What computational complexity theory tells us about (some) statistical problems

Michal Černý[1] & Milan Hladík[2]

[1] Department of Econometrics & DYME Research Center
University of Economics, Prague, Czech Republic

[2] Department of Applied Mathematics
Faculty of Mathematics and Physics, Charles University, Prague, Czech Republic

ISI 2015, Rio de Janeiro

Part I. Computational complexity: A pair of examples

The main role of computational complexity. Consider the following computational problem: given a function f and data x, compute $f(x)$.

Introduction: Computational complexity

The main role of computational complexity. Consider the following computational problem: given a function f and data x, compute $f(x)$.

- Can $f(x)$ be computed efficiently ($=$ in polynomial time)?

The main role of computational complexity. Consider the following computational problem: given a function f and data x, compute $f(x)$.

- Can $f(x)$ be computed efficiently (= in polynomial time)?
- Is computation of $f(x)$ NP-hard? (Informally: only **exponential-time algorithms** exist?)

The main role of computational complexity. Consider the following computational problem: given a function f and data x, compute $f(x)$.

- Can $f(x)$ be computed efficiently (= in polynomial time)?
- Is computation of $f(x)$ NP-hard? (Informally: only **exponential-time algorithms** exist?)
- Is $f(x)$ computable (recursive) at all?

The main role of computational complexity. Consider the following computational problem: given a function f and data x, compute $f(x)$.

- Can $f(x)$ be computed efficiently (= in polynomial time)?
- Is computation of $f(x)$ NP-hard? (Informally: only **exponential-time algorithms** exist?)
- Is $f(x)$ computable (recursive) at all?
- If computation of $f(x)$ is hard, can we compute at least an approximation of $f(x)$ efficiently? Or is $f(x)$ efficiently inapproximable?

The main role of computational complexity. Consider the following computational problem: given a function f and data x, compute $f(x)$.

- Can $f(x)$ be computed efficiently (= in polynomial time)?
- Is computation of $f(x)$ NP-hard? (Informally: only **exponential-time algorithms** exist?)
- Is $f(x)$ computable (recursive) at all?
- If computation of $f(x)$ is hard, can we compute at least an approximation of $f(x)$ efficiently? Or is $f(x)$ efficiently inapproximable?
- Can the computation of $f(x)$ be efficiently parallelized? Or is it intrinsically sequential?

The main role of computational complexity. Consider the following computational problem: given a function f and data x, compute $f(x)$.

- Can $f(x)$ be computed efficiently (= in polynomial time)?
- Is computation of $f(x)$ NP-hard? (Informally: only **exponential-time algorithms** exist?)
- Is $f(x)$ computable (recursive) at all?
- If computation of $f(x)$ is hard, can we compute at least an approximation of $f(x)$ efficiently? Or is $f(x)$ efficiently inapproximable?
- Can the computation of $f(x)$ be efficiently parallelized? Or is it intrinsically sequential?
- Many further and finer questions (weak/strong polynomiality, pseudopolynomiality, randomized computing, reductions among problems, polynomial-time hierarchy, space (memory) complexity . . .)

Examples

Complexity theory goes across many areas of mathematics:

- **Number theory.** Given $a, b, c \in \mathbb{N}$, finding a solution $x, y \in \mathbb{N}$ of the equation $ax^2 + by = c$ has a solution is NP-hard.

Examples

Complexity theory goes across many areas of mathematics:

- **Number theory.** Given $a, b, c \in \mathbb{N}$, finding a solution $x, y \in \mathbb{N}$ of the equation $ax^2 + by = c$ has a solution is NP-hard.
- **Graph theory.** Traveling salesman is NP-hard.

Examples

Complexity theory goes across many areas of mathematics:

- **Number theory.** Given $a, b, c \in \mathbb{N}$, finding a solution $x, y \in \mathbb{N}$ of the equation $ax^2 + by = c$ has a solution is NP-hard.
- **Graph theory.** Traveling salesman is NP-hard.
- **Calculus.** Given a function $f : \mathbb{R} \to \mathbb{R}$ composed of $+, -, \times, \div, \sin$, determining whether it has a root is undecidable.

