To Count or Not to Count

A Personal Perspective

Supratik Chakraborty
I.I.T. Bombay

A Short Talk at VardiFest 2022, Haifa

Some historical context ….

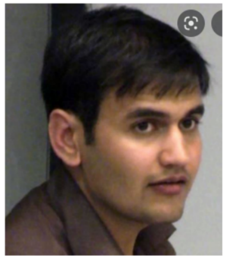
- FSTTCS, Dec 2011, Mumbai:
    - Moshe gives keynote talk, and Institute Colloquium talk

Some historical context …

- FSTTCS, Dec 2011, Mumbai:
    - Moshe gives keynote talk, and Institute Colloquium talk
    - During a break, we chat about a bright young undergraduate student from IIT Bombay who had just worked with Moshe during an exchange program

Some historical context

- FSTTCS, Dec 2011, Mumbai:
  - Moshe gives keynote talk, and Institute Colloquium talk
  - During a break, we chat about a bright young undergraduate student from IIT Bombay who had just worked with Moshe during an exchange program



Some historical context …

- FSTTCS, Dec 2011, Mumbai:
  - Moshe gives keynote talk, and Institute Colloquium talk
  - During a break, we chat about a bright young undergraduate student from IIT Bombay who had just worked with Moshe during an exchange program

Some historical context …

- FSTTCS, Dec 2011, Mumbai:
    - Moshe gives keynote talk, and Institute Colloquium talk
    - During a break, we chat about a bright young undergraduate student from IIT Bombay who had just worked with Moshe during an exchange program
    - We agree to jointly work with the student on uniform sampling of SAT solutions

Approximate Counting and Sampling with PAC guarantees

- FSTTCS, Dec 2011, Mumbai:
  - Moshe gives keynote talk, and Institute Colloquium talk
  - During a break, we chat about a bright young undergraduate student from IIT Bombay who had just worked with Moshe during an exchange program
  - We agree to jointly work with the student on uniform sampling of SAT solutions
- **And so a journey began in 2012 …**
  - Almost Uniform Sampling and Approximate Counting with XOR-based 2-universal hash functions

Approximate Counting and Sampling with PAC guarantees

- FSTTCS, Dec 2011, Mumbai:
  - Moshe gives keynote talk, and Institute Colloquium talk
  - During a break, we chat about a bright young undergraduate student from IIT Bombay who had just worked with Moshe during an exchange program
  - We agree to jointly work with the student on uniform sampling of SAT solutions
- **And so a journey began in 2012 …**
  - Almost Uniform Sampling and Approximate Counting with XOR-based 2-universal hash functions
    - Given propositional formula F with solution set Sol, params $e > 0$ and $0 < d < 1$
      - Generate a random y s.t.
        - $1/(1+ e) \times 1/|Sol| <= Pr[y \text{ is generated}] <= (1+e) \times 1/|Sol|$
      - Find a number C s.t.
        - $Pr[1/(1+e) \times C <= | Sol | <= (1+e) \times C] >= 1 - d$

Approximate Counting and Sampling with PAC guarantees

- Rich history of work from both theory and applied communities

Approximate Counting and Sampling with PAC guarantees

- Rich history of work from both theory and applied communities
  - Beautiful theoretical work with strong approximation bounds
    - **Scales only to few 10s of variables in practice**

Approximate Counting and Sampling with PAC guarantees

- Rich history of work from both theory and applied communities
  - Beautiful theoretical work with strong approximation bounds
    - **Scales only to few 10s of variables in practice**
  - Rich legacy of heuristics applied in varied application
    - **Hardly any rigorous guarantees**

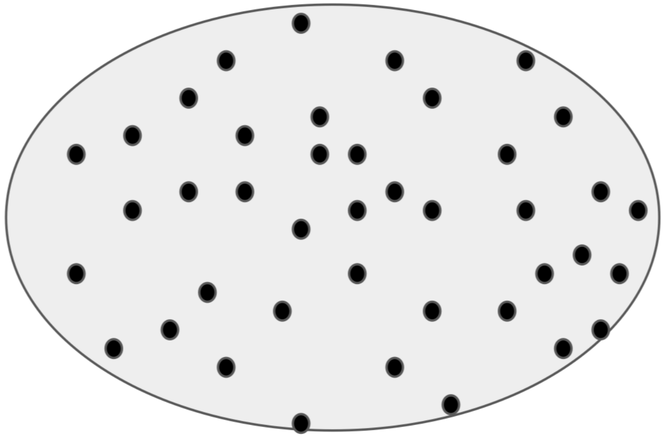
Approximate Counting and Sampling with PAC guarantees

