Lecture 1:

Probabilistic Notions of Kolmogorov Complexity

Igor Carboni Oliveira

University of Warwick

CIRM - Randomness, Information & Complexity

February/2024

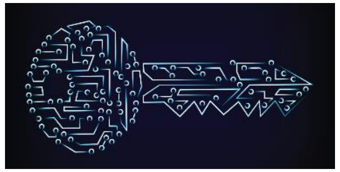
Plan for the Week

Lecture 1 (Monday)

Probabilistic Notions of (Time-Bounded) Kolmogorov Complexity

“**Unconditional** results & applications to average-case complexity”

Lecture 2 (Tuesday)



OWF P vs NP

Connections to Cryptography and Complexity Theory

“Major questions in complexity are **equivalent** to statements about Kolmogorov Complexity”

Lecture 3 (Thursday)

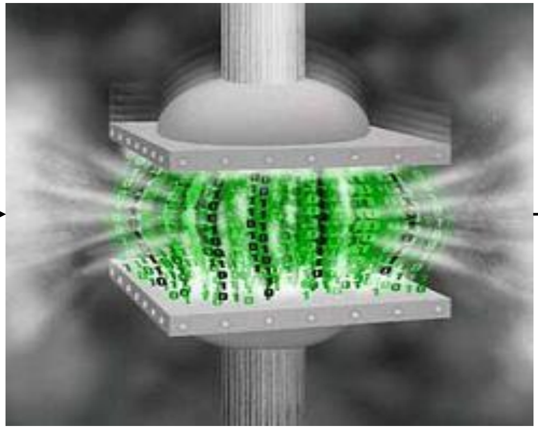
Connections to Algorithms (explicit constructions, generating primes, etc.)

“Existence of large primes with efficient short descriptions”

Kolmogorov Complexity

string x

01001101 … 0111010111



short encoding of x

010001110

$\mathrm{K}(x)$ = minimum **length** of a **program** M that outputs x

Formal Definition:

Let U be a Turing machine.

$$\mathsf{K}_U(x) = \min_{M \in \{0,1\}^*} \{|M| \; : \; U(M) \text{ outputs } x\}.$$

We formally define $\mathsf{K}(x)$ with respect to a fixed U (time-efficient universal machine)

Formal Definition:

Let U be a Turing machine.

$$\mathsf{K}_U(x) = \min_{M \in \{0,1\}^*} \{|M| \,:\, U(M) \text{ outputs } x\}.$$

We formally define $\mathsf{K}(x)$ with respect to a fixed U (time-efficient universal machine)

For simplicity, we abuse notation and refer to M directly.

Time-Bounded Kolmogorov Complexity

Kolmogorov complexity:

$$\boxed{\mathrm{K}(x) = \min_{M} \{|M| : M \text{ outputs } x\}}$$

Time-Bounded Kolmogorov Complexity

Kolmogorov complexity:

$$\mathrm{K}(x) = \min_{M} \{|M| : M \text{ outputs } x\}$$

Levin Kolmogorov complexity:

$$\mathrm{Kt}(x) = \min_{M,t} \{|M| + \log t : M \text{ outputs } x \text{ within } t \text{ steps}\}$$

Time-Bounded Kolmogorov Complexity

Kolmogorov complexity:

$$\mathrm{K}(x) = \min_{M} \{|M| : M \text{ outputs } x\}$$

Levin Kolmogorov complexity:

$$\mathrm{Kt}(x) = \min_{M,t} \{|M| + \log t : M \text{ outputs } x \text{ within } t \text{ steps}\}$$

t-time-bounded Kolmogorov complexity:

$$\mathrm{K}^t(x) = \min_{M} \{|M| : M \text{ outputs } x \text{ within } t(|x|) \text{ steps}\}$$

Despite the usefulness of time-bounded Kolmogorov complexity, many basic questions remain open:

Is it computationally hard to compute Kt(x)?

Do classical results in Kolmogorov complexity survive in the time-bounded setting?

Do natural objects (e.g., prime numbers) have small K^t or Kt complexity?

Despite the usefulness of time-bounded Kolmogorov complexity, many basic questions remain open:

Is it computationally hard to compute Kt(x)?

Do classical results in Kolmogorov complexity survive in the time-bounded setting?

Do natural objects (e.g., prime numbers) have small K^t or Kt complexity?

A more recent theory of probabilistic Kolmogorov complexity provides new insights.

Overview of this lecture

rKt rK^t pK^t

Probabilistic notions and some recent advances

Probabilistic versus deterministic

Two applications of pK^t to average-case complexity:

1. Worst-case complexity of easy-on-average problems
2. Worst-case to average-case reductions

Overview of this lecture

rKt rK^t pK^t

Probabilistic notions and some recent advances

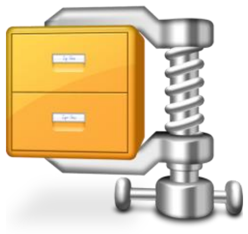
Probabilistic versus deterministic

Two applications of pK^t to average-case complexity:

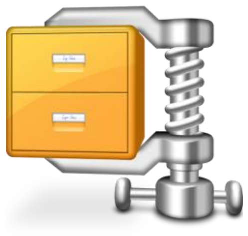
1. Worst-case complexity of easy-on-average problems

2. Worst-case to average-case reductions

Probabilistic Notions of Kolmogorov Complexity



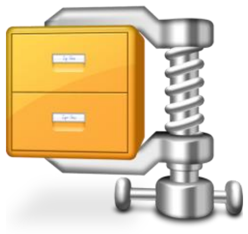
Probabilistic Notions of Kolmogorov Complexity



