コロモゴロフ複雑性と
アルゴリズム論的情報理論

コロモゴロフ複雑性と
アルゴリズム論的情報理論
Agenda

Part I
ランダムさの不思議

Part II
計算可能性理論とコロモゴロフの複雑性

Part III
アルゴリズム論的情報理論

Part I
ランダムさの不思議
Agenda

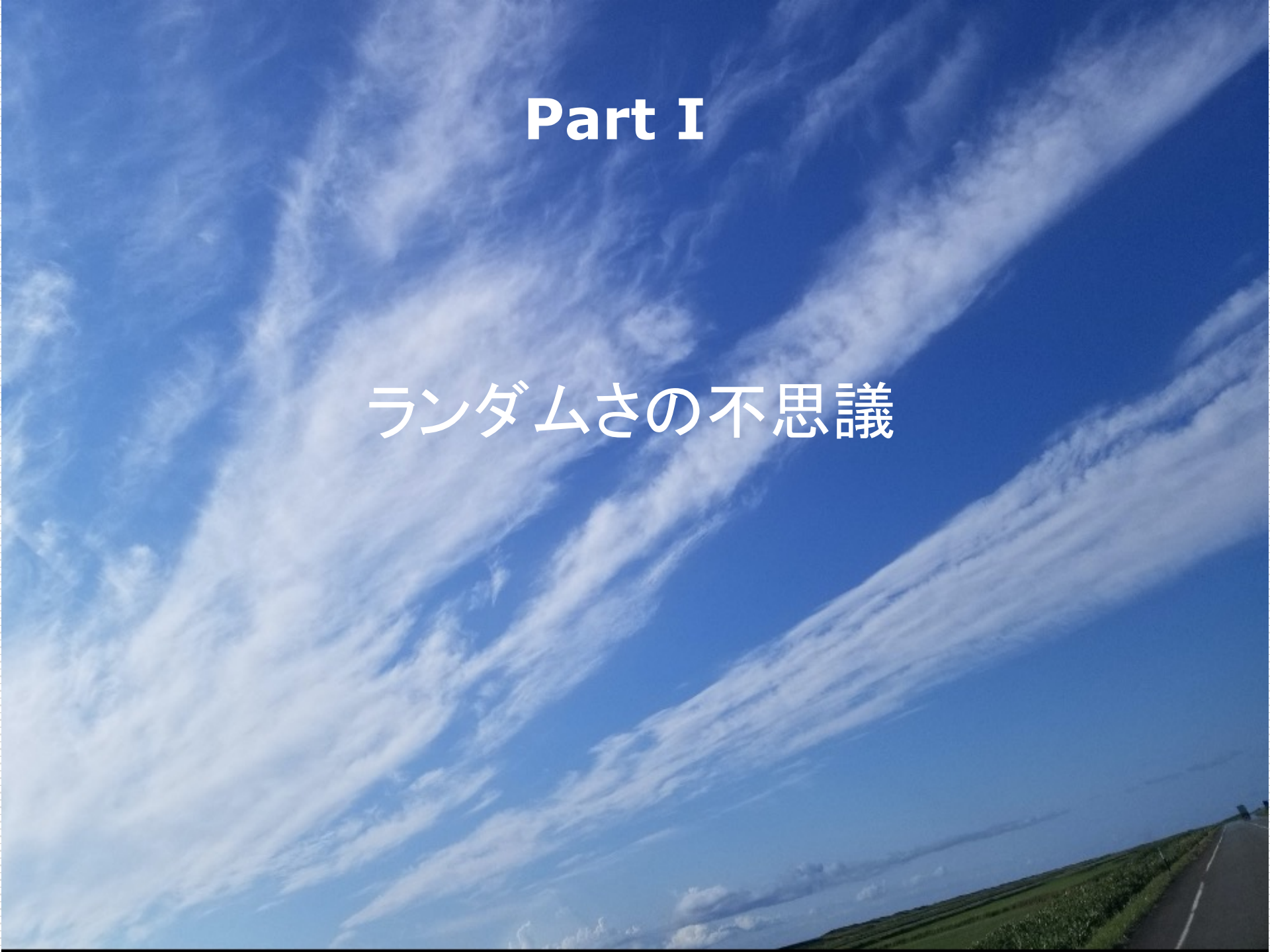
- ランダムさの不思議
- Randomness と Symmetry
- ランダムの力 ― 量子論のインパクト
- ランダムさの定義の難しさ
- ランダムさ への新しいアプローチ

Part II
計算可能性理論とコロモゴロフの複雑性
Agenda

- コロモゴロフの複雑性
- 全ての計算可能な関数はプログラムで表現できる
- コロモゴロフ複雑性の「不変性」
- コロモゴロフ複雑性の「計算不能性」
- チャイティンの不完全性定理

Part III
アルゴリズム論的情報理論
Agenda

- チャイティンのΩとバエズのエントロピー
- アルゴリズム論的情報量
- 二つのアルゴリズム論的情報量の関係
- アルゴリズム論的情報量に確率の概念を導入する
- 分配関数 Z – 統計力学の方法を振り返る」
- アルゴリズム論的情報理論への分配関数Zの応用
- Gibbs Ensemble と共役変数
- アルゴリズム論的熱力学


Part I
ランダムさの不思議

Part I
ランダムさの不思議
Agenda

- ランダムさの不思議
- Randomness と Symmetry
- ランダムの力 ― 量子論のインパクト
- ランダムさの定義の難しさ
- ランダムさ への新しいアプローチ

ランダムさの不思議

ランダムさの不思議

「ランダムさ」に対して、我々はある種の直観を持っているように見えます。ただ、そうした直観が、必ずしも正確なものでないことを、次のような例で話そうと思います。

- ランダムな0と1の並び
- ランダムな文字列

「ランダムさ」というのは、実は、定義するのが難しい、不思議な性質を持っています。

ランダムな 0と1の並び
コイントス

ランダムな 0と1の並びを作ろうと思ったら、コインを投げて、表が出たら0、裏が出たら1とすればいいですね。そうして作られた並びは、きっとランダムなものに見えるでしょう。

ところで、もしも、0が30個並んでいる並びを見せられたら、それはランダムなものには見えないでしょう。

なぜでしょう？

「コインを投げて、30回連続して表が出ることは、確率的にあり得ないから」

ランダムな 0と1の並び
コイントス

ただ、確率的には、コイントスで得られる30桁の0,1の 並びは、全て同じ確率 (1/2)^30 を持っています。

みなに「ランダム認定」されるだろう 0,1の30桁の並びと、0が30個並ぶ並びとは、同じ確率で出現します。

確率だけでは、ランダムなものとそうでないものを区別することはできません。

ランダムな文字列
猿のタイプライター

ランダムな文字列を得るために、猿にタイプライターを叩かせましょう。

ランダムな文字列
猿のタイプライター

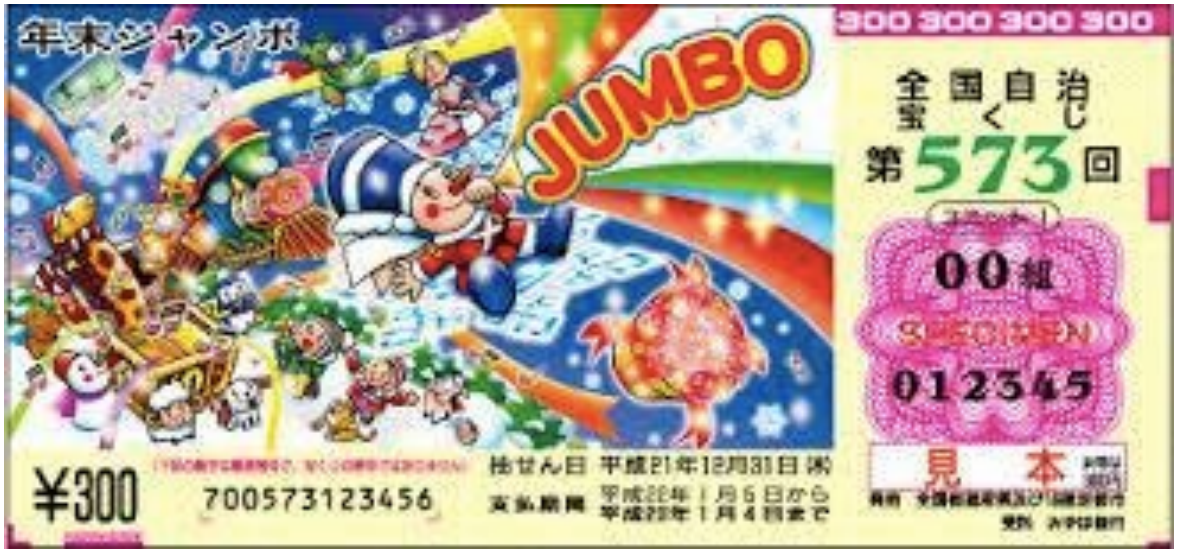
猿が打ち出す文字列は、ランダムと言えるでしょうか？

アルファベット（小文字）は、‘a’から‘z’まで26文字です。
もしも猿が26キーのタイプライターを6回適当に叩いたとすると、この時、猿がうちだすことが可能な文字列の総数は、26の6乗で、308,915,776になります。9桁の数字で3億ちょっとです。

この中には、”monkey”という文字列が含まれています。
“monkey”はランダムな文字列でしょうか？　英語の辞書で6文字の単語は、全てこの中に含まれています。それらは、ランダムな文字列とは言われないような気がします。

宝くじに当たる確率と比較する

宝くじの番号は、75組 159149番のように8桁なので、タイプライター猿が、"monkey"と打ち出すのは、サマージャンボで5億円当てるより、30倍ほど難しいことがわかります。
30回コインを投げて全部表が出る確率は、
2^30=(2^10)^3 ≈ (10^3)^3 = 10^9 ですので、宝くじに当たるより、10倍ほど難しいことになります。



