CS 58500 – Theoretical Computer Science Toolkit

Lecture 0 (01/13)

Introduction

https://ruizhezhang.com/course_spring_2026.html

About Me

Ruizhe Zhang

- Assistant Professor, Purdue CS
- Research: Theory--quantum computing, optimization, ML theory
- Homepage: https://ruizhezhang.com/
- Email: rzzhang@purdue.edu

Today's Lecture

- Course content
- Logistics
- Some motivating examples

Course Content

TCS toolkit

TCS = Theoretical Computer Science

≈ Algorithms + Computational Complexity

≈ STOC/FOCS topics

- algorithms and data structures
- computational complexity
- randomness in computing
- algorithmic graph theory and combinatorics
- analysis of Boolean functions
- approximation algorithms
- cryptography
- computational learning theory
- continuous and discrete optimization
- economics and computation
- parallel and distributed algorithms
- quantum computing
- algorithmic coding theory
- computational geometry and topology
- computational applications of logic
- algebraic computation
- computational and foundational aspects of areas such as machine learning, fairness, privacy, networks, data management, databases and computational biology.

What is TCS?

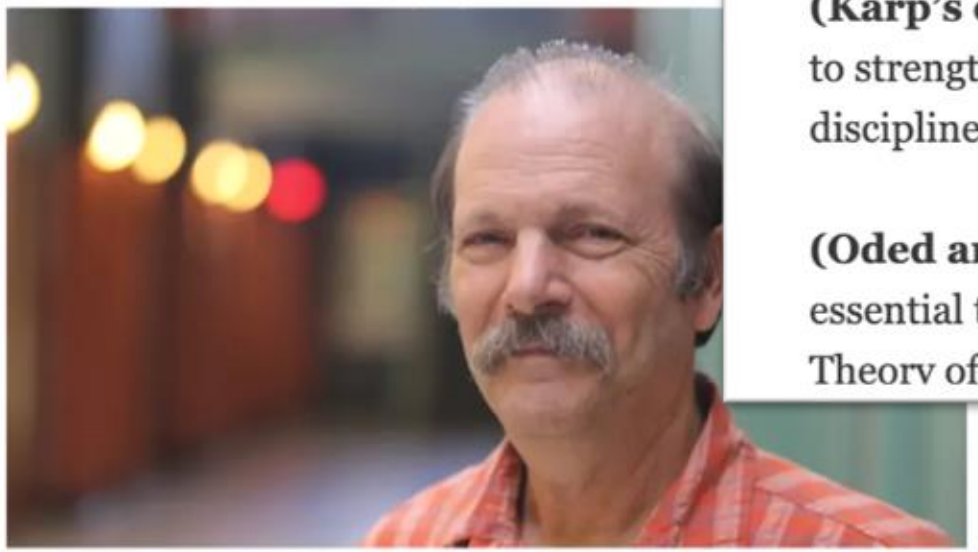
7. **The very old debate: Karp's committee vs. Wigderson & Goldreich.** There was a related very interesting debate almost thirty years ago that is discussed in [this blog post] over the blog Computational Complexity (CC) that present the two positions as follows:

(Karp's committee:) In order for TOC to prosper in the coming years, it is essential to strengthen our comm
disciplines, and to incr

