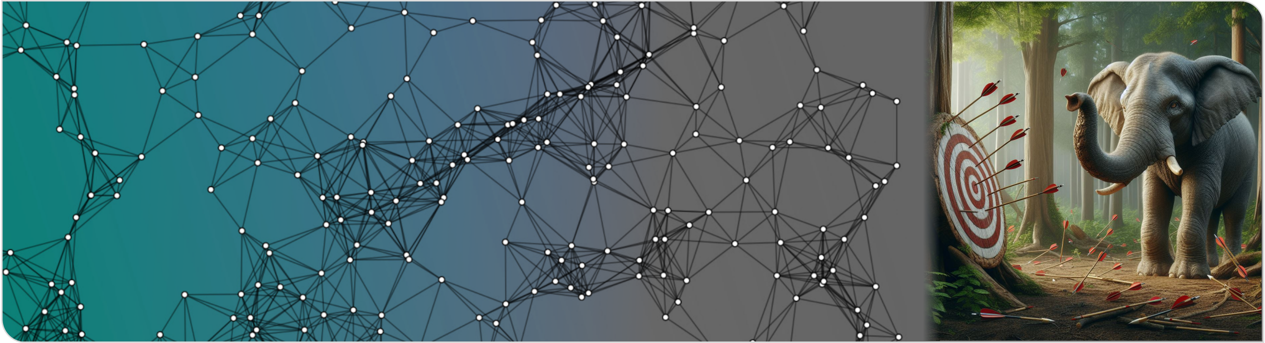
Probability and Computing – Approximation Algorithms

Stefan Walzer, Maximilian Katzmann | WS 2023/2024



Lecture Notes by Worsch

This lecture's content is covered in Thomas Worsch's notes from 2019.

Content

1. What is Randomised Approximation?

2. Approximately counting satisfying assignments for Boolean formulas

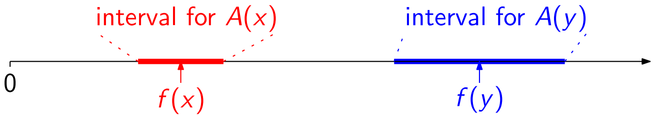
Randomised Approximate Counting

Definition

A randomised algorithm A *approximates* a quantity $f(x)$ if for any input x the output $A(x)$ satisfies:

$$\Pr[|A(x) - f(x)| \leq \varepsilon \cdot f(x)] \geq 1 - \delta.$$

The parameters are the *relative error* ε and the *failure probability* δ.



$(\varepsilon = \frac{1}{4})$

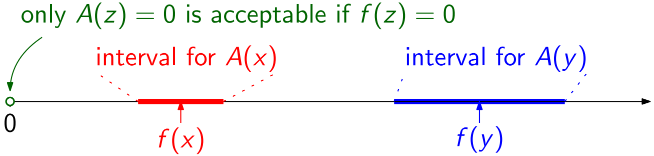
Randomised Approximate Counting

Definition

A randomised algorithm A *approximates* a quantity $f(x)$ if for any input x the output $A(x)$ satisfies:

$$\Pr[|A(x) - f(x)| \leq \varepsilon \cdot f(x)] \geq 1 - \delta.$$

The parameters are the *relative error* ε and the *failure probability* δ.



$(\varepsilon = \frac{1}{4})$

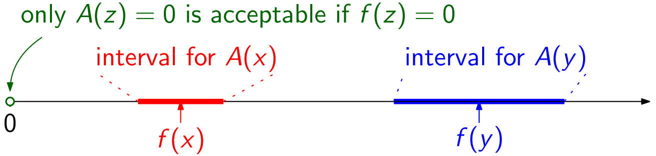
Randomised Approximate Counting

Definition

A randomised algorithm A *approximates* a quantity $f(x)$ if for any input x the output $A(x)$ satisfies:

$$\Pr[|A(x) - f(x)| \leq \varepsilon \cdot f(x)] \geq 1 - \delta.$$

The parameters are the *relative error* ε and the *failure probability* δ.



$(\varepsilon = \frac{1}{4})$

Remark: Related Complexity Classes

PRAS. Problems admitting A with running time polynomial in $|x|$, but not necessarily in $\frac{1}{\varepsilon}$ (for $\delta = 1/4$).

FPRAS. Problems admitting A with running time polynomial in $|x|$ and $\frac{1}{\varepsilon}$ (for $\delta = 1/4$).