Examples

Complexity theory goes across many areas of mathematics:

- **Number theory.** Given $a, b, c \in \mathbb{N}$, finding a solution $x, y \in \mathbb{N}$ of the equation $ax^2 + by = c$ has a solution is NP-hard.
- **Graph theory.** Traveling salesman is NP-hard.
- **Calculus.** Given a function $f : \mathbb{R} \to \mathbb{R}$ composed of $+, -, \times, \div, \sin$, determining whether it has a root is undecidable.
- **Optimization.** Continuous linear programming is “easy”, while integer linear programming is “hard”.

Examples

Complexity theory goes across many areas of mathematics:

- **Number theory.** Given $a, b, c \in \mathbb{N}$, finding a solution $x, y \in \mathbb{N}$ of the equation $ax^2 + by = c$ has a solution is NP-hard.
- **Graph theory.** Traveling salesman is NP-hard.
- **Calculus.** Given a function $f : \mathbb{R} \to \mathbb{R}$ composed of $+, -, \times, \div, \sin$, determining whether it has a root is undecidable.
- **Optimization.** Continuous linear programming is "easy", while integer linear programming is "hard".
- **Set theory.** It is undecidable whether a statement is provable in Zermelo-Fraenkel set theory.

Examples

Complexity theory goes across many areas of mathematics:

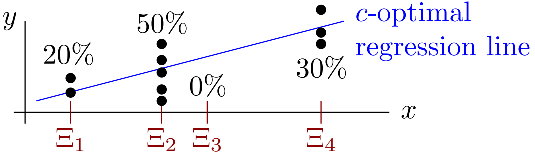
- **Number theory.** Given $a, b, c \in \mathbb{N}$, finding a solution $x, y \in \mathbb{N}$ of the equation $ax^2 + by = c$ has a solution is NP-hard.
- **Graph theory.** Traveling salesman is NP-hard.
- **Calculus.** Given a function $f : \mathbb{R} \to \mathbb{R}$ composed of $+, -, \times, \div, \sin$, determining whether it has a root is undecidable.
- **Optimization.** Continuous linear programming is "easy", while integer linear programming is "hard".
- **Set theory.** It is undecidable whether a statement is provable in Zermelo-Fraenkel set theory.
- **And statistics ??**

An example of our previous work: c-optimal design

Formulation. Consider the linear regression model $y = X\beta + \varepsilon$, where ε_i are $N(0, \sigma^2)$ independent. We are given a finite experimental domain Ξ and $c \neq 0$. Two natural problems:

Formulation. Consider the linear regression model $y = X\beta + \varepsilon$, where ε_i are $N(0, \sigma^2)$ independent. We are given a finite experimental domain Ξ and $c \neq 0$. Two natural problems:

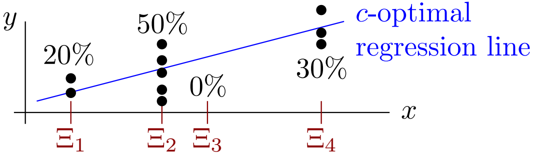
- **ApproxDesign**: Find the approximately c-optimal design over Ξ.
  - Find the probabilistic measure ξ on Ξ minimizing $\mathrm{var}(c^T\widehat{\beta})$, where $\widehat{\beta}$ is the OLS-estimator.
  - Example: if $\xi = (0.2, 0.5, 0, 0.3)^{\mathrm{T}}$, then it is optimal to make 20% observations in Ξ_1, then 50% observations in Ξ_2 etc.