(Oded and Avi's viev
essential that Theoreti
Theory of Computing a

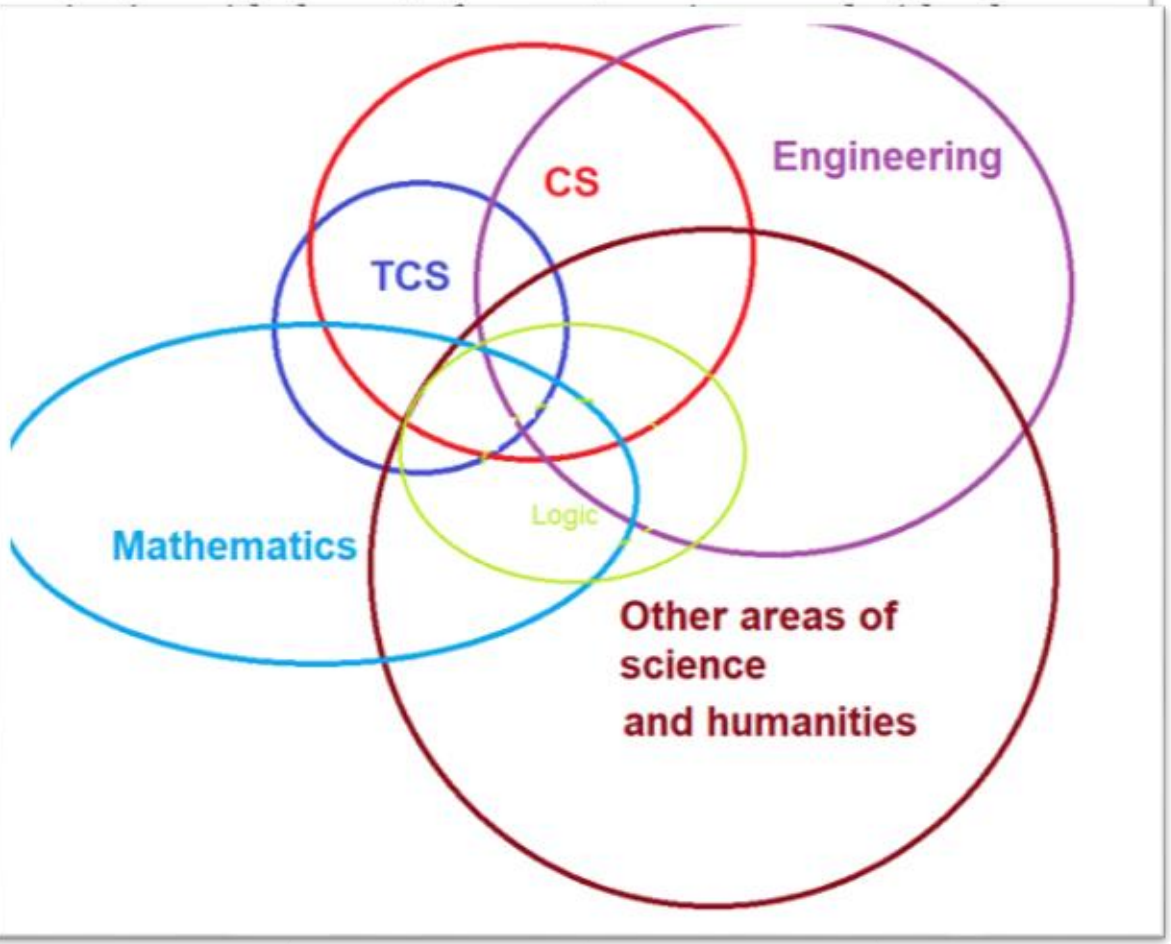
Moshe Vardi: What is Theoretical

Posted on October 20, 2024 by Gil Kalai



Moshe Vardi wrote a short interesting essay "What is theoretical computer scie
followed by interesting posts on Facebook.) Moshe argues that

> Thinking of theoretical computer science (TCS) as a branch of mathemat
> is harmful to the discipline.



Course Content

Introduce Foundational Topics in Theoretical Computer Science & Machine Learning

- Mathematical Foundations
- Wide Applications in Computer Science

Grand Aim

Covers fundamental techniques and a range of mathematical tools that underlie today's research in theoretical computer science

- Essential knowledge for students pursuing research in theoretical computer science or machine learning theory

  - Reading TCS papers
  - Attending TCS talks
  - Solving TCS problems

Target audience

Current graduate and undergrad students interested in pursuing research in these areas

- Undergraduates interested in taking the course should contact the instructor for permission

Prerequisites: Mastery of the material covered in

- Calc III (Math 261)
- Linear Algebra (Math 265)
- Probability (STAT 416)
- Foundations of CS (CS 182)
- Analysis of Algorithms (CS 381 or CS 580)

"Mathematical maturity" is important!