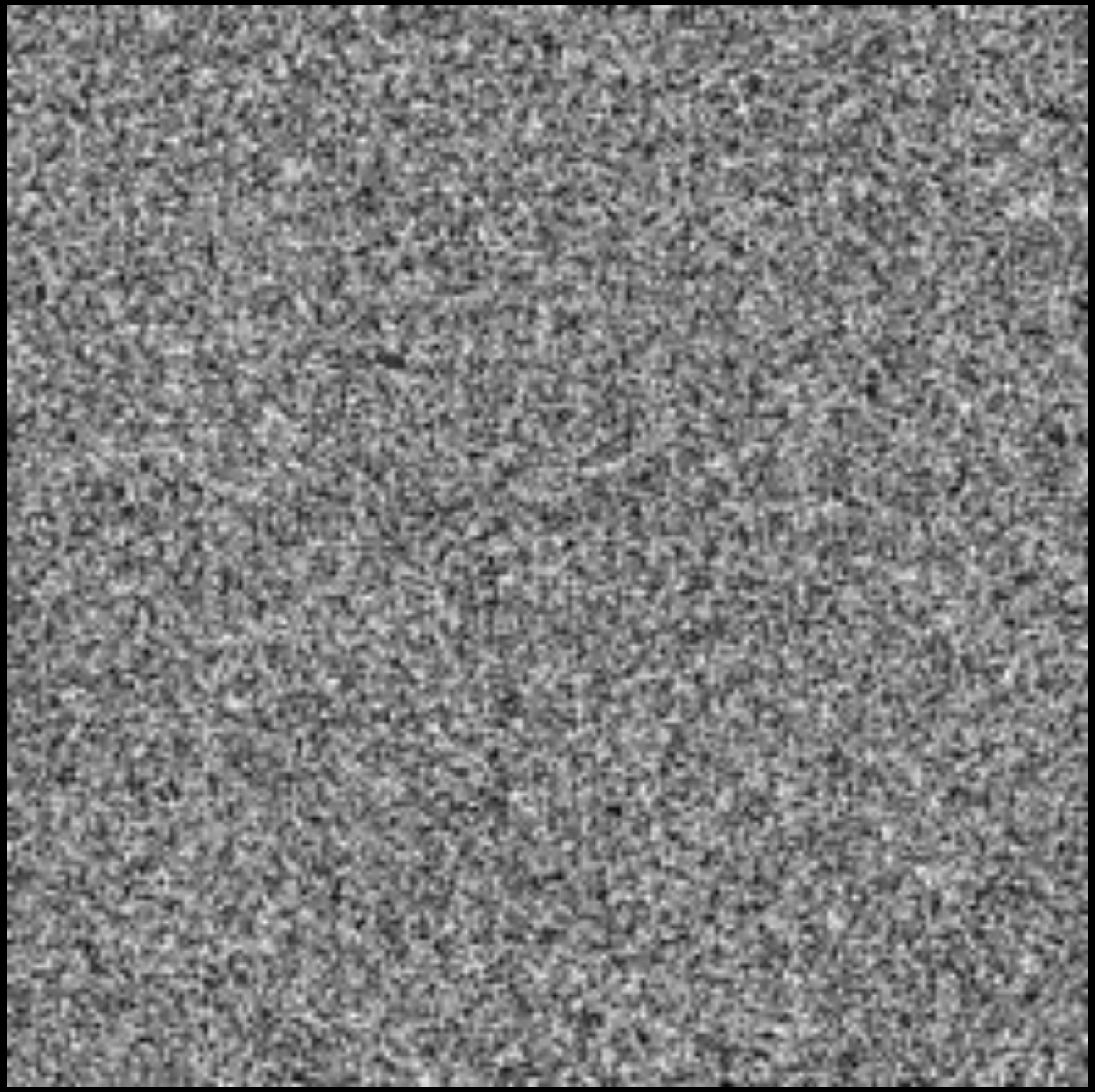
「ランダムさ」を定義することの難しさ

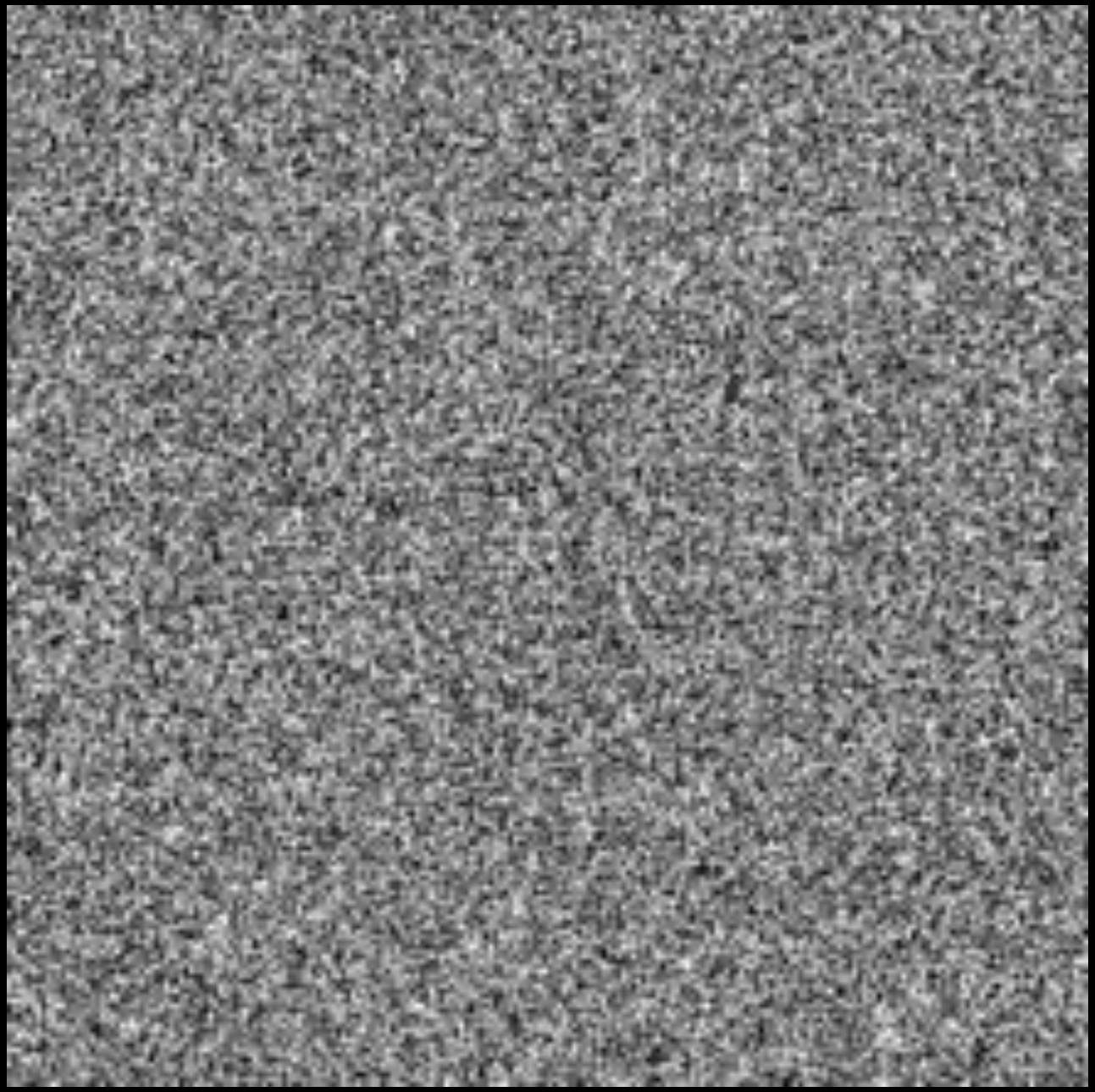
ランダムさを定義することの難しさに関して、フォン・ノイマンは、かつて、こう言っていました。

“There is no such thing as a random number—there are only methods to produce random numbers, and a strict arithmetical procedure is of course not such a method.”

ただ、コロモゴロフは、この問題に新しい光を当てることに成功します。それについては、第二部で扱います。

RandomnessとSymmetry

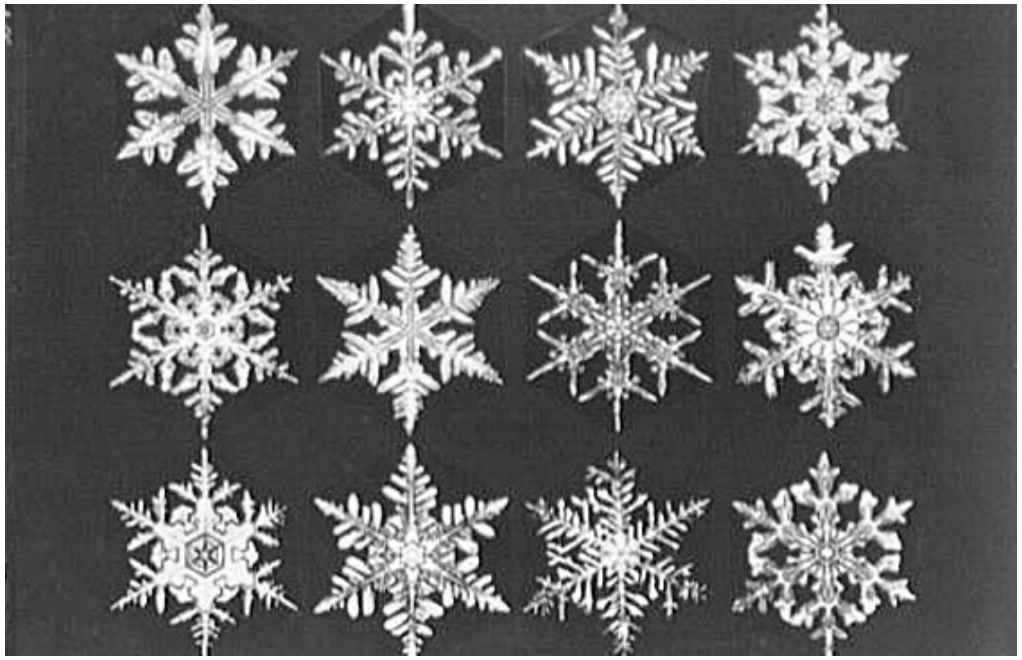




Crystal

Symmetry

Hope_Diamond



Twinned pyrite crystal group.

Biology


Spiral

Turing Pattern

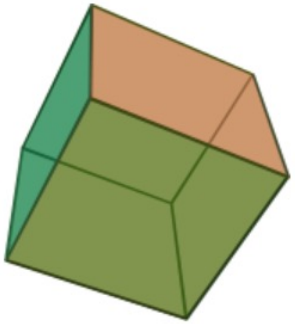
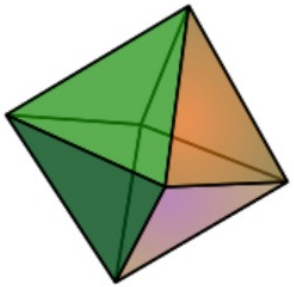
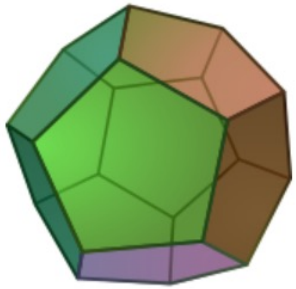
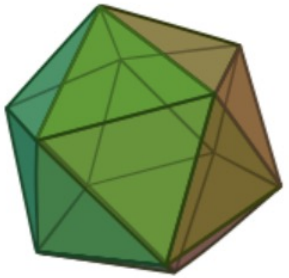
Structure