An example of our previous work: c-optimal design

Formulation. Consider the linear regression model $y = X\beta + \varepsilon$, where ε_i are $N(0, \sigma^2)$ independent. We are given a finite experimental domain Ξ and $c \neq 0$. Two natural problems:

- **ApproxDesign**: Find the approximately c-optimal design over Ξ.
  - Find the probabilistic measure ξ on Ξ minimizing $\mathrm{var}(c^T \widehat{\beta})$, where $\widehat{\beta}$ is the OLS-estimator.
  - Example: if $\xi = (0.2, 0.5, 0, 0.3)^{\mathrm{T}}$, then it is optimal to make 20% observations in Ξ_1, then 50% observations in Ξ_2 etc.

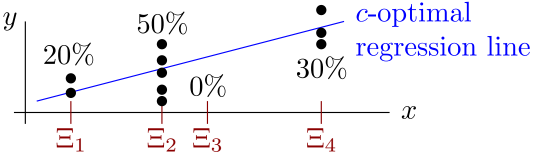


- **ExactDesign**: Given n, find the n-exact c-optimal design over Ξ.
  - Given n (the number of observations), find the probabilistic measure ξ on Ξ minimizing $\mathrm{var}(c^T \widehat{\beta})$ s.t. $n\xi$ is an integer vector.

An example of our previous work: c-optimal design

Formulation. Consider the linear regression model $y = X\beta + \varepsilon$, where ε_i are $N(0, \sigma^2)$ independent. We are given a finite experimental domain Ξ and $c \neq 0$. Two natural problems:

- **ApproxDesign**: Find the approximately c-optimal design over Ξ.
  - Find the probabilistic measure ξ on Ξ minimizing $\text{var}(c^T\widehat{\beta})$, where $\widehat{\beta}$ is the OLS-estimator.
  - Example: if $\xi = (0.2, 0.5, 0, 0.3)^{\mathrm{T}}$, then it is optimal to make 20% observations in Ξ_1, then 50% observations in Ξ_2 etc.



- **ExactDesign**: Given n, find the n-exact c-optimal design over Ξ.
  - Given n (the number of observations), find the probabilistic measure ξ on Ξ minimizing $\text{var}(c^T\widehat{\beta})$ s.t. $n\xi$ is an integer vector.
- Are these problems computational easy or hard?

c-optimal design (contd.)

Some results (Černý & Hladík, *Comput Optim Appl*, 2012):

- **ExactDesign** is NP-hard.
- **ApproxDesign** is computable in polynomial time, but it is P-complete — the "hardest" among all efficiently computable problems.
- **ApproxDesign** cannot be efficiently parallelized.
- **ApproxDesign** is equivalent to general linear programming.
  - The message: *Do not try to design "good" algorithms for the problem. If you try to, then you will be competing against interior point algorithms and it is not easy to defeat them. But if you succeed, you'll be truly famous in optimization.*

Part II. Computational complexity and analysis of one-dimensional interval data

One-dimensional interval data: a model

Assumptions.

- Let $x_1, \ldots, x_n$ be a dataset; for example, let the data be a random sample from a distribution Φ. The dataset is unobservable.

One-dimensional interval data: a model

Assumptions.

- Let $x_1, \ldots, x_n$ be a dataset; for example, let the data be a random sample from a distribution Φ. The dataset is unobservable.
- What is observable is a collection of intervals $\mathbf{x}_1 = [\underline{x}_1, \overline{x}_1], \ldots,$ $\mathbf{x}_n = [\underline{x}_n, \overline{x}_n]$ such that

$$x_1 \in \mathbf{x}_1, \ldots, x_n \in \mathbf{x}_n \quad \text{a.s.}$$

One-dimensional interval data: a model

Assumptions.

- Let $x_1, \ldots, x_n$ be a dataset; for example, let the data be a random sample from a distribution Φ. The dataset is unobservable.
- What is observable is a collection of intervals $\mathbf{x}_1 = [\underline{x}_1, \overline{x}_1], \ldots,$ $\mathbf{x}_n = [\underline{x}_n, \overline{x}_n]$ such that

$$x_1 \in \mathbf{x}_1, \ldots, x_n \in \mathbf{x}_n \quad \text{a.s.}$$

- **A general goal:** We want to make inference about the original dataset $x = (x_1, \ldots, x_n)$, about the generating distribution Φ, about its parameters, we want to test hypotheses etc.

