Stochasticity in Algorithmic Statistics for Polynomial Time

Alexey Milovanov
(The joint work with Nikolay Vereshchagin)

National Research University Higher School of Economics, almas239@gmail.com

3 July 2017, Mysore

Motivation

Motivation

Let x be a binary string. Our goal is to find a good explanation for x among all distribution on binary strings.

Motivation

Let x be a binary string. Our goal is to find a good explanation for x among all distribution on binary strings.

Example

Let x be n-natural number which is a square and has not other features.

Motivation

Let x be a binary string. Our goal is to find a good explanation for x among all distribution on binary strings.

Example

Let x be n-natural number which is a square and has not other features.
Hypothesis μ_1: the uniform distribution among all n-bit squares.

Motivation

Let x be a binary string. Our goal is to find a good explanation for x among all distribution on binary strings.

Example

Let x be n-natural number which is a square and has not other features.
Hypothesis μ_1: the uniform distribution among all n-bit squares.
Hypothesis μ_2: the uniform distribution among all n-bit numbers.

Motivation

Let x be a binary string. Our goal is to find a good explanation for x among all distribution on binary strings.

Example

Let x be n-natural number which is a square and has not other features.
Hypothesis μ_1: the uniform distribution among all n-bit squares.
Hypothesis μ_2: the uniform distribution among all n-bit numbers.

We can refute μ_2 since there is the set T of all n-bit squares such that $x \in T$ and $\mu_2(T) \ll 1$.

Motivation

Let x be a binary string. Our goal is to find a good explanation for x among all distribution on binary strings.

Example

Let x be n-natural number which is a square and has not other features.
Hypothesis μ_1: the uniform distribution among all n-bit squares.
Hypothesis μ_2: the uniform distribution among all n-bit numbers.

We can refute μ_2 since there is the set T of all n-bit squares such that $x \in T$ and $\mu_2(T) \ll 1$.
By the same reason we can refute μ_1 considering $\{x\}$.

Motivation

Let x be a binary string. Our goal is to find a good explanation for x among all distribution on binary strings.

Example

Let x be n-natural number which is a square and has not other features.
Hypothesis μ_1: the uniform distribution among all n-bit squares.
Hypothesis μ_2: the uniform distribution among all n-bit numbers.

We can refute μ_2 since there is the set T of all n-bit squares such that $x \in T$ and $\mu_2(T) \ll 1$.
By the same reason we can refute μ_1 considering $\{x\}$.
However the property 'to be equal x' is not *simple*: there is not a short program that decides a membership in $\{x\}$ in short time.

Motivation

Let x be a binary string. Our goal is to find a good explanation for x among all distribution on binary strings.

Example

Let x be n-natural number which is a square and has not other features.
Hypothesis μ_1: the uniform distribution among all n-bit squares.
Hypothesis μ_2: the uniform distribution among all n-bit numbers.

We can refute μ_2 since there is the set T of all n-bit squares such that $x \in T$ and $\mu_2(T) \ll 1$.
By the same reason we can refute μ_1 considering $\{x\}$.
However the property 'to be equal x' is not *simple*: there is not a short program that decides a membership in $\{x\}$ in short time.
(There exists such a program for T.)

Stochasticity

A probability distribution μ is called an *acceptable hypothesis* for x if there is no simple set $T \ni x$ with negligible $\mu(T)$.

Stochasticity

A probability distribution μ is called an *acceptable hypothesis* for x if there is no simple set $T \ni x$ with negligible $\mu(T)$.

Example

For every x the probability distribution μ such that $\mu(x) = 1$ is an acceptable hypothesis for x.

Stochasticity

A probability distribution μ is called an *acceptable hypothesis* for x if there is no simple set $T \ni x$ with negligible $\mu(T)$.

Example

For every x the probability distribution μ such that $\mu(x) = 1$ is an acceptable hypothesis for x.

A string x is called *stochastic* if it has simple and acceptable hypothesis.

Stochasticity

A probability distribution μ is called an *acceptable hypothesis* for x if there is no simple set $T \ni x$ with negligible $\mu(T)$.

Example

For every x the probability distribution μ such that $\mu(x) = 1$ is an acceptable hypothesis for x.