Ppantheon, Vatican Museums

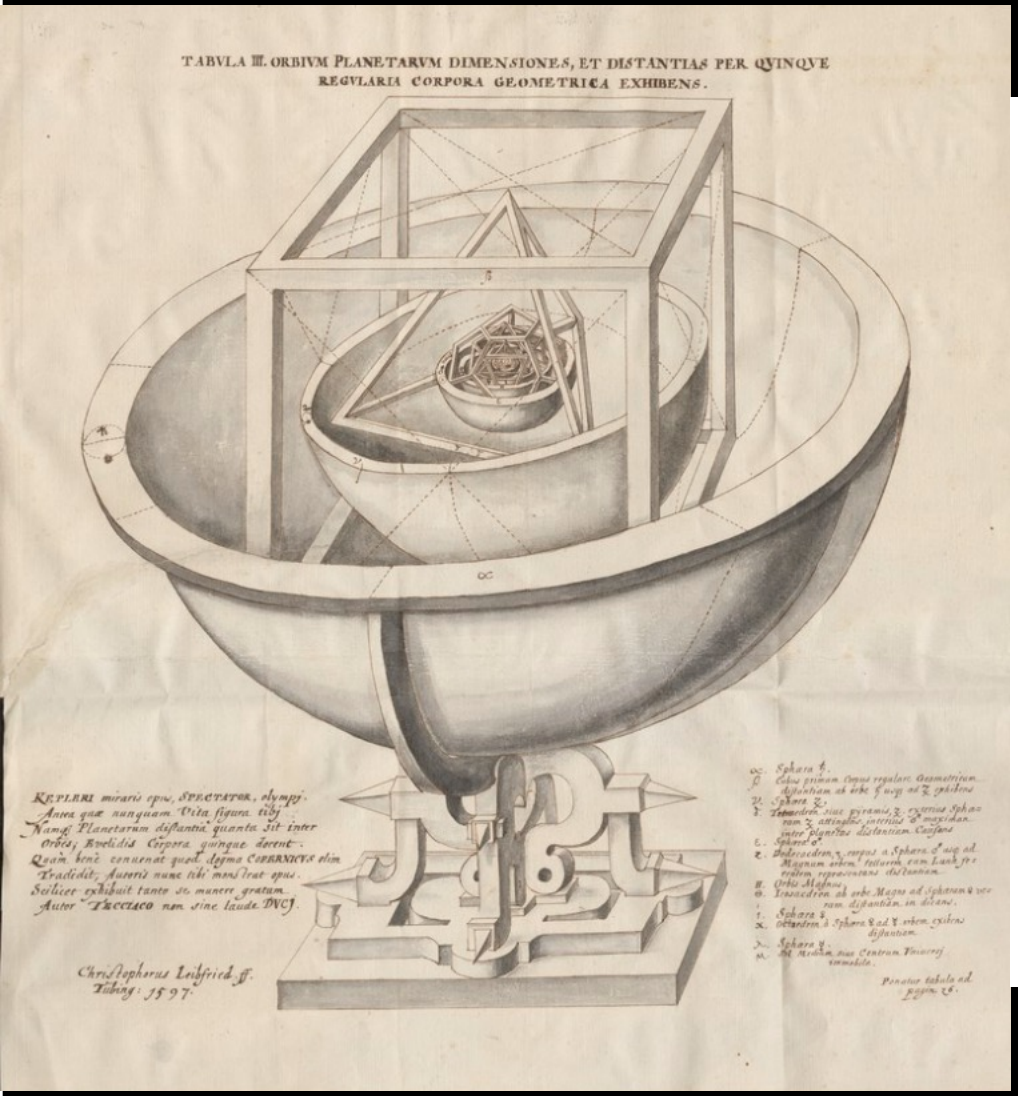

Piazza San Pietro

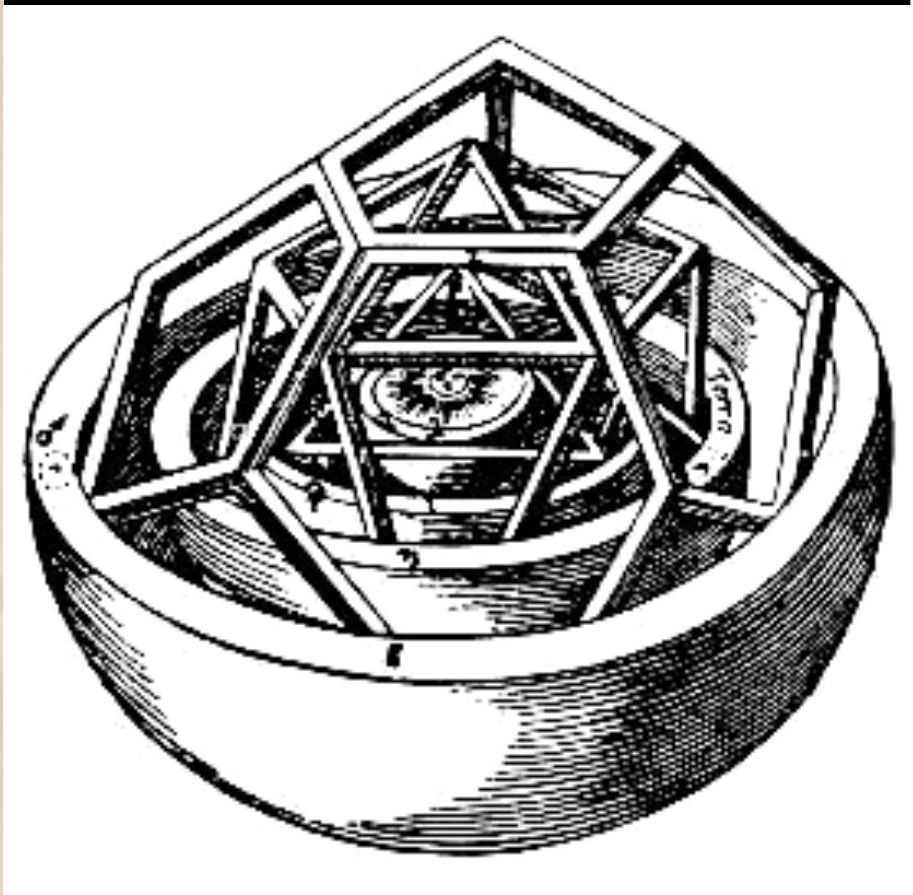
Wakkanai Dome

Astronomy

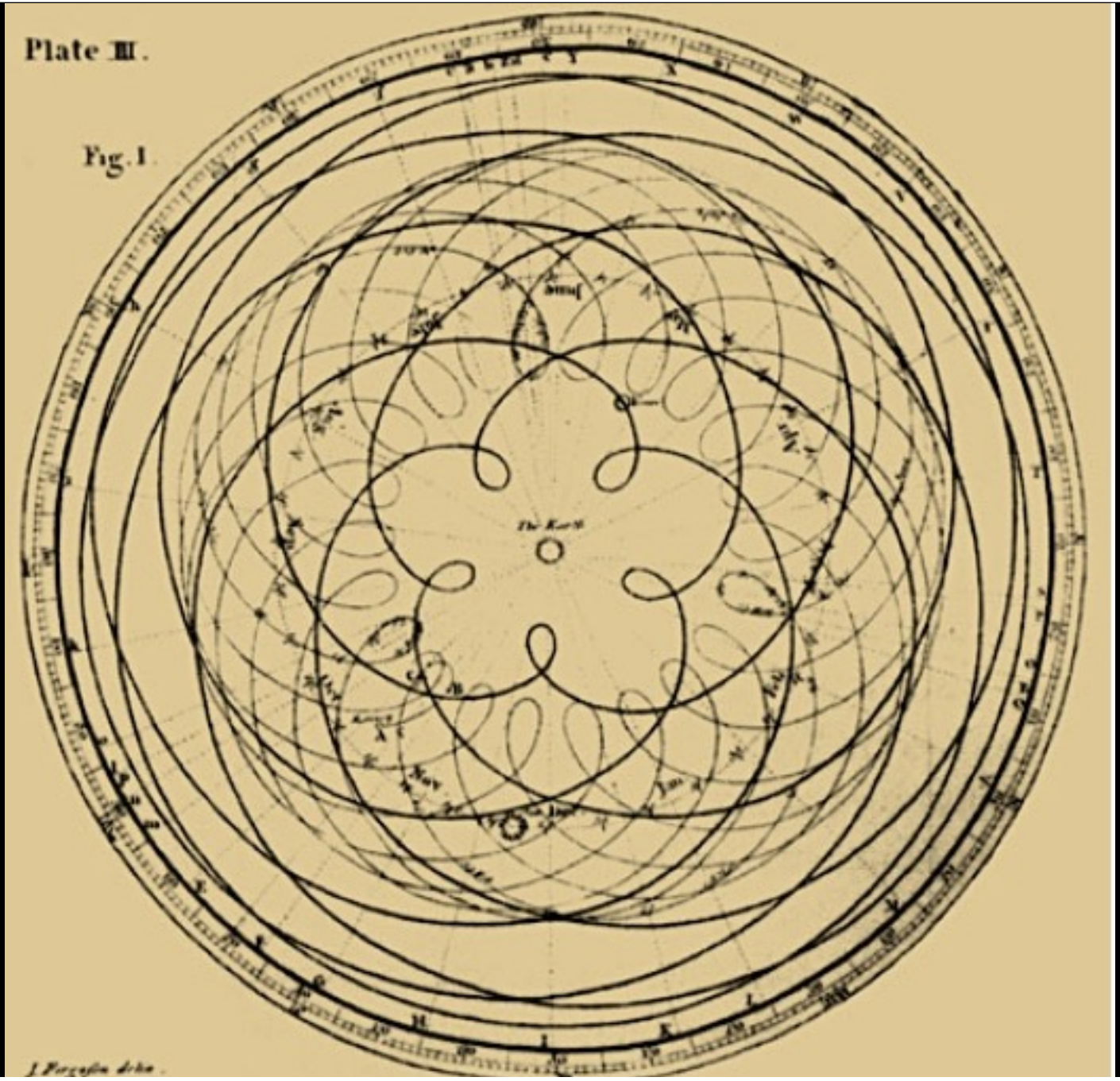
Tetrahedron	Cube	Octahedron	Dodecahedron	Icosahedron
Four faces	Six faces	Eight faces	Twelve faces	Twenty faces
				

Platonic solid





Kepler , Mysterium Cosmographicum



James Ferguson, *Astronomy Explained Upon Sir Isaac Newton's Principles*:


複雑なものはランダムか？

$\pi = 3.$
1415926535 8979323846 2643383279 5028841971 6939937510 5820974944 5923078164 0628620899
8628034825 3421170679 8214808651 3282306647 0938446095 5058223172 5359408128 4811174502
8410270193 8521105559 6446229489 5493038196 4428810975 6659334461 2847564823 3786783165
2712019091 4564856692 3460348610 4543266482 1339360726 0249141273 7245870066 0631558817
4881520920 9628292540 9171536436 7892590360 0113305305 4882046652 1384146951 9415116094
3305727036 5759591953 0921861173 8193261179 3105118548 0744623799 6274956735 1885752724
8912279381 8301194912 9833673362 4406566430 8602139494 6395224737 1907021798 6094370277
0539217176 2931767523 8467481846 7669405132 0005681271 4526356082 7785771342 7577896091
7363717872 1468440901 2249534301 4654958537 1050792279 6892589235 4201995611 2129021960
8640344181 5981362977 4771309960 5187072113 4999999837 2978049951 0597317328 1609631859
5024459455 3469083026 4252230825 3344685035 2619311881 7101000313 7838752886 5875332083
8142061717 7669147303 5982534904 2875546873 1159562863 8823537875 9375195778 1857780532
1712268066 1300192787 6611195909 2164201989 …

0：4999億9897万6328回
1：4999億9996万6055回
2：5000億0070万5108回
3：5000億0015万1332回
4：5000億0026万8680回
5：4999億9949万4448回
6：4999億9893万6471回
7：5000億0000万4756回
8：5000億0121万8003回
9：5000億0027万8819回

ランダムの力
-- 量子論のインパクト --

ランダムの力
-- 量子論のインパクト --

私たちのこころには、「秩序」「シンメトリー」を持つものを「美しい」と感ずる傾向があります。

こうした傾向は、芸術だけに見られるものでなく、物理学も、ある意味で「対称性」を追求します。

しかし、少なくとも、自然認識に関して言えば、20世紀の物理学は大きな転換に遭遇することになります。

それは、量子論が、自然の原理には、「ランダムさ」が深く組み込まれていることを発見したからです。

物理学者の人間の「認識」への関心

アインシュタインが「神はサイコロを振らない」といって、量子論を不完全な理論と見做したことはよく知られています。

ここでは、物理学者たちが、量子論の非決定論的性質、自然の「ランダムさ」を、どのように理解しようとしたかを見てみたいと思います。

興味深いのは、多くの物理学者が、人間の「認識」の能力との関係で、これらの問題を捉えようとしたことだと思います。

“Quantum mind”

https://en.wikipedia.org/wiki/Quantum_mind

Eugene Wigner

彼は、量子力学は、こころの働きと何らかの関係があるという考えを発展させます。彼は、波動関数の崩壊は、意識との相互作用によるものだと提案します。

Freeman Dyson

彼は、「こころ」は選択をする能力によって特徴づけられるのだが、それは、ある程度は、全ての電子にも内在する性質であると論じます。

Bohm

彼は、物質と意識の双方に適用可能な秩序があると論じます。それによって、両者のあいだの関係が説明できるだろうと示唆します。こころと物質は、より基本的な内的な秩序からの、外的な秩序への射影だと考えます。