One-dimensional interval data: a model

Assumptions.

- Let $x_1, \ldots, x_n$ be a dataset; for example, let the data be a random sample from a distribution Φ. The dataset is unobservable.
- What is observable is a collection of intervals $\mathbf{x}_1 = [\underline{x}_1, \overline{x}_1], \ldots,$ $\mathbf{x}_n = [\underline{x}_n, \overline{x}_n]$ such that

$$x_1 \in \mathbf{x}_1, \ldots, x_n \in \mathbf{x}_n \quad \text{a.s.}$$

- **A general goal:** We want to make inference about the original dataset $x = (x_1, \ldots, x_n)$, about the generating distribution Φ, about its parameters, we want to test hypotheses etc.
- We are given a statistic $S(x_1, \ldots, x_n)$ and we want to determine/estimate its value, distribution, or other properties, using only the observable intervals $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$.

One-dimensional interval data: a model

Assumptions.

- Let $x_1, \ldots, x_n$ be a dataset; for example, let the data be a random sample from a distribution Φ. The dataset is unobservable.
- What is observable is a collection of intervals $\mathbf{x}_1 = [\underline{x}_1, \overline{x}_1], \ldots,$ $\mathbf{x}_n = [\underline{x}_n, \overline{x}_n]$ such that

$$x_1 \in \mathbf{x}_1, \ldots, x_n \in \mathbf{x}_n \quad \text{a.s.}$$

- **A general goal:** We want to make inference about the original dataset $x = (x_1, \ldots, x_n)$, about the generating distribution Φ, about its parameters, we want to test hypotheses etc.
- We are given a statistic $S(x_1, \ldots, x_n)$ and we want to determine/estimate its value, distribution, or other properties, using only the observable intervals $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$.
- **Now:** the appropriate toolbox depends on whether we can make further assumptions on the distribution of $(x, \mathbf{x})$.
  - *For example, in which practical situations can we assume that x is uniformly distributed on $\mathbf{x}$ and when we cannot?*

The possibilistic approach

- In this lecture: our only knowledge about $(x, \mathbf{x})$ is $x \in \mathbf{x}$ a.s. **Nothing more.**

The possibilistic approach

- In this lecture: our only knowledge about $(x, \mathbf{x})$ is $x \in \mathbf{x}$ a.s. **Nothing more.**
- Then, given a statistic S, the only information we can infer about S from the observable intervals $\mathbf{x}$ is the pair of tight bounds

$$\overline{S} = \max\{S(\xi) : \xi \in \mathbf{x}\},$$
$$\underline{S} = \min\{S(\xi) : \xi \in \mathbf{x}\},$$

clearly satisfying

$$\underline{S} \leqslant S(x) \leqslant \overline{S} \ \text{ a.s.}$$

- **Remark.** In econometrics, partial knowledge about the distribution $(x, \mathbf{x})$ is referred to as partial identification: see the survey paper E. Tamer, **Partial identification in econometrics**, Annual Review of Economics 2 (2010), pp. 167–195.
- Also many papers in Econometrica and other journals.

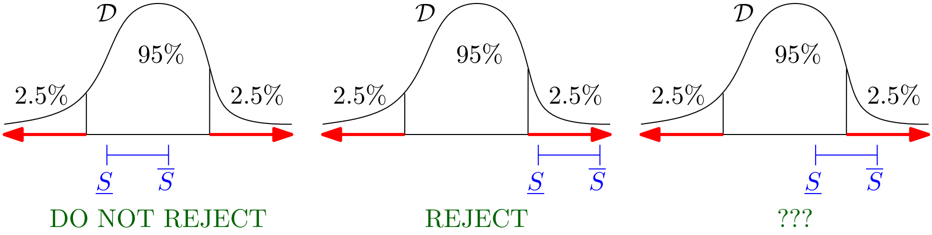
The core problem is:

- Given a statistic $S(x_1, \ldots, x_n)$ and the intervals $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$, is it computationally easy or difficult to determine

$$\overline{S} = \max\{S(\xi) : \xi \in \mathbf{x}\} \quad \text{and} \quad \underline{S} = \min\{S(\xi) : \xi \in \mathbf{x}\} \ ?$$

Example: Test statistics

- We are to test a null hypothesis (H_0) against an alternative A using a test statistic S
- Let $\mathcal{D}$ be the distribution of S under H_0
- Given the intervals $\mathbf{x}_1, \ldots, \mathbf{x}_n$: if we can compute $\underline{S}, \overline{S}$, then we can make at least partial conclusions:



Sample variance & complexity

$$\overline{s^2} = \max \left\{ \frac{1}{n-1} \sum_{i=1}^{n} \left(x_i - \frac{1}{n} \sum_{j=1}^{n} x_j \right)^2 \ : \ x \in \mathbf{x} \right\},$$

$$\underline{s^2} = \min \left\{ \frac{1}{n-1} \sum_{i=1}^{n} \left(x_i - \frac{1}{n} \sum_{j=1}^{n} x_j \right)^2 \ : \ x \in \mathbf{x} \right\}.$$

Sample variance & complexity

$$\overline{s^2} = \max\left\{ \frac{1}{n-1}\sum_{i=1}^{n}\left(x_i - \frac{1}{n}\sum_{j=1}^{n} x_j\right)^2 \ : \ x \in \mathbf{x} \right\},$$

$$\underline{s^2} = \min\left\{ \frac{1}{n-1}\sum_{i=1}^{n}\left(x_i - \frac{1}{n}\sum_{j=1}^{n} x_j\right)^2 \ : \ x \in \mathbf{x} \right\}.$$

- **Observation:** $\underline{s^2} \to$ CQP $\to$ weakly polynomial time
- **Ferson et al.:** a strongly polynomial algorithm $O(n^2)$
- **Unfortunately:** $\overline{s^2}$ is NP-hard
- **Even worse:** $\overline{s^2}$ is NP-hard to approximate with an **arbitrary** absolute error
  - *The message: if somebody claims that (s)he can design an efficient algorithm for computing $\overline{s^2}$ with an error at most ± 1000, then (s)he has proved $P = NP$ and will get the \$1M award from the Clay Math Institute...*

Sample variance & complexity

- **Well...** NP-hardness of computation of $\overline{s^2}$ means: When we have $n = 100$ data points, then we need computation time $2^{100} = \infty$. This is truly bad news.

- **Well. . .** NP-hardness of computation of $\overline{s^2}$ means: When we have $n = 100$ data points, then we need computation time $2^{100} = \infty$. This is truly bad news.
- Even approximate computation of $\overline{s^2}$ is impossible.

Sample variance & complexity

- **Well...** NP-hardness of computation of $\overline{s^2}$ means: When we have $n = 100$ data points, then we need computation time $2^{100} = \infty$. This is truly bad news.
- Even approximate computation of $\overline{s^2}$ is impossible.
- So what can we do now?

Sample variance & complexity

- **Well. . .** NP-hardness of computation of $\overline{s^2}$ means: When we have $n = 100$ data points, then we need computation time $2^{100} = \infty$. This is truly bad news.
- Even approximate computation of $\overline{s^2}$ is impossible.
- So what can we do now?
- **Option 1.** Give up and go home.

Sample variance & complexity

- **Well...** NP-hardness of computation of $\overline{s^2}$ means: When we have $n = 100$ data points, then we need computation time $2^{100} = \infty$. This is truly bad news.
- Even approximate computation of $\overline{s^2}$ is impossible.
- So what can we do now?
- **Option 1.** Give up and go home.
- **Option 2.** Investigate the class of instances making the problem computationally hard ("complexity core") and ask a question whether they occur often or rarely.