Definitions inspired by the notion of **pseudodeterministic algorithm:**

A **randomized algorithm** that produces the **same output string w.h.p.**

Probabilistic Notions of Kolmogorov Complexity



Definitions inspired by the notion of **pseudodeterministic algorithm:**

A **randomized algorithm** that produces the **same output string w.h.p.**

In Kolmogorov complexity terminology: **probabilistic decompression**

rKt complexity

$\mathsf{Kt} \longrightarrow \mathsf{rKt}$

Recall Levin Kolmogorov complexity:

$$\boxed{\mathrm{Kt}(x) = \min_{M,\, t} \{\, |M| + \log t \,:\, M \text{ outputs } x \text{ in } \leq t \text{ steps} \,\}}$$

rKt complexity

Kt $\longrightarrow$ rKt

Recall Levin Kolmogorov complexity:

$$\mathrm{Kt}(x) = \min_{M,\, t} \{\, |M| + \log t \;:\; M \text{ outputs } x \text{ in} \leq t \text{ steps} \,\}$$

Randomized Levin Kolmogorov complexity:

$$\mathrm{rKt}(x) = \min_{\text{Randomized } M,\, t} \{\, |M| + \log t \;:\; M \text{ outputs } x \text{ in} \leq t \text{ steps with probability} \geq \tfrac{2}{3} \,\}$$

An unconditional lower bound

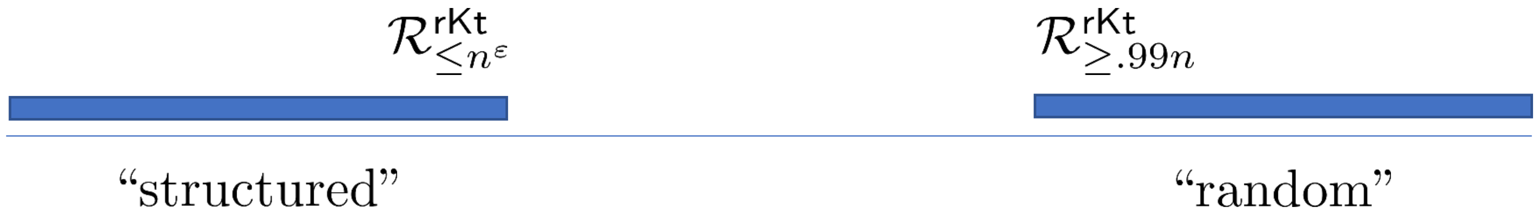
Is it hard to detect patterns?

$\mathcal{R}^{\mathsf{rKt}}_{\leq n^{\varepsilon}}$ $\mathcal{R}^{\mathsf{rKt}}_{\geq .99n}$

"structured" "random"

An unconditional lower bound

Is it hard to detect patterns?



Theorem [O'19]. $\forall \varepsilon > 0$, there is no randomised algorithm running in quasi-polynomial time that accepts strings in $\mathcal{R}^{\mathsf{rKt}}_{\leq n^{\varepsilon}}$ and rejects strings in $\mathcal{R}^{\mathsf{rKt}}_{\geq .99n}$

Fixed time bounds: rK^t

Recall t-time-bounded Kolmogorov complexity:

$$\boxed{\mathrm{K}^t(x) = \min_M \{|M| : M \text{ outputs } x \text{ within } t(|x|) \text{ steps}\}}$$

Fixed time bounds: rK^t

Recall t-time-bounded Kolmogorov complexity:

$$\mathrm{K}^t(x) = \min_{M} \{|M| : M \text{ outputs } x \text{ within } t(|x|) \text{ steps}\}$$

Randomized t-time-bounded Kolmogorov complexity:

$$\mathrm{rK}^t(x) = \min_{\text{Randomized } M} \{|M| : M \text{ runs in } t(|x|) \text{ steps and outputs } x \text{ with probability} \geq \tfrac{2}{3}\}$$

Succinct probabilistic representations

[Lagarias-Odlyzko'87] $\implies$ For every large n, there is an n-bit prime p_n with $\mathsf{Kt}(p_n) \leq \frac{n}{2} + o(n)$.

Recall: Open to show $\exists$ primes of Kt complexity $< n/2$.

Succinct probabilistic representations

[Lagarias-Odlyzko'87] $\implies$ For every large n, there is an n-bit prime p_n with $\mathsf{Kt}(p_n) \leq \frac{n}{2} + o(n)$.

Recall: Open to show $\exists$ primes of Kt complexity $< n/2$.

Theorem [O-Santhanam'17, O'19]. $\forall \varepsilon > 0$, for infinitely many values of n, $\exists$ n-bit prime p_n such that $\mathsf{rKt}(p_n) \leq n^{\varepsilon}$.

Succinct probabilistic representations

[Lagarias-Odlyzko'87] $\implies$ For every large n, there is an n-bit prime p_n with $\mathsf{Kt}(p_n) \leq \frac{n}{2} + o(n)$.

Recall: Open to show $\exists$ primes of Kt complexity $< n/2$.

Theorem [O-Santhanam'17, O'19]. $\forall \varepsilon > 0$, for infinitely many values of n, $\exists$ n-bit prime p_n such that $\mathsf{rKt}(p_n) \leq n^{\varepsilon}$.

running time can be exponential

Succinct probabilistic representations

[Lagarias-Odlyzko'87] $\implies$ For every large n, there is an n-bit prime p_n with $\mathsf{Kt}(p_n) \leq \frac{n}{2} + o(n)$.

Recall: Open to show $\exists$ primes of Kt complexity $< n/2$.