ボームは、我々は物質を見ても、意識を理解するのを助ける何ものも見つけることはできないと主張します。

ある哲学者は、こうしたボームの考えは、次のような考えに自然に導かれると言っています。すなわち、

「論理的思考の物理的連関は、脳の古典的に記述可能なレベルに属するが、基本的な思考過程は、量子論的に記述可能なレベルに属する」と。

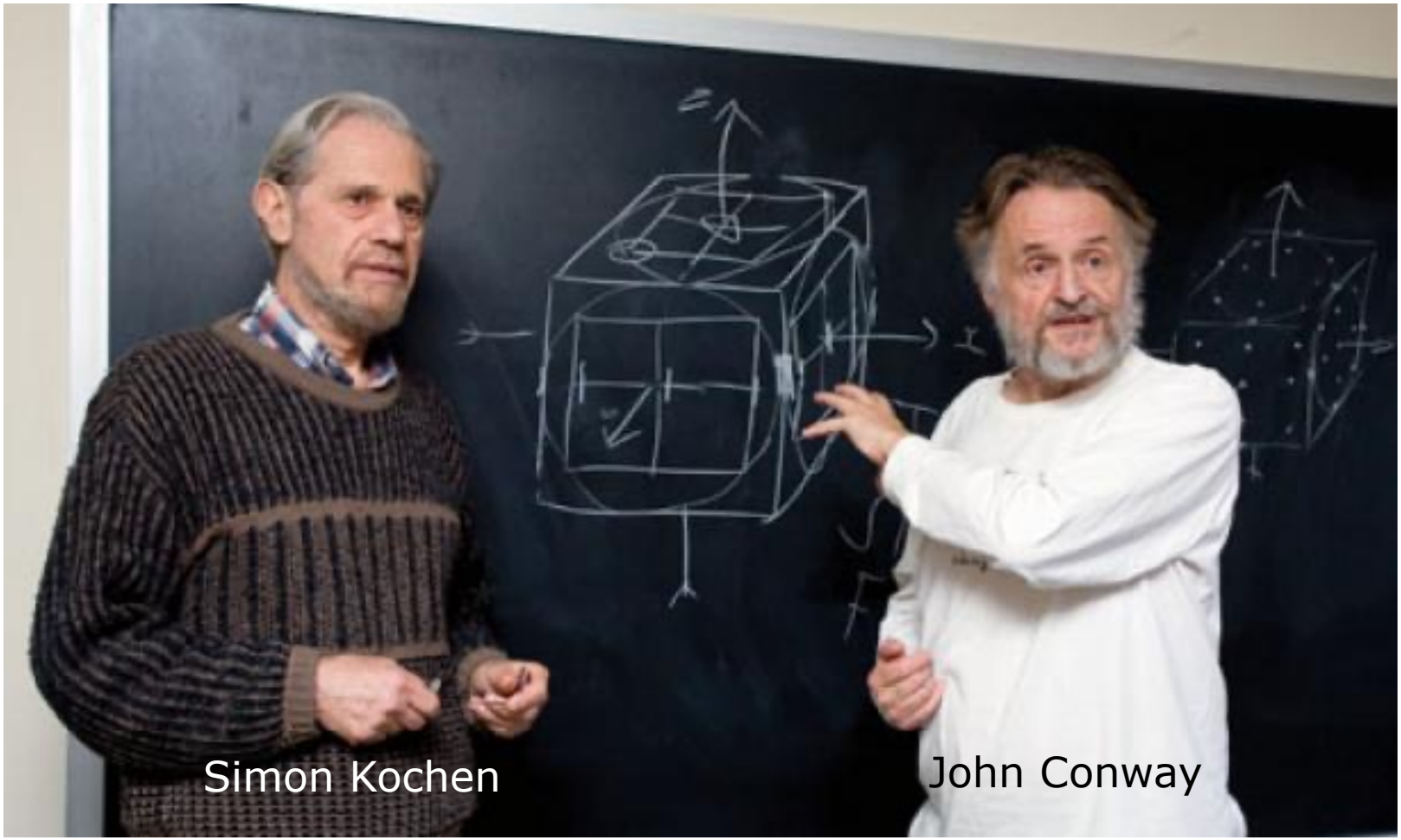
Penrose

ペンローズの議論は、ゲーデルの不完全性定理に拠ったものです。彼の意識についての最初の本「皇帝の新しいこころ」(1989年)で、次のような主張を行います。形式的システムは自身の無矛盾性を証明できないが、人間の数学者は、ゲーデルの定理のように、証明できないことを証明できる。このことは、人間の数学者は、形式的な証明システムではないし、彼が行っているのは、計算可能なアルゴリズムを実行している訳ではないことを意味する

「しかし、脳のどこか深いところで、一つの量子の変化を感じることのできる細胞が見つかるかもしれないと考えることができる。もしも、そうだということが証明されれば、量子力学は脳の活動に深く関わっていることになる。

ペンローズは、波動関数の崩壊は、人間の脳の非計算可能的なプロセスの、唯一の可能な物理的基礎となる。」

Conway と Kochen
「量子は自由意志を持つ」



2009/07/15 Conway Lectures
https://paw.princeton.edu/article/conway-lectures

Conway と Kochenは、「量子は自由意志を持つ」と論じた。

ここでの「量子の自由意志」とは、ある時点での量子の振る舞いの観測は、その量子の過去の振る舞いでは決定されないオープンな結果を返すということ。

過去の状態の関数として現在をとらえるのが、決定論なのだが、量子はそういう決定論には従わないランダムな振る舞いをする。

ランダムさの定義の難しさ

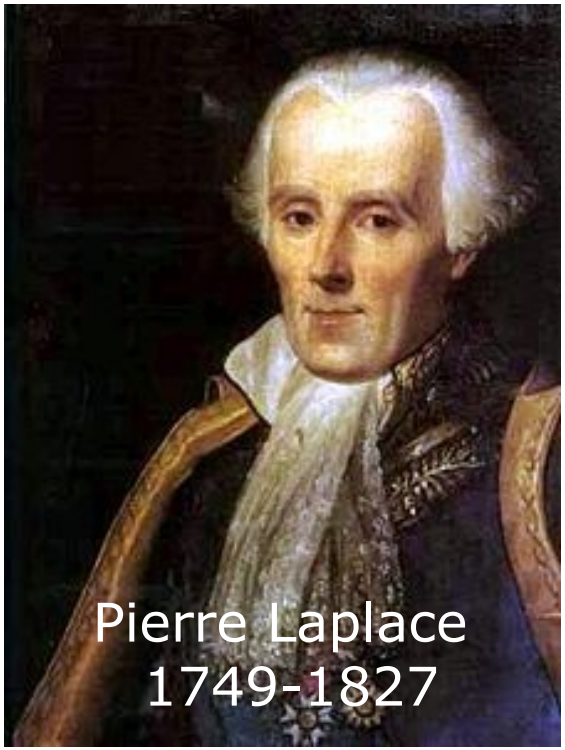
「ランダムさ」の定義
ラプラス

ある数列は、規則性をほとんど含まないなら、ランダムである

0がn個続く数列は規則的である。これはいい。

π の並びは、規則的か？

あるいは、規則的でないものの例を、示せるのか？



「ランダムさ」の定義
フォン･ミーゼス

1919年、フォン･ミーゼスは、次の二つの条件を、数列の「ランダムさ」の定義として提案した。

1. Sの1項目からn項目までの部分列$S_{1:n}$での1の出現確率は、nが大きくなるにつれて一定の確率に近づく
2. ある関数によって選ばれた任意のSの部分列についても、その部分列での1の出現確率は、nが大きくなるにつれて一定の確率に近づく

これは正しいか？

「ランダムさ」の定義
フォン・ミーゼス

ただ、この「ランダムさ」の定義は、Sからその部分列S'を選ぶ関数に依存する。例えば、ミーゼスの条件2の関数として、Sの中の全ての0を返す関数を考えれば、この部分列は、条件1を満たさない。任意の関数が、「ランダムさ」の定義に使える訳ではない。

WardとChurchは、この関数に、(決定可能な)帰納的関数を使うことを提案する。

しかし、J. Villeは、こうした "von Mises-Wald-Church random sequences " は、すべての「ランダムさ」をカバー出来ないことを示した。

ランダムさ への新しいアプローチ

コロモゴロフの登場と
ランダムさへの新しいアプローチ

ランダムさへのアプローチは、1960年代の半ばに、旧ソヴィエトの数学者コロモゴロフとその学派が登場して、新しい段階に入りました。

「コロモゴロフ複雑性」を中心的な武器とした「アルゴリズム論的情報理論」は、チャーチ＝チューリングらの計算理論との接点を深化し（Chaitinの「不完全性定理」）、また、シャノンの情報理論との結びつきを探求します(「条件付きエントロピー」、「相互エントロピー」)。


Andrey Kolmogorov
1903-1987


Per Martin-Löf
1942-


Gregory Chaitin
1947-


Leonid Levin
1948-

コロモゴロフの登場と
ランダムさへの新しいアプローチ

ここでは、まず、彼らのランダムさへのアプローチの概略を、
次の三つのトピックで、紹介しようと思います。

- 「オブジェクトの記述」＝「プログラム」
- プログラムの長さへの注目
- 圧縮可能な情報と圧縮不可能な情報

「オブジェクトの記述」＝「プログラム」

ある認識の対象を「記述」するには、いろいろな方法があります。

- その定義を、記述する。
- その特徴を、記述する。
- その時間的変化を、記述する。
- その意味を、記述する。
- …

アルゴリズム論的情報理論では、少なくとも文字列として表現されるもの（それは、基本的には $\{0,1\}^*$ すなわち、0と1の並びに還元されるのですが）、その「記述」とは、その文字列を出力する「プログラム」に等しいと考えます。

プログラムの「長さ」への注目

私たちの認識の対象は、多様です。対象の「記述」＝「プログラム」だとしても、ある対象を「記述」する「プログラム」も多様です。

一般的にいって、対象が複雑であれば、その「記述」、すなわちその対象を出力する「プログラム」も複雑なものになり、対象が単純なものであれば、その対象を出力する「プログラム」も単純なものになるのは、容易に想像できると思います。

プログラムの複雑さを、そのプログラムの「長さ」で代表させましょう。そうすることで、我々の認識の対象に、その複雑さを表す尺度を導入できます。

プログラムの「長さ」への注目
コロモゴロフ複雑性

ある対象xを出力するプログラムは、複数あり得ます。この複数あるxを出力するプログラムで、最小の長さを持つものの長さを、xの「コロモゴロフ複雑性」と言います。

確認して欲しいのは、このように、xのコロモゴロフ複雑性は、一つの自然数だということです。

もうひとつ確認しておきましょう。「記述」＝「プログラム」のプログラムは、基本的には、$\{0,1\}^*$ すなわち、0と1の並びだと考えることが出来るということです。

圧縮可能なプログラム

同じxについて異なる二つの「記述」があったとしましょう。それは、xを出力とする異なる二つのプログラムp1とp2があるということです。

プログラム pの長さを |p|で表します。|p1| < |p2|だとしましょう。「xの記述」＝「プログラム」 p2は、「xの記述」＝「プログラム」 p1より長いということです。それは、同じxを出力するのに、プログラムp2は、プログラムp1より、冗長で無駄が多いということです。