Sample variance & complexity

- **Well...** NP-hardness of computation of $\overline{s^2}$ means: When we have $n = 100$ data points, then we need computation time $2^{100} = \infty$. This is truly bad news.
- Even approximate computation of $\overline{s^2}$ is impossible.
- So what can we do now?
- **Option 1.** Give up and go home.
- **Option 2.** Investigate the class of instances making the problem computationally hard ("complexity core") and ask a question whether they occur often or rarely.
- **Following Option 2, we'll try to show that the situation isn't so catastrophic.**

An algorithm by Ferson et al.

- **Notation:** for an interval $\mathbf{x} = [\underline{x}, \overline{x}]$, define

$$x^C := \tfrac{1}{2}(\overline{x} + \underline{x}) \text{ (center)}, \quad x^\Delta := \tfrac{1}{2}(\overline{x} - \underline{x}) \text{ (radius)}$$

- **Notation:** for an interval $\mathbf{x} = [\underline{x}, \overline{x}]$, define

$$x^C := \tfrac{1}{2}(\overline{x} + \underline{x}) \text{ (center)}, \quad x^\Delta := \tfrac{1}{2}(\overline{x} - \underline{x}) \text{ (radius)}$$

- **Ferson et al.**: consider the "$\frac{1}{n}$-narrowed" intervals

$$\tfrac{1}{n}\mathbf{x}_i := [\mathbf{x}_i^C - \tfrac{1}{n}\mathbf{x}_i^\Delta, \mathbf{x}_i^C + \tfrac{1}{n}\mathbf{x}_i^\Delta], \quad i = 1, \ldots, n.$$

Theorem: If $\frac{1}{n}\mathbf{x}_i \cap \frac{1}{n}\mathbf{x}_j = \emptyset$ for all $i \neq j$, then $\overline{s^2}$ can be computed in polynomial time.

- **Notation:** for an interval $\mathbf{x} = [\underline{x}, \overline{x}]$, define

$$x^C := \tfrac{1}{2}(\overline{x} + \underline{x}) \text{ (center)}, \quad x^\Delta := \tfrac{1}{2}(\overline{x} - \underline{x}) \text{ (radius)}$$

- **Ferson et al.**: consider the "$\frac{1}{n}$-narrowed" intervals

$$\tfrac{1}{n}\mathbf{x}_i := [\mathbf{x}_i^C - \tfrac{1}{n}\mathbf{x}_i^\Delta, \mathbf{x}_i^C + \tfrac{1}{n}\mathbf{x}_i^\Delta], \quad i = 1, \dots, n.$$

  Theorem: If $\frac{1}{n}\mathbf{x}_i \cap \frac{1}{n}\mathbf{x}_j = \emptyset$ for all $i \neq j$, then $\overline{s^2}$ can be computed in polynomial time.

- Another formulation: If there is no k-tuple of indices $1 \leqslant i_1 < \cdots < i_k \leqslant n$ such that

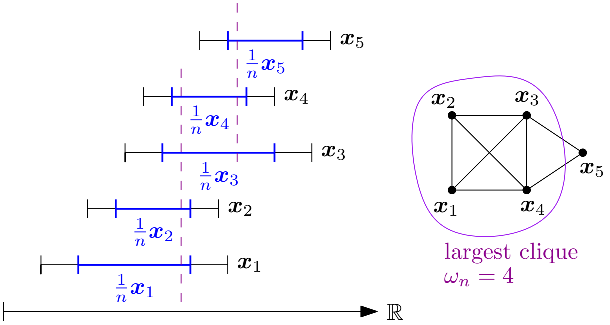
$$\bigcap_{\ell \in \{i_1, \dots, i_k\}} \tfrac{1}{n}\mathbf{x}_\ell \neq \emptyset,$$

  then $\overline{s^2}$ can be computed in time $O(p(n) \cdot 2^k)$, where p is a polynomial.

Computation of $\overline{s^2}$ & Ferson et al. (contd.)