Theorem [O-Santhanam'17, O'19]. $\forall \varepsilon > 0$, for infinitely many values of n, $\exists$ n-bit prime p_n such that $\mathsf{rKt}(p_n) \leq n^{\varepsilon}$.

running time can be exponential

Theorem [Lu-O-Santhanam'21]. $\forall \varepsilon > 0$, for infinitely many values of n, $\exists$ n-bit prime q_n such that $\mathsf{rK}^{\mathrm{poly}}(q_n) \leq n^{\varepsilon}$.

succinct and efficient representation

pK^t

Probabilistic t-time-bounded Kolmogorov complexity [Goldberg-Kabanets-Lu-O'22]:

$$\mathrm{pK}^t(x) = \min_{k}\left\{ k : \Pr_{w \in \{0,1\}^{t(|x|)}} [\exists\, M \in \{0,1\}^k \text{ s.t. } M(w) \text{ outputs } x \text{ within } t(|x|) \text{ steps}] \geq \frac{2}{3} \right\}$$

For **most w** there exists a small program M that outputs x given w

pK^t

Probabilistic t-time-bounded Kolmogorov complexity [Goldberg-Kabanets-Lu-O'22]:

$$\mathrm{pK}^t(x) = \min_k \left\{ k : \Pr_{w \in \{0,1\}^{t(|x|)}} [\exists\, M \in \{0,1\}^k \text{ s.t. } M(w) \text{ outputs } x \text{ within } t(|x|) \text{ steps}] \geq \frac{2}{3} \right\}$$

For **most** w there exists a small program M that outputs x given w

Randomized t-time-bounded Kolmogorov complexity:

$$\mathrm{rK}^t(x) = \min_k \left\{ k : \exists\ t(|x|) \text{ time program } M \in \{0,1\}^k \text{ s.t. } \Pr_{\text{randomness of } M} [M \text{ outputs } x] \geq \frac{2}{3} \right\}$$

There exists a **fixed** small (randomized) program that outputs x **w.h.p over its internal randomness**

Overview of this lecture

rKt rK^t pK^t

☑ Probabilistic notions and some recent advances

Probabilistic versus deterministic

Two applications of pK^t to average-case complexity:

1. Worst-case complexity of easy-on-average problems

2. Worst-case to average-case reductions

Overview of this lecture

rKt rK^t pK^t

- [x] Probabilistic notions and some recent advances

Probabilistic versus deterministic

Two applications of pK^t to average-case complexity:

1. Worst-case complexity of easy-on-average problems
2. Worst-case to average-case reductions

Probabilistic Kolmogorov Complexity

$\mathrm{K}^{\mathrm{t}}(x) \leq k$

Resembles **NP**

$\mathrm{rK}^{\mathrm{t}}(x) \leq k$

Resembles **MA**

$\mathrm{pK}^{\mathrm{t}}(x) \leq k$

Resembles **AM**

$M \in \{0,1\}^k$

Sends a program $M \in \{0,1\}^k$

Runs M for $t(|x|)$ steps to recover x

Probabilistic Kolmogorov Complexity

$\mathrm{K}^{\mathrm{t}}(x) \leq k$

Resembles **NP**

$\mathrm{rK}^{\mathrm{t}}(x) \leq k$

Resembles **MA**

$\mathrm{pK}^{\mathrm{t}}(x) \leq k$

Resembles **AM**

Sends a **randomized** program $M \in \{0,1\}^k$

$M \in \{0,1\}^k$

Runs M for $t(|x|)$ steps to recover x **with high probability**

Probabilistic Kolmogorov Complexity

$\mathrm{K}^{\mathrm{t}}(x) \leq k$

Resembles **NP**

$\mathrm{rK}^{\mathrm{t}}(x) \leq k$

Resembles **MA**

$\mathrm{pK}^{\mathrm{t}}(x) \leq k$

Resembles **AM**

Shared randomness $w \in \{0,1\}^t$

$M \in \{0,1\}^k$ $\longrightarrow$

Sends a program $M \in \{0,1\}^k$, **based on** w

Runs $M(w)$ for $t(|x|)$ steps to recover x

Time-Bounded Kolmogorov Complexity

Proposition: For every x and t,

$$\boxed{\mathrm{K}(x) \lesssim \mathrm{pK}^{\mathrm{t}}(x) \leq \mathrm{rK}^{\mathrm{t}}(x) \leq \mathrm{K}^{\mathrm{t}}(x)}$$

AM $\supseteq$ **MA** $\supseteq$ **NP**

Time-Bounded Kolmogorov Complexity

Proposition: For every x and t,

$$\boxed{\mathrm{K}(x) \lesssim \mathrm{pK}^{\mathrm{t}}(x) \leq \mathrm{rK}^{\mathrm{t}}(x) \leq \mathrm{K}^{\mathrm{t}}(x)}$$

AM $\supseteq$ **MA** $\supseteq$ **NP**

Proposition: For every x and t,

$\mathrm{K}^{\mathrm{poly}(t)}(x) \leq \mathrm{rK}^{t}(x) + O(\log t)$ if **E** $\not\subseteq \mathrm{i.o.}\,\mathbf{SIZE}\left[2^{\Omega(n)}\right]$ — Derandomizing **MA** (to **NP**)

$\mathrm{K}^{\mathrm{poly}(t)}(x) \leq \mathrm{pK}^{t}(x) + O(\log t)$ if **E** $\not\subseteq \mathrm{i.o.}\,\mathbf{NSIZE}\left[2^{\Omega(n)}\right]$ — Derandomizing **AM** (to **NP**)

$\mathrm{rK}^{\mathrm{poly}(t)}(x) \leq \mathrm{pK}^{t}(x) + O(\log t)$ if **BPE** $\not\subseteq \mathrm{i.o.}\,\mathbf{NSIZE}\left[2^{\Omega(n)}\right]$ — Converting **AM** to **MA**

Under strong circuit lower bound assumptions:

$$\mathsf{K}^{\mathsf{poly}}(x) \approx \mathsf{rK}^{\mathsf{poly}}(x) \approx \mathsf{pK}^{\mathsf{poly}}(x) \qquad \text{(up to } O(\log n) \text{ additive terms)}$$

$\Longrightarrow$ **Probabilistic theory** sheds light on classical time-bounded Kolm. complexity

Under strong circuit lower bound assumptions:

$$\mathsf{K}^{\mathsf{poly}}(x) \approx \mathsf{rK}^{\mathsf{poly}}(x) \approx \mathsf{pK}^{\mathsf{poly}}(x) \qquad \text{(up to } O(\log n) \text{ additive terms)}$$

$\Longrightarrow$ **Probabilistic theory** sheds light on classical time-bounded Kolm. complexity

But theory can be independently developed (unconditional results, simpler proofs, new applications, etc.)