A string x is called *stochastic* if it has simple and acceptable hypothesis.
A rigorous definition of stochasticity requires 5 parameters (two for the simplicity of μ, two for the simlicity of T and one for negligibility of $\mu(T)$).

Stochasticity

A probability distribution μ is called an *acceptable hypothesis* for x if there is no simple set $T \ni x$ with negligible $\mu(T)$.

Example

For every x the probability distribution μ such that $\mu(x) = 1$ is an acceptable hypothesis for x.

A string x is called *stochastic* if it has simple and acceptable hypothesis.
A rigorous definition of stochasticity requires 5 parameters (two for the simplicity of μ, two for the simlicity of T and one for negligibility of $\mu(T)$).

The main question: are there non-stochastic strings with some reasonable parameters?

The parameters of stochasticity

The parameters of stochasticity

Definition

A distribution μ is called t_1, α-*simple* if it can be generated by a probabilistic program of length less than α in time at most t_1.

The parameters of stochasticity

Definition

A distribution μ is called t_1, α-*simple* if it can be generated by a probabilistic program of length less than α in time at most t_1.

Definition

A t_2, β, ε-*acceptable* hypothesis for a string x is a distribution μ such that $\mu(T) > \varepsilon$ for all $T \ni x$ recognized by a program of length less than β in at most t_2 steps for all inputs of length $|x|$.

The parameters of stochasticity

Definition

A distribution μ is called t_1, α-*simple* if it can be generated by a probabilistic program of length less than α in time at most t_1.

Definition

A t_2, β, ε-*acceptable* hypothesis for a string x is a distribution μ such that $\mu(T) > \varepsilon$ for all $T \ni x$ recognized by a program of length less than β in at most t_2 steps for all inputs of length $|x|$.

A distribution μ is a "good" explanation for a string x of length n if μ is t_1, α-simple and t_2, β, ε-acceptable for x where

The parameters of stochasticity

Definition

A distribution μ is called t_1, α-*simple* if it can be generated by a probabilistic program of length less than α in time at most t_1.

Definition

A t_2, β, ε-*acceptable* hypothesis for a string x is a distribution μ such that $\mu(T) > \varepsilon$ for all $T \ni x$ recognized by a program of length less than β in at most t_2 steps for all inputs of length $|x|$.

A distribution μ is a "good" explanation for a string x of length n if μ is t_1, α-simple and t_2, β, ε-acceptable for x where

- $\alpha = O(\log n)$, $t_1 = \text{poly}(n)$;

The parameters of stochasticity

Definition

A distribution μ is called t_1, α-*simple* if it can be generated by a probabilistic program of length less than α in time at most t_1.

Definition

A t_2, β, ε-*acceptable* hypothesis for a string x is a distribution μ such that $\mu(T) > \varepsilon$ for all $T \ni x$ recognized by a program of length less than β in at most t_2 steps for all inputs of length $|x|$.

A distribution μ is a "good" explanation for a string x of length n if μ is t_1, α-simple and t_2, β, ε-acceptable for x where

- $\alpha = O(\log n)$, $t_1 = \text{poly}(n)$;
- $\beta > \alpha$, $t_2 > t_1$;

The parameters of stochasticity

Definition

A distribution μ is called t_1, α-*simple* if it can be generated by a probabilistic program of length less than α in time at most t_1.

Definition

A t_2, β, ε-*acceptable* hypothesis for a string x is a distribution μ such that $\mu(T) > \varepsilon$ for all $T \ni x$ recognized by a program of length less than β in at most t_2 steps for all inputs of length $|x|$.

A distribution μ is a "good" explanation for a string x of length n if μ is t_1, α-simple and t_2, β, ε-acceptable for x where

- $\alpha = O(\log n)$, $t_1 = \text{poly}(n)$;
- $\beta > \alpha$, $t_2 > t_1$;
- $\varepsilon < \frac{1}{\text{poly}(n)}$.

Majority principle

Majority principle

- Let μ be a probability distribution on binary strings.

Majority principle

- Let μ be a probability distribution on binary strings.
- Let x be a string that was chosen randomly with respect to μ.

Majority principle

- Let μ be a probability distribution on binary strings.
- Let x be a string that was chosen randomly with respect to μ.
- Then with high probability x should be acceptable for x.