- Rich history of work from both theory and applied communities
  - Beautiful theoretical work with strong approximation bounds
    - **Scales only to few 10s of variables in practice**
  - Rich legacy of heuristics applied in varied application
    - **Hardly any rigorous guarantees**

Moshe throws us a challenge:

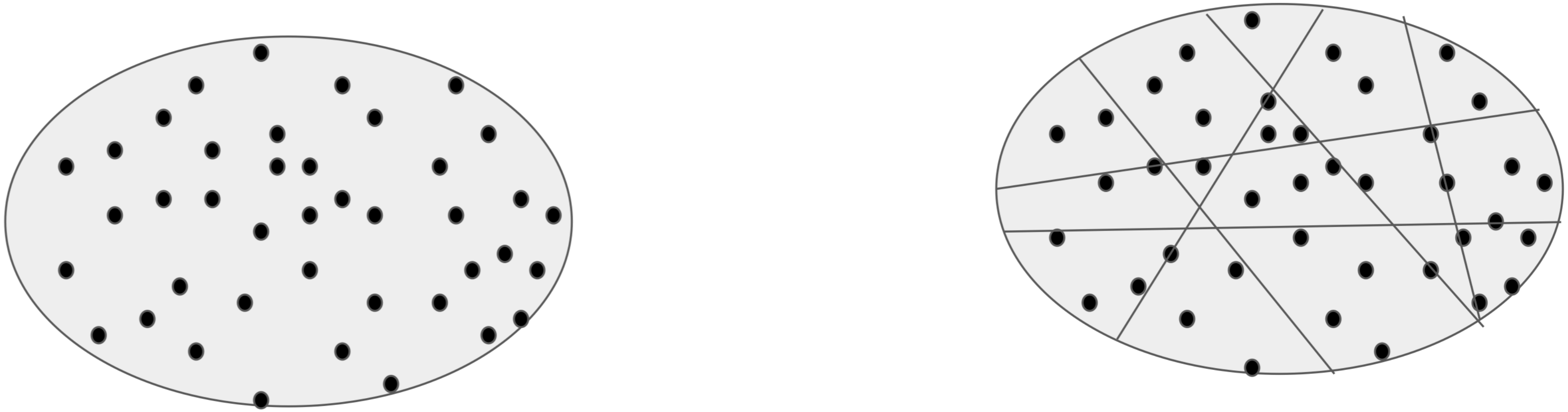
Can we marry rigorous approximation guarantees with practical scalability for counting & sampling?

The Key Idea



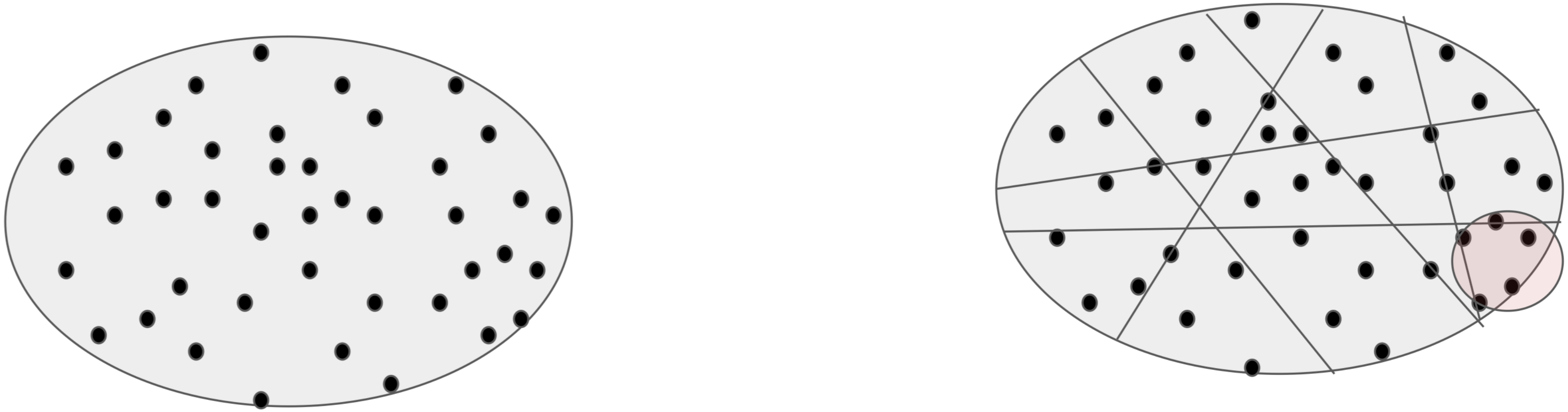
Solution space of F

The Key Idea



Use **r-independent universal hash function** to split solution space randomly until each cell is sufficiently “small”