Overview of this talk

rKt $\quad \mathrm{rK}^t \quad \mathrm{pK}^t$

- ☑ Probabilistic notions and some recent advances
- ☑ Probabilistic versus deterministic

Two applications of pK^t to average-case complexity:

1. Worst-case complexity of easy-on-average problems
2. Worst-case to average-case reductions

Overview of this talk

$\mathrm{rKt} \quad \mathrm{rK}^t \quad \mathrm{pK}^t$

☑ Probabilistic notions and some recent advances

☑ Probabilistic versus deterministic

Two applications of pK^t to average-case complexity:

1. Worst-case complexity of easy-on-average problems

2. Worst-case to average-case reductions

Average-case Complexity

A problem L is solvable in *average-case polynomial time* w.r.t a distribution family $D = \{D_n\}_n$ if there is a poly-time algorithm A such that:

- $\Pr_{x \sim D_n}\left[\mathrm{A}\left(x; 1^k\right) \neq L(x)\right] \leq 1/k,$
- $\mathrm{A}\left(x; 1^k\right) \in \{L(x), \perp\}$ for every x in $\mathrm{Support(D)}$

$\longrightarrow$ $(L, D) \in$ **AvgP**

Average-case Complexity

A problem L is solvable in *average-case polynomial time* w.r.t a distribution family $D = \{D_n\}_n$ if there is a poly-time algorithm A such that:

- $\Pr_{x \sim D_n}\left[\mathrm{A}\left(x; 1^k\right) \neq L(x)\right] \leq 1/k$,
- $\mathrm{A}\left(x; 1^k\right) \in \{L(x), \perp\}$ for every x in $\mathrm{Support(D)}$

$\longrightarrow$ $(L, D) \in$ **AvgP**

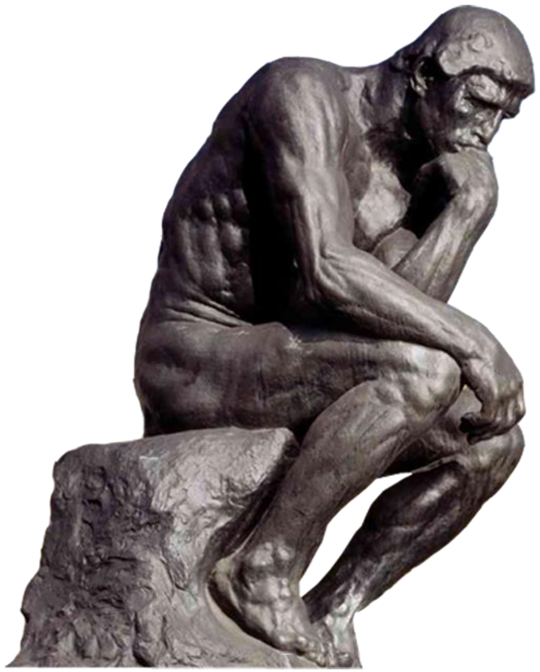
A language L is solvable in *randomized average-case polynomial time* w.r.t a distribution family $D = \{D_n\}_n$ if there is a poly-time randomized algorithm A such that:

- $\Pr_{\mathrm{A},\, x \sim D_n}\left[\mathrm{A}\left(x; 1^k\right) \neq L(x)\right] \leq 1/k$,
- $\mathrm{A}\left(x; 1^k\right) \in \{L(x), \perp\}$ w.h.p over A, for every x in $\mathrm{Support(D)}$

$\longrightarrow$ $(L, D) \in$ **AvgBPP**

Worst-case Running Times for Average-case Problems

If L is solvable in average-case polynomial time w.r.t to *all poly-time samplable distributions*, what can we say about the time needed to solve L in the *worst case*?



Worst-case Running Times for Average-case Problems

Theorem (Antunes-Fortnow'09): Under a strong derandomization assumption,
The following statements are equivalent for every language L:

- For every P-samplable distribution *D*, L can be solved in polynomial-time on average with respect to *D*.
- For every polynomial *t*, L is solvable by some algorithm that runs in time $2^{O(\mathrm{K}^t(x)-\mathrm{K}(x)+\log|x|)}$ on every input x.

$\mathrm{K}^t(x) - \mathrm{K}(x)$ is called the *t-computational depth* of x

Worst-case Running Times for Average-case Problems

$\mathsf{E} = \mathsf{DTIME}[2^{O(n)}]$ does not have $2^{o(n)}$ circuits with Σ_2^p oracle gates

Theorem (Antunes-Fortnow'09): Under a strong derandomization assumption,
The following statements are equivalent for every language L:

- For every P-samplable distribution *D*, L can be solved in polynomial-time on average with respect to *D*.
- For every polynomial *t*, L is solvable by some algorithm that runs in time $2^{O(\mathrm{K}^t(x) - \mathrm{K}(x) + \log|x|)}$ on every input x.

$\mathrm{K}^t(x) - \mathrm{K}(x)$ is called the *t-computational depth* of x

Worst-case Running Times for Average-case Problems

Theorem (Lu-O-Zimand'22):

The following statements are equivalent for every language L:

- For every P-samplable distribution *D*, L can be solved in polynomial-time on average with respect to *D*.
- For every polynomial *t*, L is solvable by some algorithm that runs in time $2^{O\left(\mathrm{pK}^t(x) - \mathrm{K}(x) + \log|x|\right)}$ on every input x.

$\mathrm{pK}^t(x) - \mathrm{K}(x)$ is the *t-probabilistic computational depth* of x

A useful ingredient of the proof

- For every **P-samplable distribution *D***, L can be solved in polynomial-time on average with respect to *D*.

- For every polynomial t, L is solvable by some algorithm that runs in time $2^{O\left(\mathrm{pK}^t(x)-\mathrm{K}(x)+\log|x|\right)}$ on every input x.

A useful ingredient of the proof

- For every **P-samplable distribution *D***, L can be solved in polynomial-time on average with respect to *D*.