こうした時、プログラムp2は、p1に「圧縮可能」だとしましょう。

圧縮可能なプログラム

先に見たような、プログラムが「圧縮可能」かのチェックを行い、それを短い圧縮されたプログラムに置き換えることは、プログラマーが、普段よく行っていることです。

それでは、こうしたプログラムの改善・圧縮は、どこまで可能でしょうか？

圧縮不可能なプログラム

nをxの「コロモゴロフ複雑性」としましょう。

「コロモゴロフ複雑性」いうのは、xを出力するプログラムの長さの最小のものですので、|p|＝n を達成したプログラムpは、これ以上短かなプログラムに圧縮することはできません。

プログラムには、より短いプログラムに圧縮可能なものと、それ以上の圧縮が不可能なプログラムの二種類が存在することになります。

「ランダムさ」を圧縮不可能なもの として特徴付ける

ここまでは、プログラムが「圧縮可能」か「圧縮不可能」かについて話してきましたが、おなじ論法で、ある対象xが「圧縮可能」か「圧縮不可能」かを、コロモゴロフ複雑性K(x)を利用して定義することができます。

次がコロモゴロフによる、xが「圧縮不可能」である条件です。

$$K(x) \geq |x| - c$$

コロモゴロフらは、この条件が、xがランダムであることを表すと考えます。

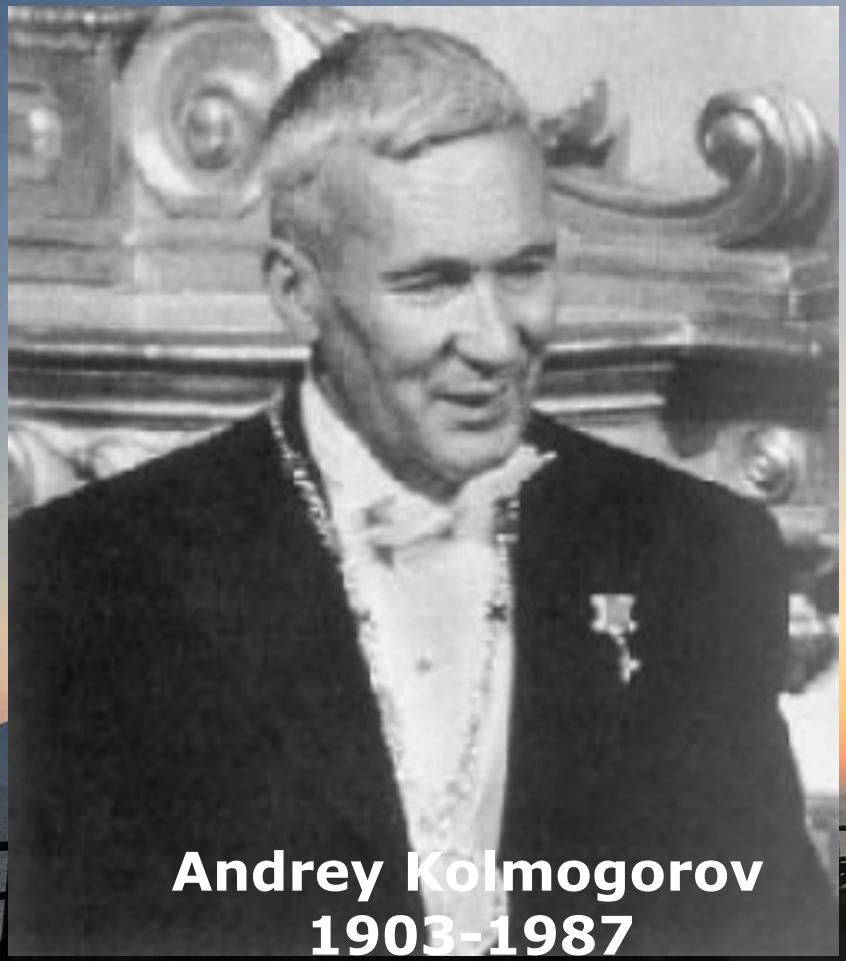
Part II

計算可能性理論と
コロモゴロフの複雑性

Part II
計算可能性理論とコロモゴロフの複雑性
Agenda

- コロモゴロフの複雑性
- 全ての計算可能な関数はプログラムで表現できる
- コロモゴロフ複雑性の「不変性」
- コロモゴロフ複雑性の「計算不能性」
- チャイティンの「不完全性定理」

コロモゴロフの複雑性



Andrey Kolmogorov
1903-1987

コロモゴロフの複雑性

ある出力Xを与えるプログラムの長さの最小値を、Xの「コロモロゴフの複雑性」と言います。「アルゴリズム論的情報理論」の出発点は、この「コロモロゴフの複雑性」です。

なぜ、「プログラムの長さの最小値」かといえば、同じ結果を返すいくらでも冗長なプログラムはかけるのですから、出力の「複雑さ」をプログラムの「大きさ」に一意に対応づけようとするのなら、その結果を返す「最も短いプログラム」の大きさを考えるのがいいことになるからです。

プログラムの実行とその出力

コロモゴロフの複雑性を、もう少しきちんと定式化するために、プログラムとその出力の関係をどう表現すればいいかを考えて見ましょう。そこには、プログラムの「実行」という考えが必要です。

プログラム pの出力が yだとします。このことは、ある実行環境のもとでプログラム pを「実行」すれば、出力 y が得られるということです。このことを、次のように表しましょう。

$$\varphi(p) = y$$

φ は、「ある実行環境でのプログラムの実行」を表しています。

コロモゴロフの複雑性の定義

この時、ある実行環境φのもとでのプログラム p の実行の出力 yのコロモゴロフの複雑性$K_\varphi(y)$は、次のように定義できます。

$$K_\varphi(y) = \min\{\, |p| : \varphi(p) = y \,\}$$

ここで、minは最小値、|p|は、プログラム p の長さを表しています。

少し形式的に

少し、形式的に関数の型を整理しましょう。

プログラム pを、0と1の有限のビット列 $\{0,1\}^*$とし、
プログラムの出力のクラスを$\mathcal{O}$で表すと、

$$\varphi : \{0,1\}^* \longrightarrow \mathcal{O}$$
$$K_\varphi : \mathcal{O} \longrightarrow \mathbb{N}$$

です。

留意すべきこと

いくつか留意すべきことがあります。

ここでのプログラムpは入力を持っていません。

関数φは、すべてのビット列 $\{0,1\}^*$に対して定義されているわけではありません。

出力yのコロモゴロフの複雑性$K_\varphi(y)$は、強く実行環境φに依存しているように見えます。実際、どのプログラム言語を使うかで、プログラムの長さは変わりますから。

ただ、複雑性は、φに依存しないことを示すことができるのです。それは重要な性質です。それについては後で述べます。

すべての計算可能な関数は、
プログラムで表現できる

コロモゴロフの複雑性の定義から、関数とプログラムの関係を考える

ここでは、関数とプログラムの関係を、コロモゴロフの複雑性の定義から、考えようと思います。

コロモゴロフの複雑性は、「yを出力するプログラムpの最小の長さ」として、次のように定義されました。

$$K_\varphi(y) = \min\{\, |p| : \varphi(p) = y \,\}$$

コロモゴロフの複雑性の定義中の関数の型

この定義の中に出てくる関数の型を確認しておきましょう。
プログラムのクラスを$\mathcal{P}$とし、プログラムの出力のクラスを$\mathcal{O}$としましょう。

$$\varphi : \mathcal{P} \longrightarrow \mathcal{O}$$

/* $\varphi(p) = y$ */

/* φは、プログラム p:$\mathcal{P}$を出力 y:$\mathcal{O}$に変える関数 */

$$K_\varphi : \mathcal{O} \longrightarrow \mathbb{N}$$

/* $K_\varphi(y) = \min\{\, |p| : \varphi(p) = y \,\}$ */

/* K_φは、出力 y:$\mathcal{O}$を自然数$\mathbb{N}$に変える関数 */

すべての計算可能な関数は、プログラムで表現できる

ここで、すこし天下り的ですが、「すべての計算可能な関数は、プログラムで表現できる」と考えることにしましょう。

計算可能な関数の値は、関数に対応するあるプログラムがあって、そのプログラムを実行して得られる出力に等しいということです。

こうした主張を、「チャーチ＝チューリングの提言」と言います。

関数とプログラム

「チャーチ＝チューリングの提言」は、基本的には、関数とプログラムは同じものであることを主張しているのですが、関数とプログラムでは、言葉の使い方が異なります。

関数の「引数」を、プログラムでは「入力」と言います。
関数の「値」を、プログラムでは「出力」と言います。

まあ、でもこの言い換えは、自然なものです。

関数の値の計算とプログラムの実行

関数 f(x) という表現は、「関数 f(x) そのもの」という意味で用いられることもあれば、「関数 f(x) の値」という意味で用いられることもあります。それは曖昧です。

関数そのものと、関数の計算で得られた関数の値とは違うものです。二つを区別するために、fがxについての関数そのものであることを、次のように表します。

$$\lambda x.\, f\,(x)$$

もちろん、プログラムそのものと、そのプログラムの実行とは別のものです。

コロモゴロフの複雑性の定義の中の関数φ を、プログラムとして考える

関数とプログラムの対応の例として、コロモゴロフの複雑性の定義の中に出てくる、「プログラムpを出力yに変える関数φ」を、プログラムとして考えて見ましょう。

φの働きは、$\varphi(p) = y$で、入力のないプログラム p を、それを実行して、出力 y に変えることでした。

関数φに対応するプログラムを e としましょう。

この時、プログラムeの実行 は、入力として プログラム p を受け取って、出力 y を返すと考えることができます。

関数φを拡大する

$\varphi : \mathcal{P} \longrightarrow \mathcal{O}$

/* φ: (プログラム) → (出力) */

でしたが、このプログラムは、入力を持っているわけではありません。ここで、プログラムとそれへの入力を、プログラムを実行して出力に変える、もう少し一般的な、関数 Uを考えます。

入力のクラスを $\mathcal{I}$とすれば、Uの型は次のように表現できます。

$U : \mathcal{P} \times \mathcal{I} \longrightarrow \mathcal{O}$

/* U: (プログラム, 入力) → (出力) */

eの実行をUで表す

関数φに対応するプログラムを e とすると、プログラムeの実行は、プログラムe は、入力として プログラム p を受け取って、出力 y を返すのですから、Uを使えば、次のように書けます。

$$U(e,p) = \varphi(p) = y$$

関数Uの性質

このUは、面白い性質を持っています。

$U : \mathcal{P} \times \mathcal{I} \longrightarrow \mathcal{O}$ で、$U_c = \lambda i{:}\,\mathcal{I}.\,\mathrm{U(c, i)}$とすると、

/* U: (プログラム, 入力) ⟶ (出力) */

$\mathrm{U_c} : \mathcal{I} \longrightarrow \mathcal{O}$

/* U_c: (入力) ⟶ (出力) */

$\mathrm{U_c}$ は、入力を出力に変えるという、極めて一般的なプログラムの特徴を表現しています。

それについては、後でもっと詳しく見ていきます。

コロモゴロフ複雑性の「不変性」

コロモゴロフ複雑性は実行環境依存？

コロモゴロフ複雑性$K_\varphi(y)$の定義：「yを出力するプログラムpの最小の長さ」

$$K_\varphi(y) = \min\{\, |p| : \varphi(p) = y \,\}$$

の問題の一つは、この$K_\varphi(y)$が、プログラムを実行するφに強く依存していることです。

事実、プログラムが Pythonで書かれているか、Javaで書かれているか、Assemblerで書かれているか，…. で、同じyを出力するプログラムの長さは、それぞれ違います。