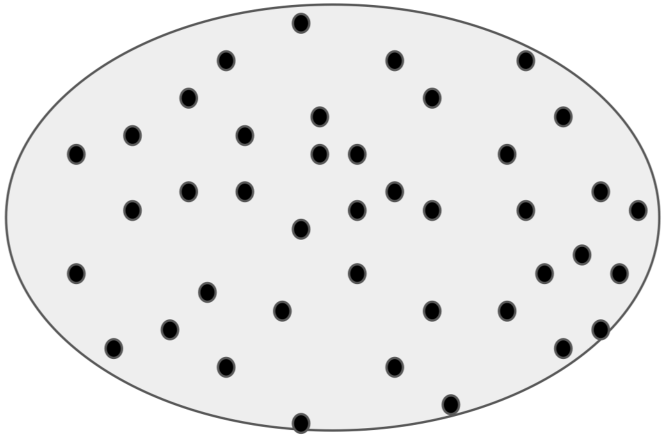
The Key Idea

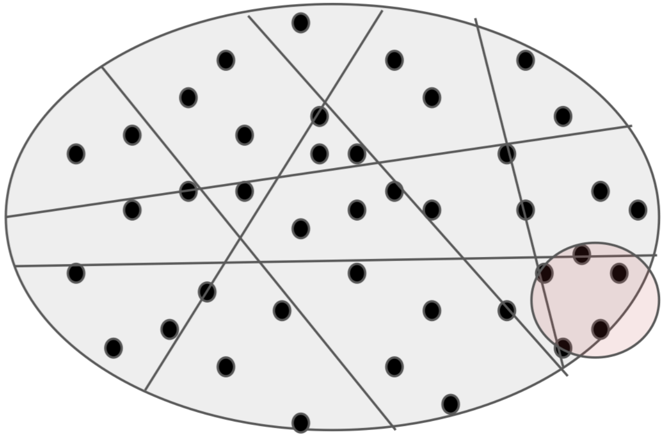


Use **r-independent universal hash function** to split solution space randomly until each cell is sufficiently “small”

Choose a random “small” cell

The Key Idea



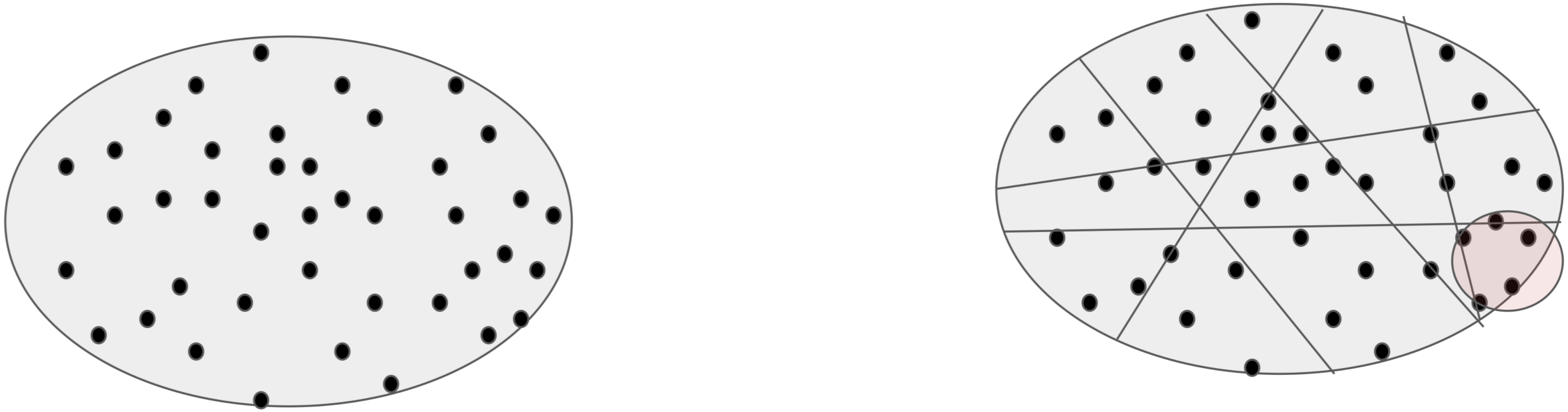


Use **r-independent universal hash function** to split solution space randomly until each cell is sufficiently “small”