Majority principle

- Let μ be a probability distribution on binary strings.
- Let x be a string that was chosen randomly with respect to μ.
- Then with high probability x should be acceptable for x.

Proposition

For every probability distribution μ over binary strings of length n and for all β, ε and t we have
$\mu\{x \mid \mu \text{ is not } t, \beta, \varepsilon\text{-acceptable for } x\} < \varepsilon 2^{\beta}$.

The main result

The main result

- NE is the class of languages accepted in time $2^{O(n)}$ by non-deterministic Turing machines.

The main result

- NE is the class of languages accepted in time $2^{O(n)}$ by non-deterministic Turing machines.
- RE is the class of languages recognized in time $2^{O(n)}$ by probabilistic Turing machines that err with probability at most $\frac{1}{2}$ for all strings in the language and do not err for strings outside the language.

The main result

- NE is the class of languages accepted in time $2^{O(n)}$ by non-deterministic Turing machines.
- RE is the class of languages recognized in time $2^{O(n)}$ by probabilistic Turing machines that err with probability at most $\frac{1}{2}$ for all strings in the language and do not err for strings outside the language.

Theorem

If $\mathrm{RE} \neq \mathrm{NE}$ then for some constant d for all c for infinitely many n there is a string of length n that has no $n^c, c\log n$-simple, n^d, d, n^c-acceptable hypotheses.

The main result

- NE is the class of languages accepted in time $2^{O(n)}$ by non-deterministic Turing machines.
- RE is the class of languages recognized in time $2^{O(n)}$ by probabilistic Turing machines that err with probability at most $\frac{1}{2}$ for all strings in the language and do not err for strings outside the language.

Theorem

If $\mathrm{RE} \neq \mathrm{NE}$ then for some constant d for all c for infinitely many n there is a string of length n that has no $n^c, c \log n$-simple, n^d, d, n^c-acceptable hypotheses.

Existence of non-stochastic strings for such parameters implies that $\mathrm{P} \neq \mathrm{PSPACE}$.

An application of non-stochasticity

An application of non-stochasticity

Non-stochastic objects that were constructed under assumption $\mathrm{RE} \neq \mathrm{NE}$ have some interesting properties.

An application of non-stochasticity

Non-stochastic objects that were constructed under assumption $\mathrm{RE} \neq \mathrm{NE}$ have some interesting properties.
They can be used in proof of some statements of time-bounded Kolmogorov complexity.

An application of non-stochasticity

Non-stochastic objects that were constructed under assumption $\mathrm{RE} \neq \mathrm{NE}$ have some interesting properties.
They can be used in proof of some statements of time-bounded Kolmogorov complexity.

Denote by $\mathrm{C}^t(x)$ the minimum length of a program that produce x in time at most t.

An application of non-stochasticity

Non-stochastic objects that were constructed under assumption $\mathrm{RE} \neq \mathrm{NE}$ have some interesting properties.
They can be used in proof of some statements of time-bounded Kolmogorov complexity.

Denote by $\mathrm{C}^t(x)$ the minimum length of a program that produce x in time at most t.
Denote by $\mathrm{CD}^t(x)$ the minimum length of a program that distinguish x from other strings in time at most t.

An application of non-stochasticity

Non-stochastic objects that were constructed under assumption $\mathrm{RE} \neq \mathrm{NE}$ have some interesting properties.
They can be used in proof of some statements of time-bounded Kolmogorov complexity.

Denote by $\mathrm{C}^t(x)$ the minimum length of a program that produce x in time at most t.
Denote by $\mathrm{CD}^t(x)$ the minimum length of a program that distinguish x from other strings in time at most t.

Open problem: what are the relationships between $\mathrm{C}^{\mathrm{poly}(|x|)}(x)$ and $\mathrm{CD}^{\mathrm{poly}(|x|)}(x)$?

Not-stochastic objects give a particular answer to this question.

