. Levin, “A hard-core predicate for all one-way functions,” in Proc. STOC 1989, pp. 25–32,

1989.

- (16) M. Sudan, “Decoding of Reed-Solomon codes beyond the error-correction bound,” J. Complexity, vol. 13, no. 1, pp. 180–193, 1997
- (17) F. Parvaresh and A. Vardy, “Correcting errors beyond the Guruswami-Sudan radius in polynomial time,” in Proc. FOCS 2005, pp. 285–294, 2005.

(幹事団提案, 平成 23 年 4 月 26 日受付 6 月 9 日最終受付)

安永 憲司 (正員)

2003 阪大・基礎工・情報卒, 2005 同大学院情報科学研究科博士前期課程了, 2008 同大学院博士後期課程了, 同年関西学院大・博士研究員, 同年東工大大学院情報理工学研究科・数理・計算科学専攻・助教・博士(情報), 符号理論・暗号理論等の研究に従事.