Tentative List of Topics

Mathematical Basics

- Inequalities: Jensen’s Inequality and consequences
- Summations/Integrals
- Stirling Approximation

Tentative List of Topics

Concentration Inequalities

- Markov Inequality, Chebyshev Inequality
- Chernoff-Hoeffding Bound
- Azuma's Inequality, McDiarmid's Inequality
- Concentration of measure
- Talagrand Inequality
- Matrix concentration inequalities

Tentative List of Topics

Selected Topics in Convex Analysis and Optimization

- Introduction to convex sets and functions
- Separating hyperplane theorem, Grünbaum's lemma
- Brunn-Minkowski inequality
- Isoperimetric inequalities and localization
- Strong convexity and oracle complexity of gradient descent
- Polynomial approximations

Tentative List of Topics

Foundations of Spectral Methods

- Positive semi-definiteness, Spectral and singular value decompositions
- Courant–Fischer–Weyl minimax theorems
- Perron–Frobenius theory
- Matrix norms and perturbation theory
- Exander graphs
- Random walks and Markov chains
- Ramanujan graphs, interlacing polynomials, and Free Probability Theory

Tentative List of Topics

Discrete Fourier Analysis on the Boolean Hypercube

- Basic properties of discrete Fourier transform
- BLR linearity testing
- Query complexity, degrees, and sensitivity
- Hypercontractivity
- KKL Theorem, and isoperimetric inequality over Boolean hypercube

Today's lecture

- Course content
- Logistics
- Some motivating examples

Course Instruction

- Course website: https://ruizhezhang.com/course_spring_2026.html
- Brightspace for announcements and problem set submissions
- Piazza for discussions
  - Access 1: Brightspace → Content → Piazza → Piazza 1.3
  - Access 2: https://piazza.com/class/mkc2aeu03z2m4

Grading

- 30% Problem sets (roughly three or four)
- 25% Midterm (date will be announced later, probably a late midterm)
- 40% Final project (oral presentation + report; Individual or pairs)
- 5% Class Participation

Office Hours

- Office hour with Instructor: By Appointment Only

- Office hour with TA: *Friday 10:30 – 11:30 am at DSAI B055*

Concluding Remarks

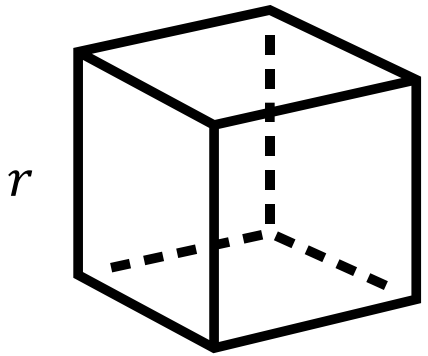
- This course will be challenging; We are here to work together
- We will collaboratively learn from each other
- Don’t hesitate to stop me at any point to ask questions.

Today's Lecture

- Course content
- Logistics
- Some motivating examples

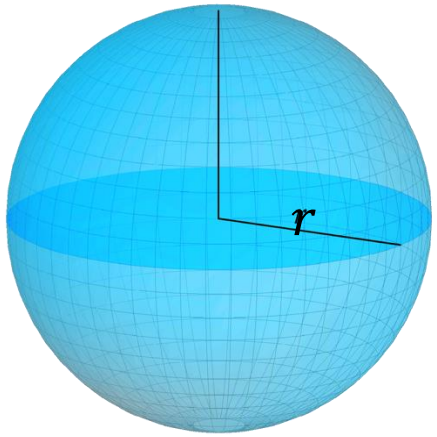
Example 1: Computing volumes in high dimensions

- Let $d \geq 2$ be the dimension of the space
- High-dimensional geometric problem: the size of input is a function of d, and we care about the complexity of the computational problem as d grows