Graph-theoretic reformulation: Let $G_n(V_n, E_n)$ be the interval graph over $\frac{1}{n}\mathbf{x}_1, \ldots, \frac{1}{n}\mathbf{x}_n$:

- Vertices: V_n = set of the narrowed intervals $\frac{1}{n}\mathbf{x}_1, \ldots, \frac{1}{n}\mathbf{x}_n$
- Edges: $\{i, j\} \in E$ $(i \neq j)$ iff $\frac{1}{n}\mathbf{x}_i \cap \frac{1}{n}\mathbf{x}_j \neq \emptyset$
- Let ω_n be the size of the largest clique of G_n. Now: the algorithm works in time $O(p(n) \cdot 2^{\omega_n})$.



Remark. The worst case is bad — e.g. when $\mathbf{x}_1^C = \mathbf{x}_2^C = \cdots = \mathbf{x}_n^C$. (Such instances result from the NP-hardness proof.)

But: What if the data are generated by a random process? Then, do the "ugly" instances occur frequently, or only rarely?

Simulations

Assumption: The centers and radii of intervals $\mathbf{x}_i$ are generated by a "reasonable" random process:

- Centers $\mathbf{x}_i^C$: sampled from a "reasonable" distribution (continuous, finite variance) — uniform, normal, exp, ...
- Radii $\mathbf{x}_i^\Delta$: sampled from a "reasonable" nonnegative distribution (continuous, finite variance) — uniform, one-sided normal, exp, ...

Simulations

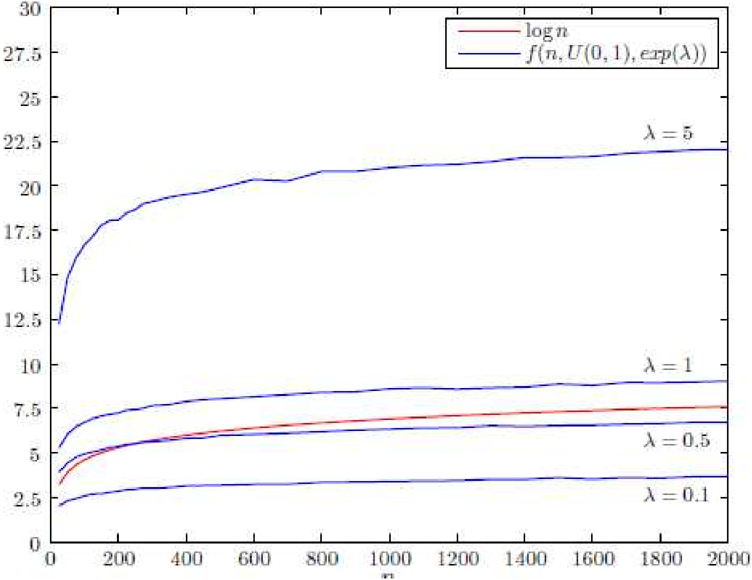
Assumption: The centers and radii of intervals $\mathbf{x}_i$ are generated by a "reasonable" random process:

- Centers $\mathbf{x}_i^C$: sampled from a "reasonable" distribution (continuous, finite variance) — uniform, normal, exp, ...
- Radii $\mathbf{x}_i^\Delta$: sampled from a "reasonable" nonnegative distribution (continuous, finite variance) — uniform, one-sided normal, exp, ...
- Simulations show Sokol's conjecture: The clique is **logarithmic** on average!
- If indeed $\mathrm{E}\omega_n = O(\log n)$, then the average computation time is

$$O(p(n) \cdot 2^{\omega_n}) = O(p(n) \cdot 2^{O(\log n)}) = \text{polynomial}(n).$$

- Thus: The algorithm is polynomial on average (even if exponential in the worst case).

Sokol's conjecture



Furthermore: It seems that $\mathrm{var}(\omega_n) = O(1)$ ("Sokol's conjecture II").

- Say, for simplicity, that indeed $\mathrm{E}\omega_n = \log n$. By Chebyshev's inequality we get:

$$\Pr[\omega_n \geqslant \log n + \underbrace{10\sqrt{\mathrm{var}(\omega_n)}}_{=:K \text{ (constant)}}] \leqslant 1\%.$$

Sokol's conjecture II

Furthermore: It seems that $\text{var}(\omega_n) = O(1)$ ("Sokol's conjecture II").