コロモゴロフ複雑性は実行環境依存？

ただ、任意のφに対して、次の不等式を満たすVが存在することを示すことが出来ます。

$$K_V(y) \leq K_\varphi(y) + c$$

ここで、cは定数です。（証明は、後に回します）

コロモゴロフ複雑性の「不変性定理」

$$K_V(y) \leq K_\varphi(y) + c$$

すべてのφに対して、この不等式を満たすVと定数cが存在することを、コロモゴロフ複雑性の「不変性定理」と呼びます。

この不等式を満たすVを「最適な」Vと呼ぶことにしましょう。

「不変性定理」は、φがいろいろ変わっても、K_φ の(正確に言えば $K_\varphi(y) + c$ の)最小値K_Vを与える「最適な」Vが存在することを主張しています。

コロモゴロフ複雑性 K(y) の定義

「不変性定理」で定義される「最適な」 V を使って、新しいコロモゴロフ複雑性 K(y)を次のように定義できます。

$$K(y) \equiv K_V(y)$$

この$K(y)$が、新しいコロモゴロフ複雑性の定義になります。

コロモゴロフ複雑性 K(y) の定義

$$K(y) \equiv K_V(y) \leq K_\varphi(y) + c$$

ですので、$K(y)$は、 $K_\varphi(y)$たちの取る値の最小値です。

もし、VとV′が共に「最適」であるなら、

$$|K_V(y) - K_{V'}(y)| \leq c$$

が成り立ちます。

定数cについて
コロモゴロフが語ったこと

コロモゴロフは、論文”Three approaches to the quantitative definition of information” 1965 の中でこの定数について、次のように述べています。

「もちろん、特定の関数Vを考えることで、この定数cに関連した非決定性を避けることはできる。しかし、あからさまな恣意性抜きに、そうしたことができるかは疑わしい。

しかし、合理的な異なる「最適」な関数が、「複雑性の評価」に導き、それが数万ビットではなく、数百ビットに収斂すると想定しなくてはならない。

よって、例えば、「戦争と平和」のテキストの複雑性のような量は、ユニークなものとして定義されると推測することができる。」

情報量への三つのアプローチ

コロモゴロフ複雑性の「計算不能性」

コロモゴロフ複雑性の「計算不能性」

コロモゴロフ複雑性には、重要な性質があります。

それは、コロモゴロフ複雑性は、xを出力するプログラムの最小の長さとして定義されるのですが、計算理論的には、それは「計算不能」です。

ここでは、そのことを見ていきましょう。

コロモゴロフ複雑性の「計算不能性」の証明
Step 1: $K\big(L(n)\big) \geq 2n$

つぎのような関数Lを考えます。

$L(n) = K(k) \geq 2n$ となるような最小のk

関数 L：$\mathbb{N} \rightarrow \mathcal{O}$ は、nに対して、そのコロモゴロフ複雑性$K(k)$が2n 以上となるような最小の kを返します。もし、Kがすべてのnについて計算可能だとすると、Lもすべてのnについて計算可能です。

ここで、すべてのnに対して、$K\big(L(n)\big) \geq 2n$ です。

すべてのnについて、$L(n)$は、$K(L(n)) \geq 2n$を
満たす最小のものである。

コロモゴロフ複雑性の「計算不能性」の証明
Step 2: $K_L\big(L(n)\big) \leq n$

ここで、ある　V：$\mathcal{O} \rightarrow \mathbb{N}$ が、「最適」であるとしましょう。この時、$K = K_V$ と書くことができます。前回見た「不変性定理」から、$K \leq K_L + c$ である定数cが存在することがわかります。

K_L の定義から、$K_L\big(L(n)\big) \leq n$ が成り立ちます。

$K_\varphi(y) = \min\{\,|p| : \varphi(p) = y\,\}$ で、 $\varphi = L$ として、

$K_L(L(n)) = \min\{\,|p| : L(p) = L(n)\,\}$

$p = n$ の場合、$\mathrm{L}(p) = L(n)$ が成り立つので、

$K_L\big(L(n)\big) \leq n$

コロモゴロフ複雑性の「計算不能性」の証明
Step 3: $n \leq c$

そうすると、次の不等式が成り立ちます。

$$2n \leq K\big(L(n)\big) \leq K_L\big(L(n)\big) + c \leq n + c$$

$$2n \leq n + c \text{ から } n \leq c$$

ただし、この不等式は、全ての n については成り立ちません。
$n > c$ なるnについては矛盾します。

Kがすべてのnについて計算可能だと仮定すると、矛盾が導かれるので、Kは計算可能ではありません。

チャイティンの不完全性定理

Gregory Chaitin
1947-

チャイティンの不完全性定理

コルモゴロフは、旧ソ連の数学者で、「コルモゴロフの複雑性」が定式化されたのは、1960年代半ばのこと。今から50年以上前の話だ。

「コルモゴロフの複雑性」が奇妙な性質を持つことを、驚くような形でまとめたのは、チャイティン(G. J. Chaitin)である。

「どのような特定のビット列をとっても、そのコルモゴロフの複雑性が L 以上であることを証明できないような数 L が存在する。」

これを「チャイティンの不完全性定理」という。

その複雑性が L 以上であることを証明できない Lが存在する

どう言うことか？

いくらでも複雑なものは存在する。どんな数に対しても、その数以上のコルモゴロフ複雑性を持つビット列は、無限に存在することを、我々は証明できる。

ただ、特定のビット列が与えられた時、そのコルモゴロフ複雑性が L 以上であることを証明できない Lが存在すると言うのだ。

複雑さには、壁がある

いくらでも複雑なものを、我々は、無限に考えることができる。複雑なビット列を考えて、それとは違うもっと複雑なビット列を考えて、もっと考えて、 …… これを無限に繰り返しても、具体的なビット列を考える限り、その複雑さは L を超えないと言うのだ。

まるで、孫悟空がいくら飛び回っても、お釈迦様の手のひらから飛び出すことができないように。複雑さを求めて、宇宙の果てまで来たつもりだったのに、そこには超えられない壁 L があることを見つけたと言うことだ。

「複雑さ」（我々が具体的に考えうる）には、乗り越えられない壁 L があるのだ。

Lは、驚くほど小さい

そういうと、Lは、とんでもなく巨大な数だろうと考えたくなるのだが、Lは、驚くほど小さいらしい。

compositionality というのは、複雑なものが単純なものから構成されるという性質のことだが、単純なものから複雑なものを構成するルールをが、Nビットで記述できるとすると、複雑さの壁 L は、

$$L <= N + 2359$$

程度だという。

正確に、Lの値を求めることはできないが、あるコンピューター科学者は、数キロバイトだと推測している。

小さい！

エレガントなプログラム

特定のビット列を出力するプログラムは、そのビット列がどんなに複雑でも、数キロバイトに収まるということで、皆さんが普段作っているプログラムが、数キロバイトになるはずだということではないので、ご安心を。

チャイティンは、こういう言い方もしている。

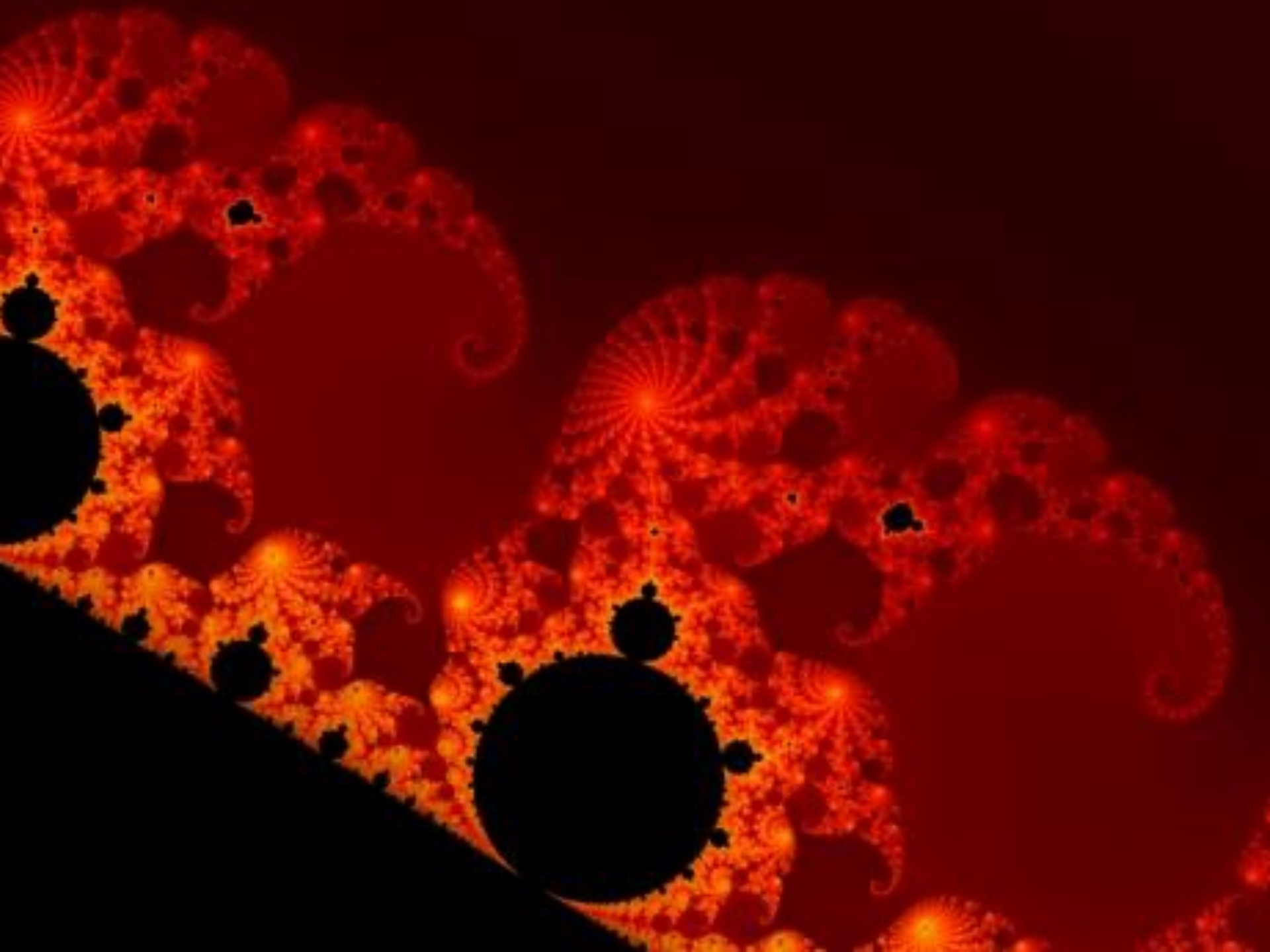
「あるプログラムを、それより小さいサイズのプログラムが同じ出力を行うことがない時、"エレガント"と呼ぶことにしよう。

あるプログラムが "エレガント" であることを、我々は証明できない。」

小さなプログラムでも、複雑なビット列を生み出すことができる

実際に、小さなプログラムでも、複雑なビット列を生み出すことができることの例を二つ紹介しよう。

一つ目は、マンデルブローのフラクタル図形の例である。この画像は、1ピクセルあたり24bitカラーを用いていて、全体で、162万bitあるのだが、これを生成するプログラムは、本当に小さい。