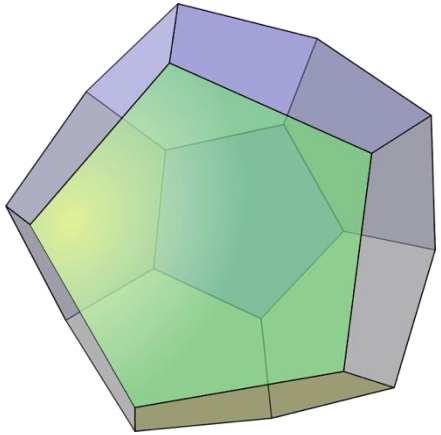
Cube (side-length r)?

$$\text{Vol} = r^d$$



Ball (radius r)?

$$\text{Vol} = \left(\frac{c}{d}\right)^{d/2} r^d$$



Polytope?
Need an algorithm!

Example 1: Computing volumes in high dimensions

Our goal is to approximately compute the volume of the given convex body K

- Exact computation is **#P-hard** (even for a polytope)
- Convexity is a natural and necessary assumption (will be formally defined later in this course)

How to get access to the convex body K?

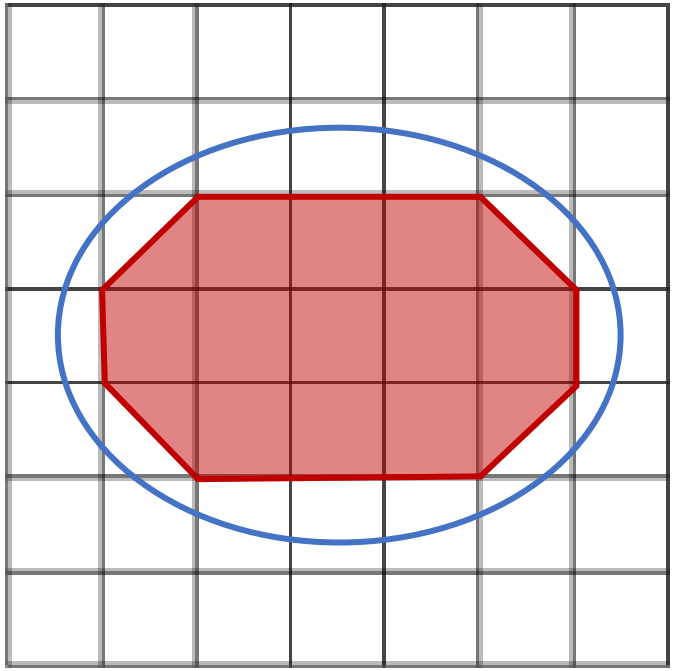
- We can query a membership oracle: "$x \in K$?" for any $x \in \mathbb{R}^d$

Let's start with the 2-d case

- *Pick a set of points* $\{x_1, \dots, x_n\}$ *and see if they are in it*
- *Then output the convex hull of* $\{x_i : x_i \in K, i \in [n]\}$

Does this give a poly time algorithm?

- Yes, if K is "well-rounded" (i.e., $B_2^d \subseteq K \subseteq dB_2^d$)
- Could even be a deterministic algorithm



Example 1: Computing volumes in high dimensions

Our goal is to approximately compute the volume of the given convex body K

- Exact computation is **#P-hard** (even for a polytope)
- Convexity is a natural and necessary assumption (will be formally defined later in this course)

How to get access to the convex body K?

- We can query a membership oracle: "$x \in K$?" for any $x \in \mathbb{R}^d$

Let's start with the 2-d case

- *Pick a set of points* $\{x_1, \dots, x_n\}$ *and see if they are in it*
- *Then output the convex hull of* $\{x_i : x_i \in K, i \in [n]\}$

Doesn't work in higher dimensions!

Does this give a poly time algorithm?