Note: Also defined where $f(x)$ is not a *number*. For instance: Want to compute a *vertex cover* with a size close to optimal.

Counting Satisfiable Assignments of Boolean Formulas

Karlsruhe Institute of Technology

A counting problem

For Boolean formula $B(x_1, \ldots, x_n)$ let $\#B$ be the number of satisfying assignments:

$$\#B = |\{(x_1, \ldots, x_n) \in \{0,1\}^n \mid B(x_1, \ldots, x_n) = 1\}|.$$

Example

$$B = (x_1 \vee \bar{x}_2) \wedge (\bar{x}_1 \vee x_3)$$

$$\#B = |\{(0,0,0), (0,0,1), (1,0,1), (1,1,1)\}| = 4$$

Counting Satisfiable Assignments of Boolean Formulas

Karlsruhe Institute of Technology

A counting problem

For Boolean formula $B(x_1, \dots, x_n)$ let $\#B$ be the number of satisfying assignments:

$$\#B = |\{(x_1, \dots, x_n) \in \{0,1\}^n \mid B(x_1, \dots, x_n) = 1\}|.$$

Example

$$B = (x_1 \vee \bar{x_2}) \wedge (\bar{x_1} \vee x_3)$$
$$\#B = |\{(0,0,0), (0,0,1), (1,0,1), (1,1,1)\}| = 4$$

Approximation algorithm for $\#B$ in general? Unlikely.

Assume A satisfies $\Pr[|A(B) - \#B| \le \varepsilon(\#B)] \le 1 - \delta$ for $\varepsilon = \frac{1}{2}$ and $\delta = \frac{1}{4}$. Then

$$B \text{ is UNSAT} \Leftrightarrow \#B = 0 \Rightarrow \Pr[|A(B) - 0| \le \tfrac{1}{2} \cdot 0] \ge \tfrac{3}{4} \Rightarrow \Pr[A(B) = 0] \ge \tfrac{3}{4}$$
$$B \text{ is SAT} \Leftrightarrow \#B > 0 \Rightarrow \Pr[|A(B) - \#B| \le \tfrac{1}{2} \cdot \#B] \ge \tfrac{3}{4} \Rightarrow \Pr[A(B) > 0] \ge \tfrac{3}{4}$$

If A is polynomial time then A is BPP algorithm for SAT.
Then SAT $\in$ BPP and NP $\subseteq$ BPP. Hard to believe...

What could be a tractable special case?

Relative error $\varepsilon < 1$ requires distinguishing:

$$\text{UNSAT} \Leftrightarrow \#B = 0 \qquad \text{from} \qquad \text{SAT} \Leftrightarrow \#B \geq 1.$$

What could be a tractable special case?

Relative error $\varepsilon < 1$ requires distinguishing:

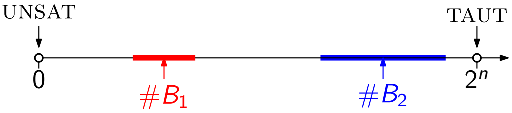
UNSAT $\Leftrightarrow \#B = 0$ from SAT $\Leftrightarrow \#B \geq 1$.

CNF is hopeless

$$B = (x_1 \vee \bar{x}_2 \vee \bar{x}_{42}) \wedge \ldots \wedge (\bar{x}_1 \vee x_3 \vee \bar{x}_{37})$$

deciding SAT is NP-hard for clause size 3.

What could be a tractable special case?



Relative error $\varepsilon < 1$ requires distinguishing:

$$\text{UNSAT} \Leftrightarrow \#B = 0 \qquad \text{from} \qquad \text{SAT} \Leftrightarrow \#B \geq 1.$$

An asymmetry for CNF formulas

- B is called TAUTology if $\#B = 2^n$.
- "is B TAUT?" is easy to decide:
  Only empty CNF-formula is TAUT.
  (assuming x_i and $\bar{x}_i$ never in the same clause)
- Try approximating *unsatisfying* assignments?

$$f(x) := 2^n - \#B.$$

CNF is hopeless

$$B = (x_1 \vee \bar{x}_2 \vee \bar{x}_{42}) \wedge \ldots \wedge (\bar{x}_1 \vee x_3 \vee \bar{x}_{37})$$

deciding SAT is NP-hard for clause size 3.

What could be a tractable special case?