$\Updownarrow$

(Informal) For every polynomial t, L can be solved in polynomial-time on average with respect to $\mu^t(x) = 2^{-\mathsf{pK}^t(x)}$

$\Updownarrow$

- For every polynomial *t*, L is solvable by some algorithm that runs in time $2^{O\left(\mathrm{pK}^t(x) - \mathrm{K}(x) + \log|x|\right)}$ on every input x.

A useful ingredient of the proof

- For every **P-samplable distribution *D***, L can be solved in polynomial-time on average with respect to *D*.

$$\Updownarrow$$

(Informal) For every polynomial t, L can be solved in polynomial-time on average with respect to $\mu^t(x) = 2^{-\mathsf{pK}^t(x)}$

$$\Updownarrow$$

- For every polynomial t, L is solvable by some algorithm that runs in time $2^{O\left(\mathrm{pK}^t(x) - \mathrm{K}(x) + \log|x|\right)}$ on every input x.

The link between **samplable distributions** and the **"universal" distribution** is obtained by a **"Coding Theorem"**

Optimal coding theorem for pK^t

[Lu-O-Zimand'22]

Coding Theorem in Kolmogorov Complexity

An object x can be sampled with probability δ $\Longrightarrow$ x admits a representation of length $\approx \log(1/\delta)$

We want an efficient version of the coding lemma.

Optimal coding theorem for pK^{t}

[Lu-O-Zimand'22]

Coding Theorem in Kolmogorov Complexity

An object x can be sampled with probability δ $\Longrightarrow$ x admits a representation of length $\approx \log(1/\delta)$

We want an efficient version of the coding lemma.

Theorem [Lu-Oliveira-Zimand'22]:

For every poly-time-samplable distribution $\{D_n\}$ over $\{0,1\}^n$, and every $x \in \mathrm{support}(D_n)$

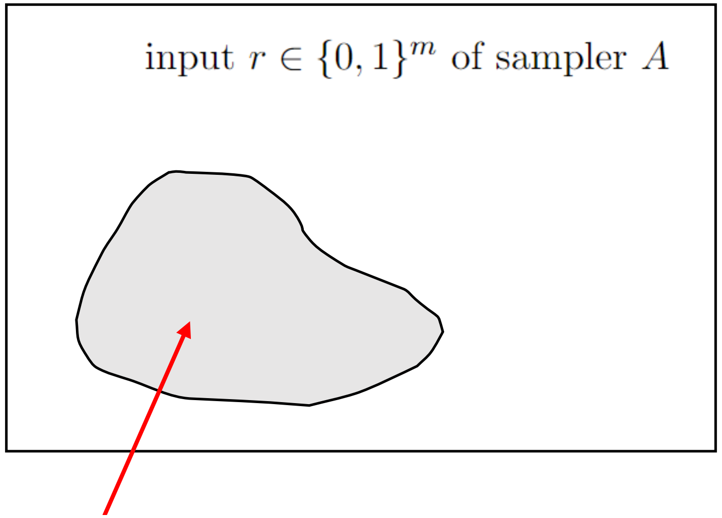
$$\mathrm{pK}^{\mathrm{poly}}(x) \leq \log\left(\frac{1}{D_n(x)}\right) + O(\log n)$$

Proof sketch: **Coding Theorem for pK^t** (adapting Antunes-Fortnow)

Let $A(1^n)$ be a poly-time sampler. Suppose it outputs $x \in \{0,1\}^n$ with probability δ.

Goal: $\mathsf{pK}^t(x) \leq \log(1/\delta) + O(\log n)$, where $t(n) = \mathsf{poly}(n)$

(For most random strings $\boldsymbol{w}$, the string x has a short description **given** $\boldsymbol{w}$)



δ-fraction of strings lead to x

Consider a random hash function $\boldsymbol{H} \colon \{0,1\}^k \to \{0,1\}^m$.

$\Pr_{\boldsymbol{H}}[\text{for no } z \in \{0,1\}^k \text{ we have } A(\boldsymbol{H}(z)) = x] \leq (1-\delta)^{2^k} \leq 1/10$

(if we let $k = \log(1/\delta) + 100$)

Claim. For most $\boldsymbol{H}$, x has a short description **given** $\boldsymbol{H}$

Issue: Efficiency (H can be of exponential size) (**Fix:** Efficiently derandomize construction of $\boldsymbol{H}$)

Back to equivalence result

- For every **P-samplable distribution *D***, L can be solved in polynomial-time on average with respect to *D*.

$\Updownarrow$ Optimal Coding Theorem for pK^{t}

(Informal) For every polynomial t, L can be solved in polynomial-time on average with respect to $\mu^t(x) = 2^{-\mathsf{pK}^t(x)}$

$\Updownarrow$ Time-bounded variant of result from Kolmogorov complexity

- For every polynomial *t*, L is solvable by some algorithm that runs in time $2^{O\left(\mathrm{pK}^t(x) - \mathrm{K}(x) + \log|x|\right)}$ on every input x.

Overview of this lecture

rKt $\quad$ rK^t $\quad$ pK^t

- ☑ Probabilistic notions and some recent advances
- ☑ Probabilistic versus deterministic

Two applications of pK^t to average-case complexity:

☑ 1. Worst-case complexity of easy-on-average problems

2. Worst-case to average-case reductions

Overview of this lecture

rKt $\quad \mathrm{rK}^t \quad \mathrm{pK}^t$

☑ Probabilistic notions and some recent advances

☑ Probabilistic versus deterministic

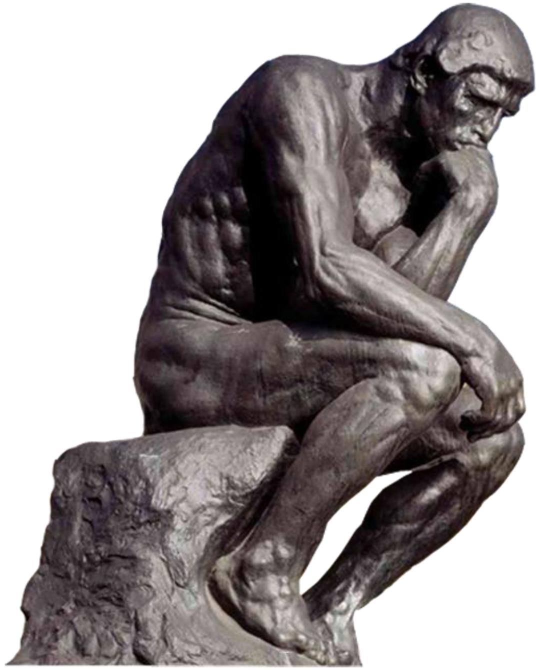
Two applications of pK^t to average-case complexity:

☑ 1. Worst-case complexity of easy-on-average problems

2. Worst-case to average-case reductions

Worst-case to Average-case Reductions

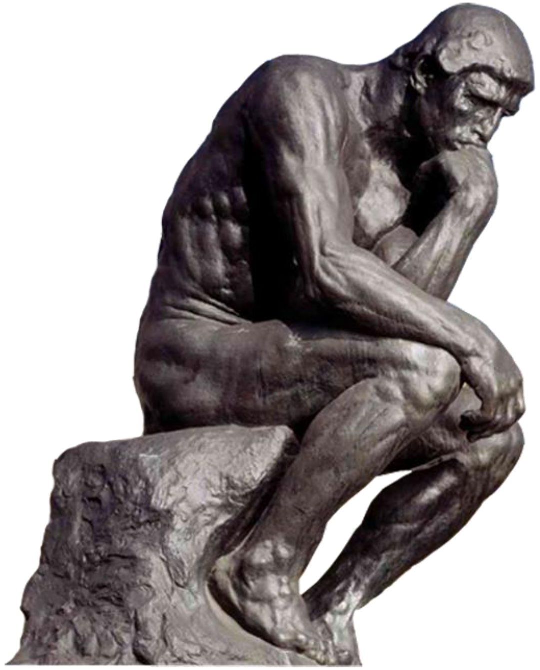
Is **NP** solvable in average-case polynomial time?



DistNP is the set of (L, D), where $L \in$ **NP** and D is poly-time samplable.

Is **DistNP** $\subseteq$ **AvgP**?

Worst-case to Average-case Reductions



Does *worst-case* hardness of **NP** imply *average-case* hardness of **NP**?

$$\boxed{\mathbf{NP} \neq \mathbf{P}} \xrightarrow{?} \boxed{\mathbf{DistNP} \not\subseteq \mathbf{AvgP}}$$

This is called a **worst-case** to **average-case** reduction.

Worst-case to Average-case Reductions

$\boxed{\textbf{DistNP} \subseteq \textbf{AvgP}} \xrightarrow{?} \boxed{\textbf{NP} = \textbf{P}}$

Theorem (Ben-David, Chor, Goldreich, Luby' 92):

$\boxed{\textbf{DistNP} \subseteq \textbf{AvgP} \text{ imples } \textbf{NP} \subseteq \mathbf{DTIME}\left[2^{O(n)}\right]}$

Worst-case to Average-case Reductions

DistNP $\subseteq$ **AvgP** $\xrightarrow{?}$ **NP** $=$ **P**

Theorem (Ben-David, Chor, Goldreich, Luby' 92):

DistNP $\subseteq$ **AvgP** imples **NP** $\subseteq$ $\mathbf{DTIME}\left[2^{O(n)}\right]$

Open to show **DistPH** $\subseteq$ **AvgP** implies **NP** $\subseteq$
DTIME$[2^{o(n)}]$
for nearly 30 years.

Worst-case to Average-case Reductions

Theorem (Hirahara'21):

DistNP ⊆ AvgP ➡ $\mathrm{UP} \subseteq \mathbf{DTIME}\left[2^{O(n/\log n)}\right]$

Extensions to NP and PH:

DistΣ_2 ⊆ **AvgP** imples **NP** ⊆ $\mathbf{DTIME}\left[2^{O(n/\log n)}\right]$

DistPH ⊆ **AvgP** imples **PH** ⊆ $\mathbf{DTIME}\left[2^{O(n/\log n)}\right]$

Worst-case to Average-case Reductions

Theorem (Hirahara'21):

$\text{DistNP} \subseteq \text{AvgP}$ ➡ $\text{UP} \subseteq \mathbf{DTIME}\left[2^{O(n/\log n)}\right]$

Extensions to NP and PH:

$\text{Dist}\Sigma_2 \subseteq \text{AvgP}$ implies $\text{NP} \subseteq \mathbf{DTIME}\left[2^{O(n/\log n)}\right]$

$\text{DistPH} \subseteq \text{AvgP}$ implies $\text{PH} \subseteq \mathbf{DTIME}\left[2^{O(n/\log n)}\right]$

Theorem (Goldberg-Kabanets-Lu-O'22):

$\text{DistNP} \subseteq \text{AvgBPP}$ ➡ $\text{UP} \subseteq \mathbf{RTIME}\left[2^{O(n/\log n)}\right]$

$\text{Dist}\Sigma_2 \subseteq \text{AvgBPP}$ implies $\text{NP} \subseteq \mathbf{RTIME}\left[2^{O(n/\log n)}\right]$

$\text{DistPH} \subseteq \text{AvgBPP}$ implies $\text{PH} \subseteq \mathbf{BPTIME}\left[2^{O(n/\log n)}\right]$

$\text{DistPH} \subseteq \text{AvgBPP}$ ➡ $\text{NP} \subseteq \mathbf{BPTIME}\left[2^{O(n/\log n)}\right]$

$\boxed{\text{DistPH} \subseteq \text{AvgBPP}} \Rightarrow \boxed{\text{NP} \subseteq \mathbf{BPTIME}\left[2^{O(n/\log n)}\right]}$

$\boxed{\text{DistPH} \subseteq \text{AvgBPP}}$ $\Rightarrow$ $\boxed{\text{NP} \subseteq \mathbf{BPTIME}\left[2^{O(n/\log n)}\right]}$

Recall the equivalence:

- For every P-samplable distribution *D*, L can be solved in polynomial-time on average with respect to *D*.
- For every polynomial *t*, L is solvable by some algorithm that runs in time $2^{O\left(\mathrm{pK}^t(x) - \mathrm{K}(x) + \log|x|\right)}$ **on every input x**.