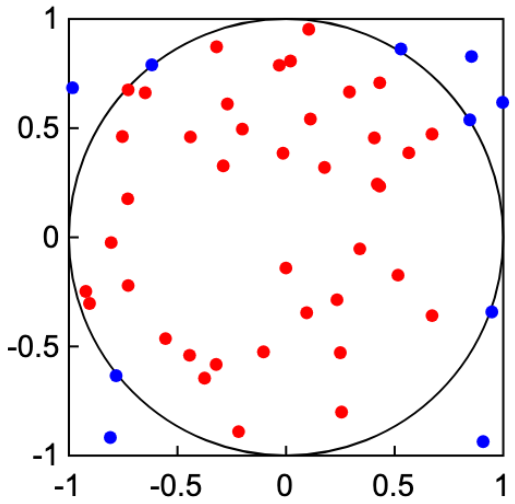
- Yes, if K is "well-rounded" (i.e., $B_2^d \subseteq K \subseteq dB_2^d$)
- Could even be a deterministic algorithm

Example 1: Computing volumes in high dimensions

Theorem. A deterministic $2^{\delta d}$-approximation algorithm requires $2^{(1-\delta)d}$ oracle queries.

Maybe a Monte Carlo algorithm?

- *Pick random points in the ball*
- *See how many of them are in K*



welcome to Randomized Algorithms

fundamentalalgorithms.com/randomized

one big .pdf w/ everything

Randomized Algorithms, CS588, Fall 2025

Lectures: 4:30 to 5:45 PM, Tuesday and Thursday, in Forney Hall G124.

Staff:	Email (@purdue.edu)	Office hours
Kent Quanrud	krq	Tuesday, 2–3PM, Lawson 1211
Tanmay Devale	tdevale	TBD

Please see the syllabus (Page 391) for more information.

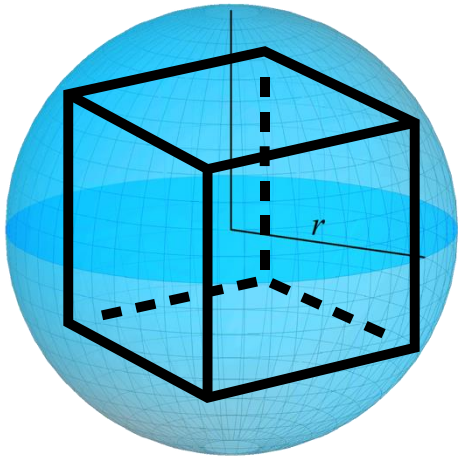
If you are unable to register for the class because it is full, just wait. Historically, slots have always opened up. You can add yourself to gradescope and submit homework in the meantime.

Example 1: Computing volumes in high dimensions

Theorem. A deterministic $2^{\delta d}$-approximation algorithm requires $2^{(1-\delta)d}$ oracle queries.

Maybe a Monte Carlo algorithm?

- *Pick random points in the ball*
- *See how many of them are in K*



Doesn't work in higher dimensions!

When K is a cube inside B_2^d,

$$\frac{\mathrm{Vol}(K)}{\mathrm{Vol}\left(B_2^d\right)} \leq e^{-\Omega(d)}$$

Need exponentially many points!

Example 1: Computing volumes in high dimensions

Dyer-Frieze-Kannan algorithm

1. *Change coordinates s.t. K is well-rounded, $B_2^d \subseteq K \subseteq RB_2^d$*
2. *Let $K_i \coloneqq 2^{i/d} B_2^d \cap K$ so that $K_0 = B_2^d$ and $K_0 \subseteq K_1 \subseteq \cdots \subseteq K_{m-1} \subseteq K_m \coloneqq K$*
3. *Compute*

$$\gamma_i \coloneqq \frac{\mathrm{Vol}(K_{i-1})}{\mathrm{Vol}(K_i)}$$

$\gamma_i \geq \frac{1}{2}$, estimate via Monte Carlo

4. *Output*

$$V \coloneqq \mathrm{Vol}(B) \cdot \prod_{i=1}^{m} \frac{1}{\gamma_i}$$

Using concentration inequality, we can show that $V \approx \mathbb{E}[V] = \mathrm{Vol}(K)$

$$\mathrm{Vol}(K) = \mathrm{Vol}(K_0) \cdot \frac{\mathrm{Vol}(K_1)}{\mathrm{Vol}(K_0)} \cdot \frac{\mathrm{Vol}(K_2)}{\mathrm{Vol}(K_1)} \cdot \cdots \cdot \frac{\mathrm{Vol}(K_m)}{\mathrm{Vol}(K_{m-1})}$$