DemoScene Contest

二つ目の例は、4K以下のプログラムサイズという制限のもとで、作品の出来を競う DemoSceneのコンテストで優勝した動画。

見知らぬ惑星の描写だが、音楽含めて、プログラム・サイズは、たったの4Kだという。信じられない！

https://www.youtube.com/watch?v=I5CTFMuFvb0

DemoScene Revision

DemoSceneのコンテストは、現在は、名前を DemoScene Revision に変え、プロラムサイズ4K以下という制限以外にも、サイズ制限付きの複数のトラックで競技が行われている。

2021年度のコンテストの結果は、こちらを見て欲しい。

https://www.geeks3d.com/20210406/demoscene-revision-2021/

Part III

アルゴリズム論的情報理論

Part III
アルゴリズム論的情報理論
Agenda

- チャイティンのΩとバエズのエントロピー
- アルゴリズム論的情報量
- 二つのアルゴリズム論的情報量の関係
- アルゴリズム論的情報量に確率の概念を導入する
- 分配関数 Z – 統計力学の方法を振り返る」
- アルゴリズム論的情報理論への分配関数Zの応用
- Gibbs Ensemble と共役変数
- アルゴリズム論的熱力学

チャイティンのΩとバエズのエントロピー

John Baez
1961-

シャノンのエントロピーと コルモゴロフの複雑性

シャノンのエントロピー $H(X)$

$$H(X) = -\sum_{x \in X} p_x \log p_x$$

コルモゴロフの複雑性　$C(x)$

$C(x) = \{\, x$ を出力するプログラム pの最小の長さ $\}$

シャノンのエントロピーは、X上の確率分布について定義される量だが、コルモゴロフの複雑性は、一つの数について定義される量である。

シャノンの情報理論と
コルモゴロフの複雑性理論

シャノンの情報理論

シャノンの理論は、複数のオブジェクトの集まりをどのようにエンコードし、そうしたオブジェクトのシークエンスを如何に効率的に送るかについて関心を持っている。

例えば、アルファベットの集まりXがある確率分布に従う時、そのエントロピーH(X)を用いて、N文字のメッセージを。
NH(X)ビットに情報圧縮できることを示せる。

コルモゴロフの複雑性理論

コルモゴロフの複雑性理論は、一つのオブジェクトを、最も効率的にエンコードすることを考える。

あるオブジェクトの最小の記述と コルモゴロフの複雑性

Berryのパラドックスは、次のようなものである。

「30文字以下では定義されない最小の自然数B]

Bは、日本語では21文字で定義されている。

こうした「定義」「名前づけ」「記述」の曖昧さを避けるために、あるオブジェクトの「記述」はそれを出力するプログラムで与えられると考える。コルモゴロフの複雑性$C_U(x)$は、万能チューリングマシンUにプログラムpが与えられた時、xを出力するプログラムpのうち、最小のプログラムの長さ(|p|)である。

$$C_U(x) = \min_p \{|p| : U(p) = x\}$$

その時、pは、オブジェクトの最小の記述を与える。

「ランダムさ」を定義する

あるオブジェクトが、それより短い記述を持たない時、そのオブジェクトを「圧縮不可能」という。

あるビット列が「ランダム」だと判断されるのは、それが「圧縮不可能」な時である。

Per Martin-Lof　1966年
“The Definition of Random Sequences”

Charles Bennet 1979年
“On Random and Hard-to-Describe Numbers”

チャイティンのΩ

prefix-freeで記述されるチューリングマシンで、停止してFを計算可能なすべての集合P_Fを考える。

P_Fに属するプログラムpの長さを|p|とする時、次の和を、チャイティンのΩという。

$$\Omega_F = \sum_{p \in P_F} 2^{-|p|}$$

バエズのエントロピー　$S(n)$

次の式をバエズのエントロピーと呼ぶ。
logの内側は、チャイティンのΩに似ているが、すべての計算可能なpではなく、計算可能で出力がnとなるpについて和を取ったものである。

$$S(n) = -\log \sum_{\mathrm{p}\in P_F,\, \mathrm{F(p)}=n} 2^{-|p|}$$

アルゴリズム論的情報量

「アルゴリズム論的情報量」の第一の定義

ここでは、「アルゴリズム論的情報量」の定義を2つ与えます。

プログラム言語を固定します。ここで、プログラムは、有限のビット列で、プログラムが停止するなら、その出力も、有限のビット列です。

この時、ある有限なビット列の「アルゴリズム論的情報量」を、そのビット列を出力するプログラムの最小の長さで定義します。

この、ある自然数nの「アルゴリズム論的情報量」の第一の定義では、それが、nの「コロモゴロフ複雑性」に等しいと考えます。

通常の、シャノン的情報量の定義
information gain

ここでは、先に見た「アルゴリズム論的情報量」＝「コロモゴロフ複雑性」を、通常の、シャノン情報理論での「情報量」＝「シャノン・エントロピー」とを比較してみましょう。

あるイベントxが確率pで起きたとしましょう。あるイベントxが起きたことを知った時に、「獲得した情報量」を、シャノンの情理論では、

$$-\log p$$

で表します。

“information gain” とか “information content”、“surprise” あるいは「自己情報量」といわれる情報量です。

通常の、シャノン的情報量の定義
エントロピー

ただ、シャノンの情報理論で、通常、「情報量」と言われるのは、X上の確率分布Pが与えられ、$x \in X$が確率p_xで起きる時、次の式で与えられる量です。

$$S(P) = -\sum_{x \in X} p_x \log p_x$$

これをシャノン・エントロピー と呼びます。

ただ、この「情報量」＝「エントロピー」は、先に見た「アルゴリズム論的情報量」と似ているようには見えません。

通常のシャノンの「情報量」との関係を、もっと分かりやすくするために、「アルゴリズム論的情報量」の二番目の定義を導入することにしましょう。

二つのアルゴリズム論的情報量の関係

アルゴリズム論的情報理論xの二つの情報量の定義

Xをプログラムの集合、$V(x)$ をプログラムの長さ、
$N(x)$ をプログラムxの出力とすると、二つの情報量の定義は、次のようになります、

第一の定義（コロモゴロフ複雑性）

$$S_{first}(n) = \min_{x \in X} \{ V(x) : N(x) = n \}$$

第二の定義

$$S_{second}(n) = -\log \sum_{x \in X,\, N(x)=n} 2^{-V(x)}$$

$S_{first}(n)$と$S_{second}(n)$の関係

$S_{first}(n)$と$S_{second}(n)$の関係を考えてみましょう。

$$S_{second}(n) = -\log \sum_{x \in X,\, N(x)=n} 2^{-V(x)}$$

で、nを出力とするようなプログラムxが一つしかなかったとしましょう。この時、総和の項は一つしかないということです。

この時、

$$S_{second}(n) = -\log 2^{-V(x)} = V(x)$$

となります。

nを出力とするようなプログラムxが一つしかないというのですから、V(x)は、$N(x) = n$ を満たすプログラムの最小の長さになります。この時、

$S_{second}(n) = S_{first}(n)$ になります。

$S_{first}(n)$と$S_{second}(n)$の関係

もちろん、nを出力するプログラムは、複数存在し得ます。

この時、アルゴリズム論的情報量の第二の定義による、$S_{second}(n)$は、それらのプログラムの長さの項$-\log 2^{-V(x)}$が寄与しますので、その分、$S_{first}(n)$ より小さくなります。

$$S_{second}(n) \leq S_{first}(n)$$

Levinの定理

1974年に、Leonid Levinは、次のような定理を証明しました。すなわち、prefix free なプログラム言語については、すべてのnについて、次の式が成り立つような定数cが存在すると。

$$S_{second}(n) \geq S_{first}(n) - c$$

このことは、アルゴリズム論的情報量の定義として、第一の定義（コロモゴロフ複雑性）をとっても、第二のものをとっても、その違いは、高々、定数の範囲内に収まることを意味します。

今後、アルゴリズム論的情報量 S(n)の定義として、第二のものを使うことにしましょう。

prefix-free

引き続き、プログラムは、有限のビット列だと考えます。もちろん、すべての有限のビット列が停止するプログラムを与えるわけではありません。

Xをプログラムのビット列の集合とする時、$x \in X$なら

任意のビット列yについて、$xy \notin X$とします。

プログラム xで始まり、その後ろにyをつなげたxyは、プログラムのビット列にはならないということです。

この条件を満たす時、Xをprefix free と言います。

「アルゴリズム論的情報量」の第二の定義

Xが prefix freeなプログラムの集合で、$x \in X$である xの長さを$V(x)$で表すと、次の式が成り立ちます。

$$\sum_{x \in X} 2^{-V(x)} \leq 1$$

この時、次の式で、「アルゴリズム論的情報量」 S(n)を定義します。

$$S(n) = -\log \sum_{x \in X,\, N(x)=n} 2^{-V(x)}$$

「アルゴリズム論的情報量」の第二の定義 S(n)

$$S(n) = -\log \sum_{x \in X,\, N(x)=n} 2^{-V(x)}$$

要するに、S(n)は、nを出力するすべてのプログラムxを考え、そのプログラムの長さV(x)について、$2^{-V(x)}$を足しあわせたものです。

この和は、prefix free なXについては、次の式が成り立っていますので、収束します。

$$\sum_{x \in X} 2^{-V(x)} \leq 1$$

アルゴリズム論的情報量に確率の概念を導入する

アルゴリズム論的情報量と シャノン的情報量の違い

ここでは、まず、シャノン的情報量とアルゴリズム論的情報量を比較してみましょう。いずれも $-\log X$ という形をしています。

シャノンの*Information Gain*: IG

$$IG(p_n) = -\log p_n$$

アルゴリズム論的情報量

$$S(n) = -\log \sum_{x \in X,\, N(x)=n} 2^{-V(x)}$$

アルゴリズム論的情報量と シャノン的情報量の違い

ただ、$-\log X$のXの部分は、ずいぶん違っています。シャノン的情報量の場合のpは$0 \leq p \leq 1$の「確率」と定義されていますが、アルゴリズム論的情報量ではそうではありません。

シャノンの*Information Gain*: IG

$$IG(p_n) = -\log p_n$$

確率として定義されている
$0 \leq p \leq 1$

アルゴリズム論的情報量

$$S(n) = -\log \sum_{x \in X,\, N(x)=n} 2^{-V(x)}$$

確率ではない

アルゴリズム論的情報量に確率の概念を導入する

アルゴリズム論的情報量には、確率の概念がかけています。
そのままでは、アルゴリズム論的情報量と、シャノンの情報量の対応づけは難しいです。

ただ、うまい方法があります。

アルゴリズム論的情報量の対数の内側の量のさらにシグマの内側の量について、ある定数Zを選んで、次のような量p_nを定義します。

$$p_n = \frac{1}{Z} 2^{-V(x)}$$

確率ではない

アルゴリズム論的情報量に確率の概念を導入する

この時、Zを次のように選びます。

$$Z = \sum_{x \in X,} 2^{-V(x)}$$

この時、

$$\sum_{n} p_n = 1$$

であることがわかります。

p_nは確率になっています。

アルゴリズム論的情報量とシャノン的情報量の対応

問題は

$$q(n) = \sum_{x \in X,\, N(x)=n} 2^{-V(x)}$$

の扱いです。

$$\delta_n(m) = \begin{cases} 1 & \text{if } m = n \\ 0 & \text{otherwise.} \end{cases}$$

とおいて、δ_nのqに対する相対エントロピーを計算します。

$$\begin{aligned} S(\delta_n, q) &= -\sum_{i \in \mathbb{N}} \delta_n(i) \ln \frac{\delta_n(i)}{q(i)} \\ &= -\ln \left(\sum_{x \in X\,:\, N(x)=n} e^{-\gamma |x|} \right) + \ln Z. \end{aligned}$$