- Say, for simplicity, that indeed $\mathsf{E}\omega_n = \log n$. By Chebyshev's inequality we get:

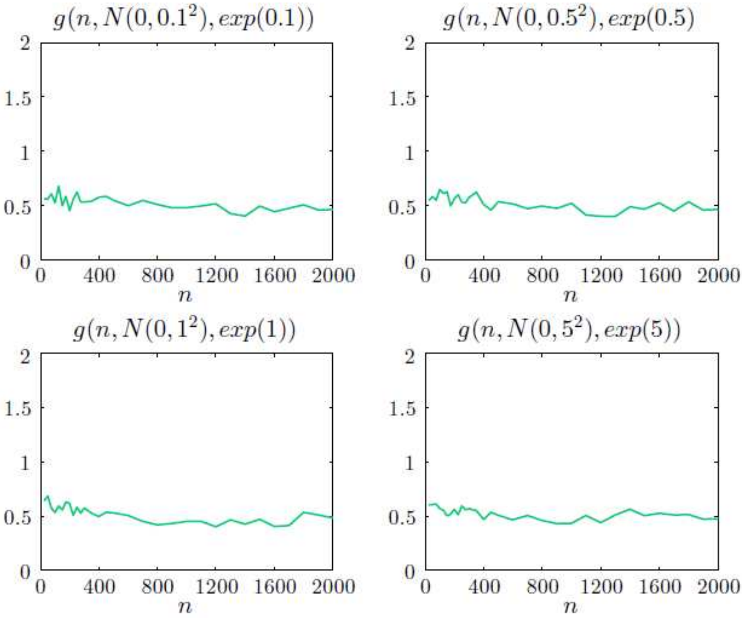
$$\Pr[\omega_n \geqslant \log n + \underbrace{10\sqrt{\text{var}(\omega_n)}}_{=:K \text{ (constant)}}] \leqslant 1\%.$$

- Thus: in 99% cases, the algorithm of Ferson et al. works in time at most

$$p(n) \cdot 2^{K + \log n},$$

where K does not grow with n.

Sokol's conjecture II



A pair of remarks

To prove the conjectures:

- We have random intersection (interval) graphs and we need to estimate the average size of the largest clique and its variance
- Interesting problem for graph theory: our model of a random graph is different from the traditional models $G_{n,p}$ and $G_{n,m}$

A pair of remarks

To prove the conjectures:

- We have random intersection (interval) graphs and we need to estimate the average size of the largest clique and its variance
- Interesting problem for graph theory: our model of a random graph is different from the traditional models $G_{n,p}$ and $G_{n,m}$

Further good news:

- $\overline{s^2}$ is computable pseudopolynomially
- Main message: although NP-hard in theory, $\overline{s^2}$ is efficiently computable "almost always" (in the probabilistic setup) — hard instances are rare
- A nice interdisciplinary problem: statistical motivation, interval-theoretic and graph-theoretic methods
- Some ideas can be generalized for other statistics which are known to be NP-hard, e.g. the F-ratio

$$F = \frac{\text{sample variance of } x_1, \ldots, x_{n/2}}{\text{sample variance of } x_{(n/2)+1}, \ldots, x_n}.$$

Conclusions & further reading

- The coefficient of variation (t-ratio) has been studied in our paper Černý M. and Hladík M. Complexity of computation and approximation of the t-ratio over one-dimensional interval data. *Comput Stat Data Anal* 80, 2014, 26–43.
- Many further results can be found in the book H. Nguyen et al. *Computing statistics under interval and fuzzy uncertainty. Applications to computer science and engineering.* Vol. 393 of Studies in Computational Intelligence, Springer, Berlin, 2012.
- We have further results of this kind in linear regression, see e.g. our preprint Hladík M. and Černý M. Linear regression with interval data: Computational issues (available from `http://nb.vse.cz/~cernym/ilr.pdf`).

Thank you!