Example 1: Computing volumes in high dimensions

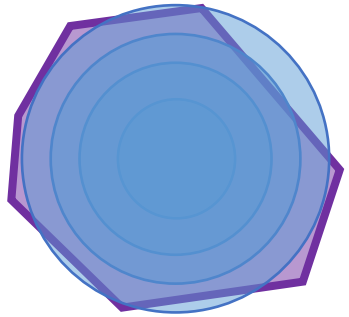
Dyer-Frieze-Kannan algorithm

1. *Change coordinates s.t. K is well-rounded, $B_2^d \subseteq K \subseteq RB_2^d$*
2. *Let $K_i := 2^{i/d} B_2^d \cap K$ so that $K_0 = B_2^d$ and $K_0 \subseteq K_1 \subseteq \cdots \subseteq K_{m-1} \subseteq K_m := K$*
3. *Compute*

$$\gamma_i := \frac{\mathrm{Vol}(K_{i-1})}{\mathrm{Vol}(K_i)} \qquad \gamma_i \geq \frac{1}{2}\text{, estimate via Monte Carlo}$$

4. *Output*

$$V := \mathrm{Vol}(B) \cdot \prod_{i=1}^{m} \frac{1}{\gamma_i}$$



DFK algorithm as a reduction:

volume estimation $\Longrightarrow$ sampling (uniformly sample from $rB_2^d \cap K$)

Example 1: Computing volumes in high dimensions

How to draw a uniformly random sample from a convex body?

- Basic idea: construct a Markov chain so that the stationary distribution is the target uniform distribution over the convex body

  "Running a randomized algorithm that simulate some stochastic process. When the simulation time $\to \infty$, the output distribution of the algorithm is the uniform distribution"

How fast does Markov chain mix? (How long should we run the simulation algorithm?)

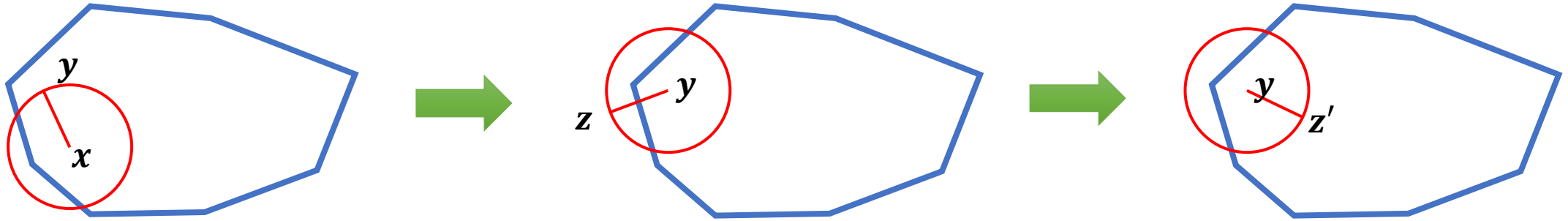
- Deep connection to important mathematical questions in *Asymptotic Convex Geometry* and *Functional Analysis* (e.g., isoperimetry, localization, concentration of measure, KLS conjecture, Bourgain's slicing conjecture, etc.)

We'll discuss some of them in this course

Example 1: Computing volumes in high dimensions

BallWalk(δ):

- Pick a uniform random point $\boldsymbol{y}$ from the ball of radius δ centered at the current point $\boldsymbol{x}$.
- If $\boldsymbol{y}$ is in K, go to $\boldsymbol{y}$; else stay at $\boldsymbol{x}$.