Interested in the quantity $\mathrm{pK}^t(x) - \mathrm{K}(x)$

Exercise: There is x of length n such that $\mathrm{pK}^t(x) - \mathrm{K}(x) > n - \mathrm{C}\log n$

Perhaps in our application we can get an exponent that is less than $\mathrm{pK}^t(x) - \mathrm{K}(x)$?

DistPH ⊆ AvgBPP ➡ **NP ⊆ BPTIME**$\left[2^{O(n/\log n)}\right]$

Recall the equivalence:

- For every P-samplable distribution *D*, L can be solved in polynomial-time on average with respect to *D*.
- For every polynomial *t*, L is solvable by some algorithm that runs in time $2^{O\left(\mathrm{pK}^t(x)-\mathrm{K}(x)+\log|x|\right)}$ **on every input x**.

Term $\mathrm{K}(x)$ derived from the **Language Compression Theorem** for K

“If A is a decidable subset of $\{0,1\}^n$, then for every string y in A, K(y) < log |A| + O(log n) ”

Similarly to the **Coding Theorem**, perhaps we can establish **Language Compression** for **pK**$^{\text{poly}}$?

$\textbf{DistPH} \subseteq \textbf{AvgBPP}$ ➡ $\textbf{NP} \subseteq \mathbf{BPTIME}\left[2^{O(n/\log n)}\right]$

Recall the equivalence:

- For every P-samplable distribution *D*, L can be solved in polynomial-time on average with respect to *D*.
- For every polynomial *t*, L is solvable by some algorithm that runs in time $2^{O\left(\mathrm{pK}^t(x)-\mathrm{K}(x)+\log|x|\right)}$ **on every input x**.

Term $\mathrm{K}(x)$ derived from the **Language Compression Theorem** for K

"If A is a ~~decidable~~ subset of $\{0,1\}^n$, then for every string y in A, $\cancel{\mathrm{K(y)}}$ < log |A| + O(log n) "

"has complexity t" $\mathrm{pK}^{\mathrm{poly}(t)}$(y) < log |A| + O(log n)

Similarly to the Coding Theorem, perhaps we can establish Language Compression for $\mathbf{pK^{poly}}$?

DistPH ⊆ AvgBPP ➡ **NP ⊆ $\mathbf{BPTIME}\left[2^{O(n/\log n)}\right]$**

Recall the equivalence:

- For every P-samplable distribution *D*, L can be solved in polynomial-time on average with respect to *D*.
- For every polynomial *t*, L is solvable by some algorithm that runs in time $2^{O\left(\mathrm{pK}^t(x)-\mathrm{K}(x)+\log|x|\right)}$ **on every input x**.

This idea can improve the time bound to $2^{O\left(\mathrm{pK}^t(x)-\mathrm{pK}^{\mathrm{poly}(t)}(x)+\log|x|\right)}$

Language Compression for **pK$^{\text{poly}}$** is not known…

But it can be established for every set **A** in **NP** under the assumption that **DistPH ⊆ AvgBPP**.

(even for **A** in **AM**)

$\textbf{DistPH} \subseteq \textbf{AvgBPP}$ ➡ $\textbf{NP} \subseteq \mathbf{BPTIME}[2^{O(n/\log n)}]$

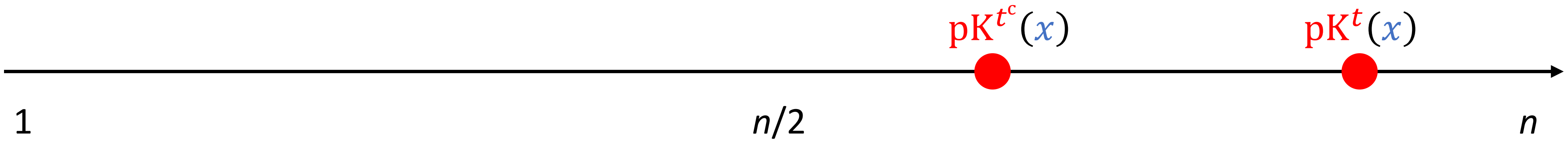
Fix **L** in **NP**. For every input x, and **for every polynomial** t,

We can decide if x is in **L** in time $2^{O\left(\mathrm{pK}^{t}(x) - \mathrm{pK}^{t^{c}}(x) + \log|x|\right)}$

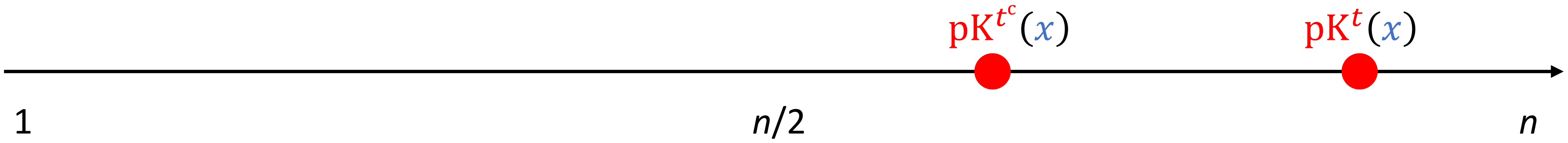
It remains to understand the bound $\mathrm{pK}^{t}(x) - \mathrm{pK}^{t^{c}}(x)$, for an arbitrary x.