Relative error $\varepsilon < 1$ requires distinguishing:

$$\textsc{unsat} \Leftrightarrow \#B = 0 \qquad \text{from} \qquad \textsc{sat} \Leftrightarrow \#B \geq 1.$$

CNF is hopeless

$$B = (x_1 \vee \bar{x}_2 \vee \bar{x}_{42}) \wedge \ldots \wedge (\bar{x}_1 \vee x_3 \vee \bar{x}_{37})$$

deciding SAT is NP-hard for clause size 3.

An asymmetry for CNF formulas

- B is called TAUTology if $\#B = 2^n$.
- "is B TAUT?" is easy to decide:
  Only empty CNF-formula is TAUT.
  (assuming x_i and $\bar{x}_i$ never in the same clause)
- Try approximating *unsatisfying* assignments?

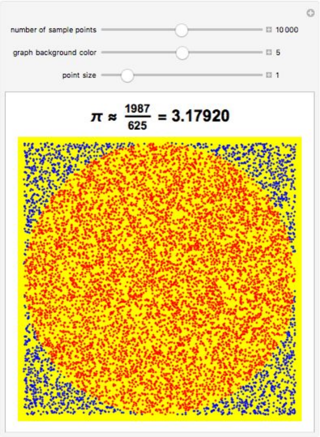
$$f(x) := 2^n - \#B.$$

Consider DNF!

$$B' = \bar{B} = (\bar{x}_1 \wedge x_2 \wedge x_{42}) \vee \ldots \vee (x_1 \wedge \bar{x}_3 \wedge x_{37})$$

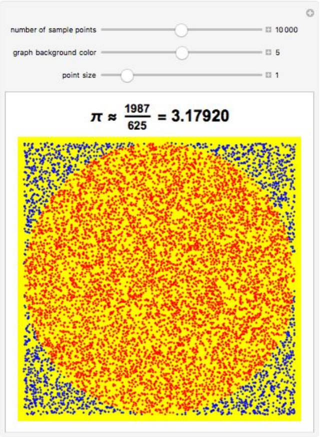
- $\#B' = 2^n - \#B$.
- "B' is SAT" is easy to decide
  (only empty DNF-formula is UNSAT.)

Intuition: Approximating π



https://demonstrations.wolfram.com/ApproximatingPiByTheMonteCarloMethod/

Intuition: Approximating π



https://demonstrations.wolfram.com/ApproximatingPiByTheMonteCarloMethod/

Requirements for estimating area of disk (and hence π):

- Know formula for area of square
- Sample uniformly from square
- decide for $x, y \in [-1, 1]$ if (x, y) in disk: $x^2 + y^2 \leq 1$

Approximate $|S|$ for $S \subseteq D$ by naive sampling

Algorithm approxSetSize($\mathbb{1}_{\cdot \in S}$, D)**:**

$\quad$ hits $\leftarrow 0$

$\quad$ **for** $i = 1$ *to* N **do**

$\quad\quad$ sample $x \sim \mathcal{U}(D)$

$\quad\quad$ hits $\leftarrow$ hits $+ \mathbb{1}_{x \in S}$

$\quad$ **return** $\frac{\text{hits}}{N} \cdot |D|$

Simple Theorem

Let D be a finite set and $S \subseteq D$ such that we can efficiently

- compute $|D|$
- sample uniformly from D
- decide for given $x \in D$ whether $x \in S$

Approximate $|S|$ for $S \subseteq D$ by naive sampling

Algorithm approxSetSize($\mathbb{1}_{\cdot \in S}$, D)**:**

```
hits ← 0
for i = 1 to N do
    sample x ~ U(D)
    hits ← hits + 1_{x∈S}
return (hits/N) · |D|
```

Simple Theorem

Let D be a finite set and $S \subseteq D$ such that we can efficiently

- compute $|D|$
- sample uniformly from D
- decide for given $x \in D$ whether $x \in S$

Let $p = |S|/|D|$. Then approxSetSize with $N = \frac{3\log(2/\delta)}{\varepsilon^2 p}$ approximates $|S|$ with relative error ε and failure probability δ.

$\hookrightarrow$ Special Case $\varepsilon, \delta = \Theta(1)$: Need $N = \Omega(1/p)$ samples.