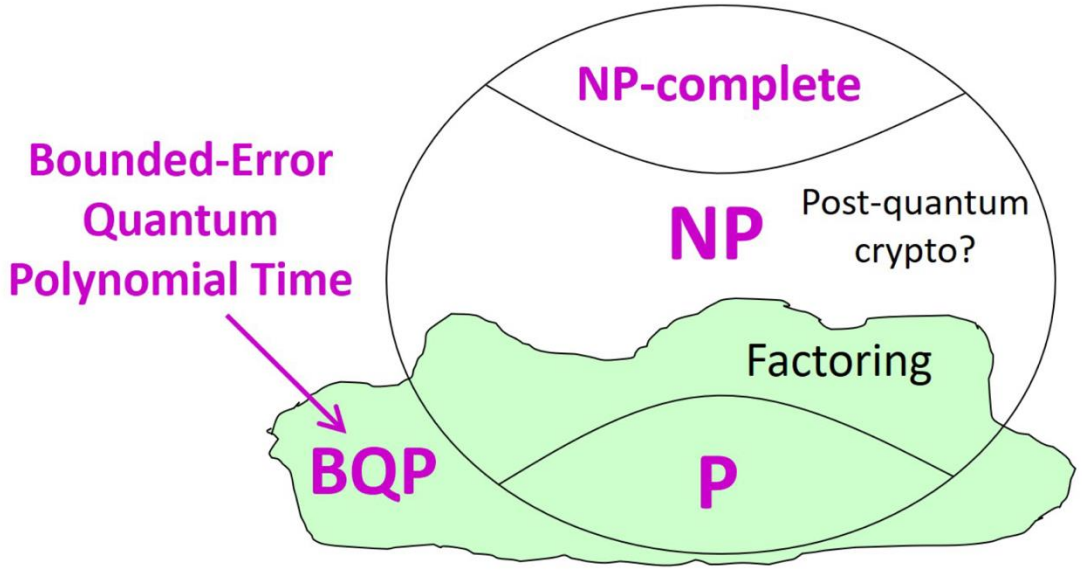
Lee-Vempala '16: under KLS conjecture and for a "good" convex body K, ball walk can sample uniformly over K in $\mathcal{O}(d^{2.5})$ steps

Example 2: Quantum and Boolean function analysis

Now, let's talk about quantum computing, a "new" topic in TCS

- It's actually not new at all.
  - Shor's algorithm was developed in 1994.
  - 'Quantum computation' became a topic of interest in STOC around 1997, and in FOCS in 1999.
- It's new probably because quantum computing headlines appear every day (>90% is hype 🤣)

How powerful is QC from a computational complexity perspective?