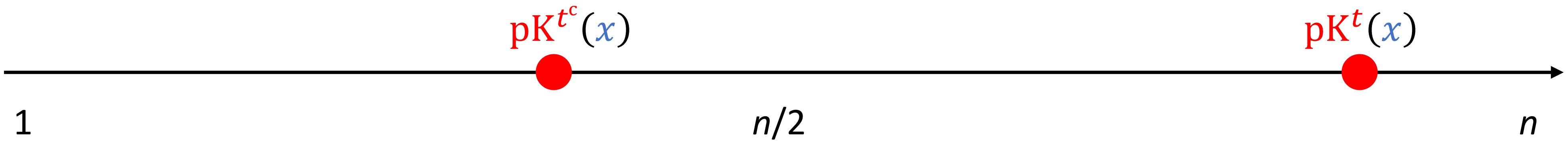
(**Crucial Point**: We can use different values of t in this upper bound!)

$$\mathrm{pK}^{t}(x) - \mathrm{pK}^{t^{c}}(x)$$

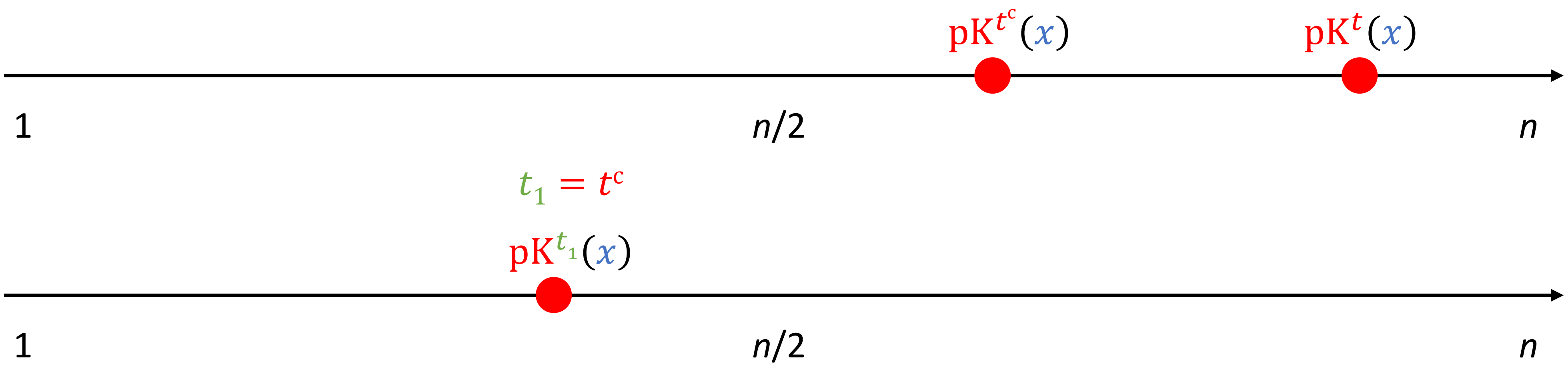


$$\mathrm{pK}^{t}(x) - \mathrm{pK}^{t^{c}}(x)$$

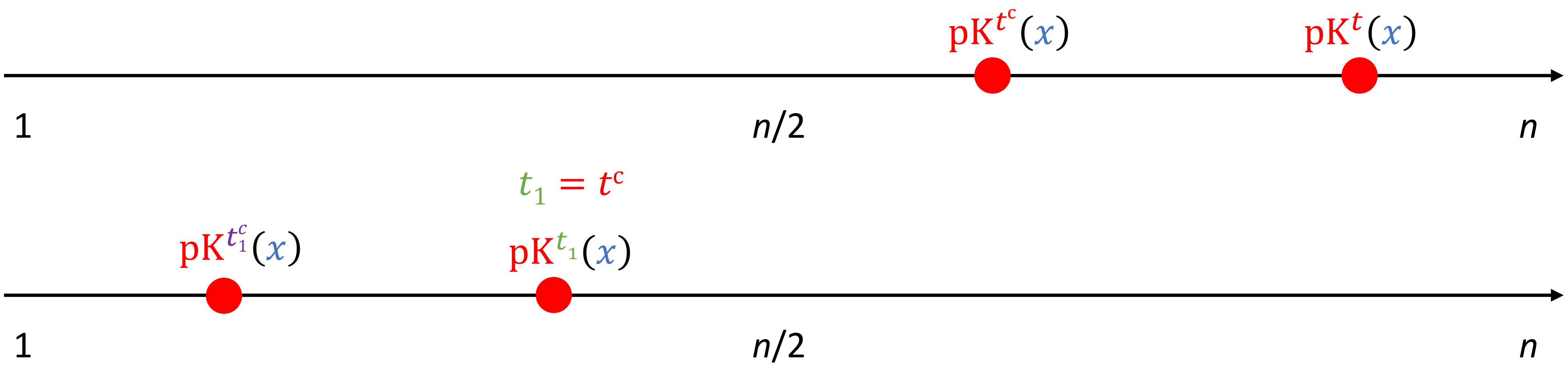




$$\mathrm{pK}^{t}(x) - \mathrm{pK}^{t^{\mathrm{c}}}(x)$$



$$\mathrm{pK}^{t}(x) - \mathrm{pK}^{t^c}(x)$$



By considering time bounds of the form t, poly(t), poly(poly(t)), …, the difference in **pK** complexity is small for some consecutive pair of time bounds.

Lemma. [Hirahara] For every $x \in \{0,1\}^n$, there is $t \in [n, 2^{o(n/\log n)}]$ such that

$$\mathrm{pK}^{t}(x) - \mathrm{pK}^{t^c}(x) = O(n/\log n).$$

Summary

rKt $\quad \mathrm{rK}^t \quad \mathrm{pK}^t$

- Probabilistic notions and some recent advances
- Probabilistic versus deterministic
- Two applications of pK^t to average-case complexity:
  - 1. Worst-case complexity of easy-on-average problems
  - 2. Worst-case to average-case reductions

Main Reference for Lecture 1:

Theory and Applications of Probabilistic Kolmogorov Complexity [Lu-O’22]

Bulletin of EATCS No 137 (The Computational Complexity Column), 2022.

Thank you