Approximate $|S|$ for $S \subseteq D$ by naive sampling

Algorithm approxSetSize($\mathbb{1}_{\cdot \in S}$, D):

$\text{hits} \leftarrow 0$
for $i = 1$ *to* N **do**
  sample $x \sim \mathcal{U}(D)$
  $\text{hits} \leftarrow \text{hits} + \mathbb{1}_{x \in S}$
return $\frac{\text{hits}}{N} \cdot |D|$

Simple Theorem

Let D be a finite set and $S \subseteq D$ such that we can efficiently

- compute $|D|$
- sample uniformly from D
- decide for given $x \in D$ whether $x \in S$

Let $p = |S|/|D|$. Then approxSetSize with $N = \frac{3\log(2/\delta)}{\varepsilon^2 p}$ approximates $|S|$ with relative error ε and failure probability δ.

$\hookrightarrow$ Special Case $\varepsilon, \delta = \Theta(1)$: Need $N = \Omega(1/p)$ samples.

Chernoff ($\rightarrow$ Concentration, slide 15)

For $\varepsilon \in (0,1)$ and $X \sim Bin(N, p)$:

$$\Pr[|X - \mathbb{E}[X]| > \varepsilon\mathbb{E}[X]] < 2\exp(-\varepsilon^2\mathbb{E}[X]/3).$$

Proof: Apply Chernoff to hits $\sim Bin(N, p)$.

$$\Pr[\text{fail}] = \Pr\left[\left|\text{result} - |S|\right| > \varepsilon|S|\right] = \Pr\left[\left|\frac{\text{hits}}{N} \cdot |D| - |S|\right| > \varepsilon|S|\right] = \Pr\left[\left|\text{hits} - \frac{|S|}{|D|}N\right| > \varepsilon\frac{|S|}{|D|}N\right]$$

$$= \Pr\left[\left|\text{hits} - pN\right| > \varepsilon pN\right] = \Pr\left[\left|\text{hits} - \mathbb{E}[\text{hits}]\right| > \varepsilon\mathbb{E}[\text{hits}]\right] \leq 2\exp(-\varepsilon^2\mathbb{E}[\text{hits}]/3) = 2\exp(-\varepsilon^2 pN/3) = \delta.$$

Approximate $|S|$ for $S \subseteq D$ by naive sampling

Algorithm approxSetSize($\mathbb{1}_{\cdot \in S}$, D):

hits $\leftarrow 0$
for $i = 1$ *to* N **do**
  sample $x \sim \mathcal{U}(D)$
  hits $\leftarrow$ hits $+ \mathbb{1}_{x \in S}$
return $\frac{\text{hits}}{N} \cdot |D|$

Chernoff ($\rightarrow$ Concentration, slide 15)

For $\varepsilon \in (0,1)$ and $X \sim Bin(N,p)$:

$$\Pr[|X - \mathbb{E}[X]| > \varepsilon \mathbb{E}[X]] < 2\exp(-\varepsilon^2 \mathbb{E}[X]/3).$$

Simple Theorem

Let D be a finite set and $S \subseteq D$ such that we can efficiently

- compute $|D|$
- sample uniformly from D
- decide for given $x \in D$ whether $x \in S$

Let $p = |S|/|D|$. Then approxSetSize with $N = \frac{3\log(2/\delta)}{\varepsilon^2 p}$ approximates $|S|$ with relative error ε and failure probability δ.

$\hookrightarrow$ Special Case $\varepsilon, \delta = \Theta(1)$: Need $N = \Omega(1/p)$ samples.

Application to $\#B$

- $S =$ satisfying assignments of B
- $D = \{0,1\}^n$
- $p = \frac{|S|}{|D|} = \frac{\#B}{2^n}$
- We may have $p = 1/2^n$
- $N = \Omega(2^n)$ required
- :-(

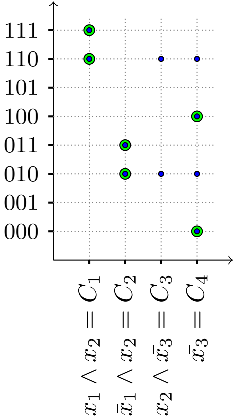
No Surprise

Of course this didn't work

Did not exploit that B is in DNF.

Approximating $\#B$ for B in DNF

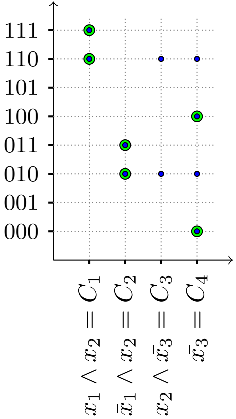
Assume $B = C_1 \vee \ldots \vee C_m$
where C_i contains ℓ_i literals.