$\mathrm{C}^{\mathsf{poly}}(x|y)$ vs $\mathrm{CD}^{\mathsf{poly}(x|y)}$

$\mathrm{C}^{\mathsf{poly}}(x|y)$ vs $\mathrm{CD}^{\mathsf{poly}(x|y)}$

Proposition

$\forall t\ \exists c\ \forall x, y\ \mathrm{CD}^{ct \log t}(x|y) < \mathrm{C}^{t}(x|y) + c.$

$\mathrm{C}^{\text{poly}}(x|y)$ vs $\mathrm{CD}^{\text{poly}(x|y)}$

Proposition

$\forall t\ \exists c\ \forall x, y\ \mathrm{CD}^{ct\log t}(x|y) < \mathrm{C}^t(x|y) + c.$

Theorem (Lance Fortnow, Martin Kummer)

The statement "For every polynomial t there is a polynomial T and a constant c such that for all x and y: $\mathrm{CD}^T(x|y) < \mathrm{C}^t(x|y) + c$" is equivalent to $(1SAT, SAT) \in P$.

The last inclusion means that there is a deterministic polynomial time algorithm which accepts all Boolean formulas with a unique satisfying assignment, and rejects all Boolean formulas which are not satisfiable.

$\mathrm{C}^{\text{poly}}(x|y)$ vs $\mathrm{CD}^{\text{poly}(x|y)}$

Proposition

$\forall t\ \exists c\ \forall x, y\ \mathrm{CD}^{ct \log t}(x|y) < \mathrm{C}^t(x|y) + c.$

Theorem (Lance Fortnow, Martin Kummer)

The statement "For every polynomial t there is a polynomial T and a constant c such that for all x and y: $\mathrm{CD}^T(x|y) < \mathrm{C}^t(x|y) + c$" is equivalent to $(1SAT, SAT) \in P$.

The last inclusion means that there is a deterministic polynomial time algorithm which accepts all Boolean formulas with a unique satisfying assignment, and rejects all Boolean formulas which are not satisfiable.

What can we say about $\mathrm{C}^{\text{poly}(|x|)}(x)$ vs $\mathrm{CD}^{\text{poly}(|x|)}(x)$?

$\mathrm{C}^{\mathsf{poly}(|x|)}(x)$ vs $\mathrm{CD}^{\mathsf{poly}(|x|)}(x)$

Theorem (Lance Fortnow, Martin Kummer)

If $\mathrm{FewP} \cap \mathrm{SPARSE} \not\subseteq \mathrm{P}$ *then for some constant* d *for all* c *there exist infinitely many strings* x *such that*
$\mathrm{CD}^{n^d}(x) < \mathrm{C}^{n^c}(x) - c \log n$.
Here and further n *denotes the length of* x.

$\mathrm{C}^{\mathrm{poly}(|x|)}(x)$ vs $\mathrm{CD}^{\mathrm{poly}(|x|)}(x)$

Theorem (Lance Fortnow, Martin Kummer)

*If $\mathrm{FewP} \cap \mathrm{SPARSE} \not\subseteq \mathrm{P}$ then for some constant d for all c there exist infinitely many strings x such that $\mathrm{CD}^{n^d}(x) < \mathrm{C}^{n^c}(x) - c \log n$.
Here and further n denotes the length of x.*

Theorem

If $\mathrm{RE} \neq \mathrm{NE}$ then for some constant d for all c there exist infinitely many strings x such that $\mathrm{CD}^{n^d}(x|r) < \mathrm{C}^{n^c}(x|r) - c \log n$ for 99 % strings r of length n^d.

Theorem (Lance Fortnow, Martin Kummer)

If $\mathrm{FewP} \cap \mathrm{SPARSE} \not\subseteq \mathrm{P}$ *then for some constant* d *for all* c *there exist infinitely many strings* x *such that*
$\mathrm{CD}^{n^d}(x) < \mathrm{C}^{n^c}(x) - c \log n$.
Here and further n *denotes the length of* x.

Theorem

If $\mathrm{RE} \neq \mathrm{NE}$ *then for some constant* d *for all* c *there exist infinitely many strings* x *such that*
$\mathrm{CD}^{n^d}(x|r) < \mathrm{C}^{n^c}(x|r) - c \log n$ *for 99 % strings* r *of length* n^d.

Assume also that there is a set that is decidable by Turing machines in time $2^{O(n)}$ *but is not decidable by Boolean circuits of size* $2^{o(n)}$ *for almost all* n. *Then*
$\mathrm{CD}^{n^d}(x) < \mathrm{C}^{n^c}(x|r) - c \log n$ *for 99 % strings* r *of length* n^d.

Thank you!