Choose a random “small” cell

Sample/count from this “small” cell, and scale if needed to lift to original domain

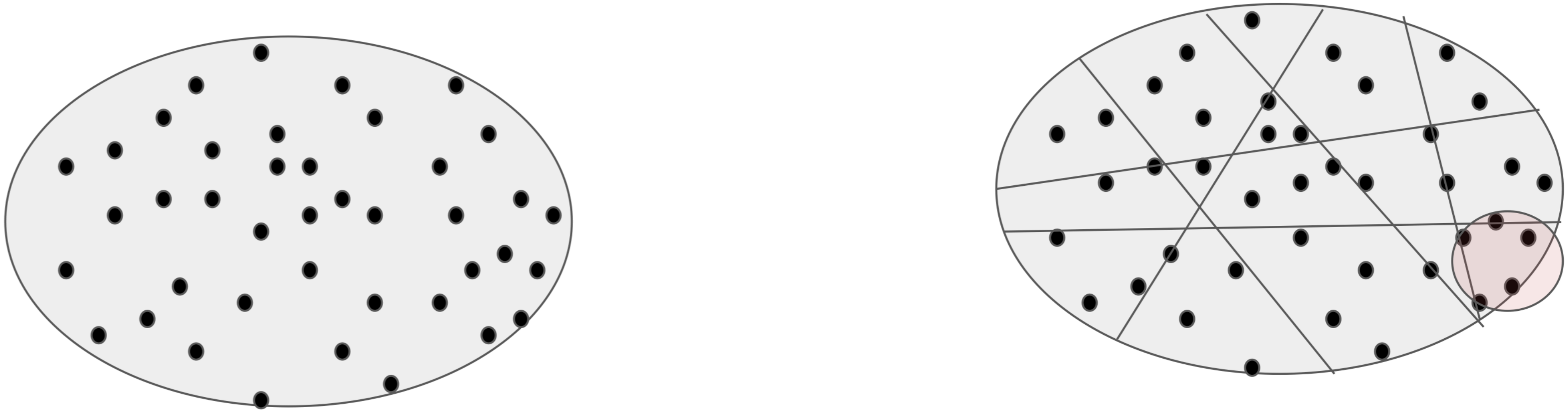
The Key Idea



r-independent universal hash functions: **How small can r be?**

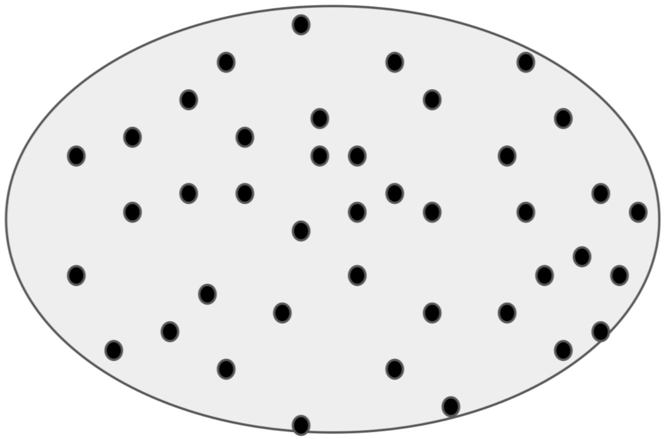
As r increases, stronger theoretical guarantees, but scalability setback

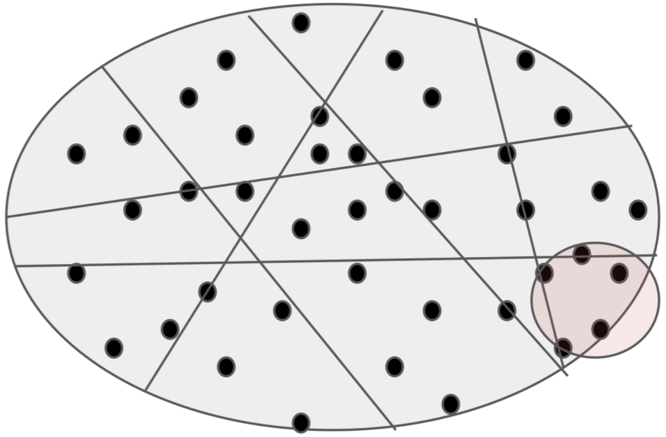
The Key Idea



Moshe’s insight: r somewhere between 2 and 3 should work

The Key Idea





Moshe’s insight: r somewhere between 2 and 3 should work

XOR-based hash functions fit the bill

Used earlier without strong guarantees

The Key Idea

Moshe’s insight: r somewhere between 2 and 3 should work

XOR-based hash functions fit the bill

Used earlier without strong guarantees

Kuldeep’s thesis showed that indeed Moshe was right.