- $D_i := \{x \in \{0,1\}^n \mid C_i(x) = 1\}$ (satisfying assignments of C_i)
- $D := \{(i, x) \mid i \in [m], x \in D_i\}$ $(= D_1 \dot\cup \ldots \dot\cup D_m)$
- $S := \{(i, x) \mid i \in [m], x \in D_i, x \notin D_1 \cup \ldots D_{i-1}\}$

Approximating $\#B$ for B in DNF

Assume $B = C_1 \vee \ldots \vee C_m$ where C_i contains ℓ_i literals.



- $D_i := \{x \in \{0,1\}^n \mid C_i(x) = 1\}$ (satisfying assignments of C_i)
- $D := \{(i,x) \mid i \in [m], x \in D_i\}$ $(= D_1 \dot\cup \ldots \dot\cup D_m)$
- $S := \{(i,x) \mid i \in [m], x \in D_i, x \notin D_1 \cup \ldots D_{i-1}\}$

Observations

- $|S| = \#B$
- $|D_i| = 2^{n-\ell_i}$ and we can efficiently sample from $\mathcal{U}(D_i)$:
  $\hookrightarrow$ set variables appearing in C_i as required, others from $Ber(1/2)$.
- We can efficiently compute $|D| = \sum_{i=1}^{m} |D_i|$ and sample $(I,X) \sim \mathcal{U}(D)$:
  - First sample I such that $\Pr[I = i] = \frac{|D_i|}{|D|}$.
  - Then sample $X \sim \mathcal{U}(D_i)$.
  - Yields $\Pr[(I,X) = (i,x)] = \frac{|D_i|}{|D|} \cdot \frac{1}{|D_i|} = \frac{1}{|D|}$ for all $(i,x) \in D$.
- We can efficiently decide "is $(i,x) \in S$?" (in time $\mathcal{O}(mn)$)
- $p = \frac{|S|}{|D|}$ satisfies $p \geq \frac{1}{m}$.

Takeaway

Karlsruhe Institute of Technology

Theorem

If B is in DNF, then we can approximate $\#B$ in polynomial time (using $N = m \cdot \frac{3\log(2/\delta)}{\varepsilon^2}$ samples) with relative error ε and failure probability δ.

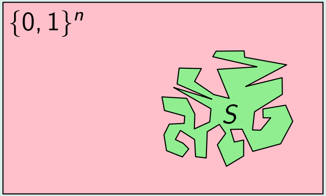
Takeaway

Theorem

If B is in DNF, then we can approximate $\#B$ in polynomial time (using $N = m \cdot \frac{3\log(2/\delta)}{\varepsilon^2}$ samples) with relative error ε and failure probability δ.

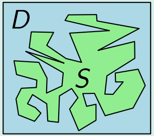
Intuition: Why did this work?

Naive strategy:



Problem: $|S|/|\{0,1\}^n|$ may be exponentially small

Improved strategy:



Advantage: $|S|/|D|$ is $\Omega(1/m)$.

Conclusion

Karlsruhe Institute of Technology

Randomised Approximation is Powerful

For B in DNF:

- Computing $\#B$ *exactly* is $\#\boldsymbol{P}$-complete.
- no *deterministic approximation* algorithm for such problems is known
- we analysed an efficient *randomised approximation* algorithm

Anhang: Mögliche Prüfungsfragen

- Was ist ein randomisierter Approximationsalgorithmus (für ein Zählproblem)?
- Wir haben das Zählproblem $\#B$ für Boolesche Formeln betrachtet. Hatten wir im allgemeinen Fall Erfolg? Warum nicht?
- Welchen Spezialfall haben wir uns vorgenommen? Wieso tritt dort nicht das selbe Problem auf wie im allgemeinen Fall?
- Wir haben einen Algorithmus gesehen der für zwei Mengen $S \subseteq D$ die Größe von $|S|$ schätzt.
  - Unter welchen Annahmen ist dieser anwendbar?
  - Wie hat der Algorithmus funktioniert?
  - Wie hängt die Anzahl der nötigen samples von $|S|$ und $|D|$ ab?
- Um $\#B$ für DNF Formel B zu schätzen haben wir einen schlaueren Ansatz kennengerlernt.
  - Wie hat dieser funktioniert?
  - Wie vermeidet dieser das Problem das naiven Ansatzes?