アルゴリズム論的情報量と シャノン的情報量の対応

これから、

$$S(n) = -\log p_n + \log Z$$

nのアルゴリズム論的情報量であるS(n)は、定数の差を除けば、$-\log p_n$ に等しいことがわかります。

$-\log p_n$は、シャノンのinformation gainの情報量です。

どんなinformation gain かと言えば、確率分布Pに従う自然数の集合から、ランダムに一つのnを選んだときに、得られる情報が、$-\log p_n$です。

アルゴリズム論的情報量と
シャノン的情報量は対応する

次のように、アルゴリズム論的情報量とシャノン的情報量の対応を考えることができます。

事前には、ある数nがランダムに選ばれたプログラムの出力であることしか知らないとしましょう。

nのアルゴリズム論的情報量は、プログラムの選択をおこなうことで、この数について我々が学ぶ、シャノン的information gain に等しい。

分配関数 Z
– 統計力学の方法を振り返る

Zとは何か？（1）
チャイティンのΩ

このZと同じ形のものを、以前に見ています。

prefix-freeで記述されるチューリングマシンで、停止してFを計算可能なすべての集合P_Fを考えます。

P_Fに属するプログラムpの長さを|p|とする時、次の和を、チャイティンのΩといいます。

$$\Omega_F = \sum_{p \in P_F} 2^{-|p|}$$

Z とは何か？（2）
分配関数

Zはまた、統計力学でエントロピーが最大になる条件を求めるときに導入された分配関数Zと同じものです。

ここでは、その方法を振り返っておきましょう。

以前のマルゼミ「情報とエントロピー 」を参照ください。
https://www.marulabo.net/docs/info-entropy2/

分配関数
統計力学の方法

エントロピーが最大になる条件と拘束条件

エントロピーSが最大になる条件を考える。

$$S = -\sum_i p_i \log p_i$$

計算の都合上、次の$-S$が最小になる条件を考える。

$$-S = \sum_i p_i \log p_i$$

ただし、次のような二つの拘束条件がある。

$$\sum_i p_i = 1$$

$$\sum_i p_i E_i = E$$

エントロピーが最大になる条件と拘束条件

二つの拘束条件は、次の形に書ける。

$$\sum_i p_i - 1 = 0$$

$$\sum_i p_i E_i - E = 0$$

この時、任意のα, βについて、[]の中はゼロだから、次の式が成り立つ。

$$-S = \sum_i p_i \log p_i$$

$$-S = \sum_i p_i \log p_i + \alpha \left[\sum_i p_i - 1 \right] + \beta \left[\sum_i p_i E_i - E \right]$$

ラグランジェの未定乗数法

$-S$は、p_i, α, βについての関数である。-Sがp_iについて最小値をとるのは、次の条件を満たす時である。

$$\frac{\partial(-S)}{\partial p_i} = 0$$

$$\frac{\partial(-S)}{\partial p_i} = \frac{\partial}{\partial p_i}\left\langle \sum_i p_i \log\, p_i + \alpha\left[\sum_i p_i - 1\right] + \beta\left[\sum_i p_i E_i - E\right]\right\rangle$$

最初の項は、$\frac{\partial\, x \log x}{\partial x} = \log x + 1$ を使うと$\log p_i + 1$ に、

二番目の項は αに、三番目の項は βE_iになる。

$$\frac{\partial(-S)}{\partial p_i} = \log \mathrm{p_i} + 1 + \alpha + \beta E_i = 0$$

これから、$\log p_i + 1 + \alpha + \beta E_i = 0$がわかる。

分配関数 Z

$$\log p_i + 1 + \alpha + \beta E_i = 0$$

$$\log p_i = -(1 + \alpha + \beta E_i)$$

だから、

$$p_i = e^{-(1+\alpha)} e^{-\beta E_i}$$

$Z = e^{1+\alpha}$ とすると、

$$p_i = \frac{1}{Z} e^{-\beta E_i}$$

分配関数 Z

これを

$$\sum_i p_i = 1$$

に代入すると、

$$\sum_i \frac{1}{Z} e^{-\beta E_i} = 1$$

よって、

$$\frac{1}{Z} \sum_i e^{-\beta E_i} = 1$$

これから、

$$Z = \sum_i e^{-\beta E_i}$$

アルゴリズム論的情報理論への分配関数Zの応用

分配関数Zとエントロピーの最大化

先に見たのは、系のエネルギーEが、各部分系にE_iとして分配されているとして、エントロピーが最大になる条件を求めるものでした。

確率が、次の式で与えられる時、

$$p_i = \frac{1}{Z} e^{-\beta E_i}$$

分配関数Zは

$$Z = \sum_i e^{-\beta E_i}$$

になります。

プログラムの確率？

先にプログラムxの確率を次のように定義しました。

$$q_x = \frac{1}{Z} 2^{-V(x)}$$

実は、プログラムの確率がこの形をしているという積極的な理由があるわけではありません。

ただ、次のような形を作れば、熱力学の手法が応用できるというのが、この形を選んだ一番大きな理由です。

$$p_x = \frac{1}{Z} e^{-\gamma V(x)}$$

$$Z = \sum_{x \in X} e^{-\gamma V(x)}$$

プログラムの確率の解釈

このp_xとZに、次のような解釈を与えることができます。

プログラムの集合上の確率分布Pのエントロピーを最大にする条件を考えます。拘束条件が、プログラムの長さだとすれば、その答えは、次のようになります。

$$p_x = \frac{1}{Z} e^{-\gamma V(x)}$$

ここで、次のZが、p_xの和が1となること、すなわち、p_xが確率であることを保証します。

$$Z = \sum_{x \in X} e^{-\gamma V(x)}$$

先のq_xは、$\gamma = \ln 2$ と置いたものです。

分配関数Zの利用の一般化

こうした、確率変数p_xと分配関数Zの関係は、一般的なものです。今まで、拘束条件として、EやVの期待値を一つだけとってきたのですが、複数の観測の期待値から、同じような導出が可能です。

$$p_x = \frac{1}{Z} e^{-\beta E(x) - \gamma V(x) - \delta N(x)}$$

ここでは、次のZが、p_xの和が1となること、すなわち、p_xが確率であることを保証します。

$$Z = \sum_{x \in X} e^{-\beta E(x) - \gamma V(x) - \delta N(x)}$$

Gibbs Ensemble と共役変数

Gibbs Ensemble

一般に、X上で定義された実数値をとる関数 $C_n(x)$があって、次のような確率分布が与えられるとき、それらを「ギブス・アンサンブル Gibbs Ensemble」と言います。

$$p(x) = \frac{1}{Z} e^{-(s_1 C_1(x) + \cdots + s_n C_n(x))}$$

この時の分配関数は、次の形になります。

$$Z = \sum_{x \in X} e^{-(s_1 C_1(x) + \cdots + s_n C_n(x))}$$

共役な変数と観測の期待値

この時、s_iを関数$C_i(x)$の共役な変数と言います。

関数$C_i(x)$と確率 $p(x)$は、関数$C_i(x)$の観測の期待値を$\overline{C_i}$ で表せば、次の式が成り立ちます。

$$\sum_{x \in X} p(x) C_i(x) = \overline{C_i}$$

エネルギーの期待値がEで与えられる場合

代表的な例が、エネルギーの期待値Eが与えられる場合です。

$$p_i = \frac{1}{Z} e^{-\beta E_i}$$

$$Z = \sum_i e^{-\beta E_i}$$

この関係式から、E_iに共役な変数βが、次の値を取ることを導くことができます。21/6/26マルゼミ「情報とエントロピー」を参照ください。https://www.marulabo.net/docs/info-entropy2/

$$\beta = \frac{1}{T}$$

古典的な熱力学での例

ガスの分子からなるある系について、エネルギーEだけでなく、体積V、圧力Pについても観測量の期待値が与えられている
すると、次の式が立ちます。

$$p_x = \frac{1}{Z} e^{-\frac{1}{T}(E(x)+PV(x)-\mu N(x))}$$

$$Z = \sum_{x \in X} e^{-\frac{1}{T}(E(x)+PV(x)-\mu N(x))}$$

古典的な熱力学での
観測量と共役変数の対応の例

観測量	共役変数
エネルギー: E	$\frac{1}{T}$
体積: V	$\frac{P}{T}$
分子数: N_i	$-\frac{\mu_i}{T}$

アルゴリズム論的熱力学

この節は、John BaezとMike Stayの論文
“Algorithmic Thermodynamics” に、依拠しています。
https://arxiv.org/abs/1010.2067

John Baezの
アルゴリズム論的熱力学

John Baez は、先に見た古典的な熱力学の観測量を、アルゴリズム論的情報理論の観測量で置き換えたモデルを考えました。

- E(x)を、プログラムxの実行時間の対数
- V(x)を、プログラムxの長さ
- N(x)を、プログラムxの出力

とします。

アルゴリズム論的情報理論の Gibbs Ensemble

この時、次のような分配関数を考えます。

$$Z = \sum_{x \in X} e^{-\frac{1}{T}(E(x)+PV(x)-\mu N(x))}$$

この和が収束する時、次のようなX上の確率分布p(x)を考えることができます。これはGibbs Ensemble です。

$$p_x = \frac{1}{Z} e^{-\frac{1}{T}(E(x)+PV(x)-\mu N(x))}$$

これらから、観測量 E(x), V(x), N(x) の共役変数 β, γ, δを使って、期待値 E,V,Nを定義できます。

$$\beta = \frac{1}{T}, \qquad \gamma = \frac{P}{T}, \qquad \delta = -\frac{\mu}{T}$$

とすると、

$$Z = \sum_{x \in X} e^{-\beta E(x) - \gamma V(x) - \delta N(x)}$$

ですので、

$$E = \sum_{x \in X} p(x) E(x) = -\frac{\partial}{\partial \beta} \ln Z$$

$$V = \sum_{x \in X} p(x) V(x) = -\frac{\partial}{\partial \gamma} \ln Z$$

$$N = \sum_{x \in X} p(x) N(x) = -\frac{\partial}{\partial \gamma} \ln Z$$

古典的な熱力学での観測量と共役変数の対応の例

観測量	共役変数
エネルギー: E	$\frac{1}{T}$
体積: V	$\frac{P}{T}$
分子数: N_i	$-\frac{\mu_i}{T}$

アルゴリズム論的情報理論での観測量と共役変数の対応の例

観測量	共役変数
実行時間の対数:E	$\frac{1}{T}$
プログラムの長さ:V	$\frac{P}{T}$
プログラムの出力:N	$-\frac{\mu}{T}$

注意！

ここでの、E(x), V(x), N(x) の選択は、必然的なものではありません。ある意味、恣意的です。

用語については、紛らわしいかもしれないですが、古典的な熱力学のGibbs Ensembleの例を流用しています。

重要なことは、、、古典的な熱力学と同じ方法で、アルゴリズム論的熱力学が構成可能なことを示せるということです。それが、Gibbs Ensembleの方法の強みです。

また、計算時間、プログラムの長さ、プログラムの出力という観測量は、アルゴリズム論的情報理論にとっては、重要なものです。

アナロジーの有効性？

さきには、「まぎらわしい」と言いましたが、次のように考えることもできます。

- アルゴリズム論的熱力学のE、すなわち、計算時間の対数は、古典的熱力学のガスの内部エネルギーに似ている。
- アルゴリズム論的熱力学のV、すなわち、プログラムの長さは、古典的熱力学の容器の体積に似ている。
- アルゴリズム論的熱力学のN、すなわち、プログラムの出力は、古典的熱力学の分子の数に似ている。

$dE = TdS - PdV + \mu dN$ もそのまま成り立ちます。

エントロピーは計算科学とも接点を持つ！