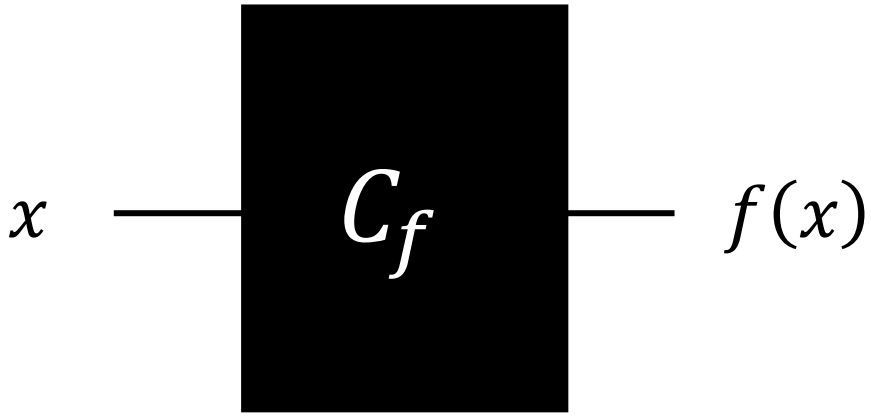


Example 2: Quantum and Boolean function analysis

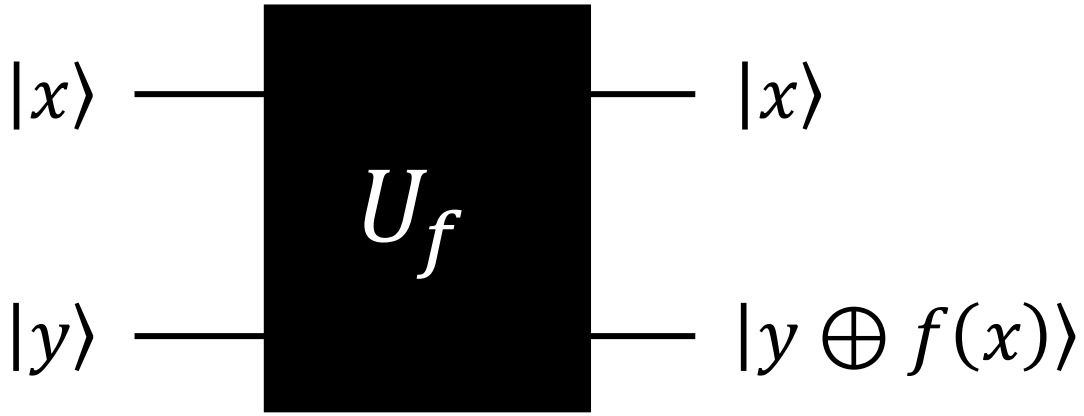
Black-box model / oracle model / query complexity model

- $f:\{0,1\}^n \to \{0,1\}$, and $\mathrm{tt}(f) \in \{0,1\}^{2^n}$ is the input of some decision problem

Classical query



Quantum query



- How many (classical/quantum) queries are needed to solve the decision problem?

Example 2: Quantum and Boolean function analysis

Positive result: Forrelation

Given $f, g \in \{0,1\}^n \to \{-1,1\}$, the forrelation is defined as

$$\Phi(f,g) \coloneqq 2^{-3n/2} \sum_{x,y \in \{0,1\}^n} f(x) g(y) (-1)^{x \cdot y}$$

≡ "Fourier Correlation"

Decide $|\Phi(f,g)| \leq 1/100$ or $|\Phi(f,g)| \geq 3/5$

- Quantum algorithm just need 1 query while any classical algorithm need $2^{\Omega(n)}$ queries!

Negative result: Aaronson-Ambainis Conjecture

Any quantum algorithm that makes T queries to compute a Boolean function, ∃ classical algorithm that makes only $\mathrm{poly(T)}$ queries and is correct for "most" of the inputs.

No "big" quantum speedup for generic unstructured decision problem!

Example 2: Quantum and Boolean function analysis

AA conjecture is equivalent to the following conjecture in Boolean function analysis with no quantum at all:

- Any function $f:\{-1,1\}^n \to \mathbb{R}$ can be expressed as

$$f(x) = \sum_{S\subset[n]} a_S \prod_{i\in S} x_i\,, \qquad a_S \in \mathbb{R}, \qquad x \in \{\pm 1\}^n$$

  and we define the degree of f as $d := \max_{a_S\neq 0} |S|$ (think of f as a polynomial)

- For any $f:\{\pm 1\}^n \to [0,1]$, there exists a coordinate $i \in [n]$ such that

$$\mathbb{E}[|D_i f|^2] \geq \left(\frac{\mathbb{E}\left[f - \mathbb{E}[f]\right]^2}{d}\right)^{\mathcal{O}(1)}, \qquad \text{where } (D_i f)(x) = \frac{1}{2}\left(f(x) - f(x_{-i})\right)$$

Influence vs. Variance

Flip the i-th bit

Announcements

1. I'll be out of town on Thursday (01/15), and the class will be canceled

2. Homework: watch the following video lectures by Ryan O'Donnell:
   - How to do CS Theory (https://youtu.be/YFUIPg8P2sY?si=lgqbx40_qv6Ouhka)
   - Street Fighting Mathematics (https://youtu.be/qP4XEZ54eSc?si=XxXrdhxuqMmdMKn_)