Can marry rigorous guarantees (PAC) with scalability (~ 10^6 vars)

So, we know how to count and sample ?

So, we know how to count and sample ?

- Counting hardly does justice to all things whose value we care about

So, we know how to count and sample ?

- Counting hardly does justice to all things whose value we care about
- All the different things that I learnt (and continue to learn) from Moshe

So, we know how to count and sample ?

- Counting hardly does justice to all things whose value we care about
- All the different things that I learnt (and continue to learn) from Moshe
  - From my student days
  - Hard to put any quantitative measure on it
- Certain things are best left not counted

So, we know how to count and sample ?

- Counting hardly does justice to all things whose value we care about
- All the different things that I learnt (and continue to learn) from Moshe
  - From my student days
  - Hard to put any quantitative measure on it
- Certain things are best left not counted
- A small (not at all uniform) sample of a few results that have left deep impact on me

Automata Theoretic Verification

My graduate school days:

- Moshe Vardi and Pierre Wolper, “An Automata Theoretic Approach to Automatic Program Verification”, LICS 1986
    - “... for any temporal formula we can construct an automaton that accepts precisely the computations that satisfy the formula. The model-checking algorithm that results from this approach is much simpler and cleaner than tableau-based algorithms…”

Automata Theoretic Verification

My graduate school days:

- Moshe Vardi and Pierre Wolper, “An Automata Theoretic Approach to Automatic Program Verification”, LICS 1986
  - “... for any temporal formula we can construct an automaton that accepts precisely the computations that satisfy the formula. The model-checking algorithm that results from this approach is much simpler and cleaner than tableau-based algorithms…”
- Revolutionized and inspired large body of work on automated model checking
  - Tools seeded in this idea widely used in practice

Automata Theoretic Verification

My graduate school days:

- Moshe Vardi and Pierre Wolper, “An Automata Theoretic Approach to Automatic Program Verification”, LICS 1986
    - “... for any temporal formula we can construct an automaton that accepts precisely the computations that satisfy the formula. The model-checking algorithm that results from this approach is much simpler and cleaner than tableau-based algorithms…”
- Revolutionized and inspired large body of work on automated model checking
    - Tools seeded in this idea widely used in practice
- My canonical example of how boundaries between theoretical and applied Computer Science are best broken

Logic and Finite Model Theory

My mid-career days, advising my student working on logic:

- Ron Fagin, “Generalized first-order Spectra and Polynomial-time Recognizable Sets”, Complexity of Computation 1973
- Moshe Vardi, “The Complexity of Relational Query Languages”, STOC 1982
    - Introduction of data complexity and query complexity of logical languages
    - Various extensions and their powers and limitations
- Neil Immerman, “Relational Queries Computable in Polynomial Time”, Information and Control 1986

Logic and Finite Model Theory

My mid-career days, advising my student working on logic:

- Ron Fagin, “Generalized first-order Spectra and Polynomial-time Recognizable Sets”, Complexity of Computation 1973
- Moshe Vardi, “The Complexity of Relational Query Languages”, STOC 1982
  - Introduction of data complexity and query complexity of logical languages
  - Various extensions and their powers and limitations
- Neil Immerman, “Relational Queries Computable in Polynomial Time”, Information and Control 1986

- Opened up new connections between logic, finite structures and complexity
- The inspiration from these and related papers led my student to complete his Ph.D. dissertation on logic and model theory in 2016

Certain Things are Best Not Counted

- Constrained (and biased) sample of Moshe’s work that influenced me
- Lost count of how many pearls of wisdom from Moshe I have benefited from over the years

Certain Things are Best Not Counted

- Constrained (and biased) sample of Moshe’s work that influenced me
- Lost count of how many pearls of wisdom from Moshe I have benefited from over the years

Certain things in life just cannot be counted, or even sampled uniformly

Certain Things are Best Not Counted

- Constrained (and biased) sample of Moshe's work that influenced me
- Lost count of how many pearls of wisdom from Moshe I have benefited from over the years

Certain things in life just cannot be counted, or even sampled uniformly

Thank you, Moshe!

Priceless Moments – No Algo can Count Their Value

Moshe and Phokion with my 2-year old daughter in Dec 2011